Signal Processing Module for a Digital Audio Amplifier Using Click Modulation

João Líbano Monteiro
Instituto Superior Técnico
Universidade de Lisboa
Email:joaolm1989@gmail.com

***Abstract*—In the context of class D audio amplifiers, Click Modulation is presented as an alternative to existing modulations, namely Pulse-Width Modulation and Sigma-Delta Modulation. The spectrum of the resulting binary wave carries several benefits in comparison with the spectra of the previous referred modulations, such as a distortion-free baseband and a guard-band between the baseband and the high frequency components. In addition, the switching frequency of the signal is considerably low, its order of magnitude being the same as half of the input signal's Nyquist frequency. Such performance is obtained at the cost of a complex signal processing. This thesis comprises a description of a real-time implementation of Click modulation in C programming language, with fixed-point operands, whose work was based in a MATLAB algorithm implementation. The signal processing unit reads the input signal file and produces a set of duty cycles of a pulse-width modulated wave that are written to another file. The obtained results, which were compared with those of the MATLAB model, were deemed substantially satisfactory, with both purely sinusoidal and multi-tone inputs. The spectra of the various generated signals, namely the pulse-width modulated synthesized from the duty cycles, not only proved the correct algorithm operation but also validated its use in the context of high-definition audio.**

I. Introduction

Nowadays switching amplifiers are growing in the audio market. The main reason behind this is the urge to increase operated gadgets autonomy, making the class-D theoretical efficiency of 100% very appealing when considering the more traditional solutions as class-A, Class-AB or class-C with maximum efficiencies of 25%, 78.5% and 90% respectively [10]. Class-D amplifiers need a binary signal to be driven that is normally obtained by a Pulse Width Modulation (PWM) process. The most common PWM processes are Uniform PWM (UPWM), Natural PWM (NPWM) and Sigma-Delta Modulation (SDM).

A new kind of modulation, Click Modulation (CM), was presented by F. Logan in [4] with an analogue theoretical proposal. Later other authors proposed digital implementations of this technique. Based on CM, a PWM system is design in [1] and tested in [2], where a numeric model of a digital implementation shows excellent results when comparing with other modulations. To ensure a competitive hardware solution based on [1], a hardware-like description of a real-time implementation of Click Modulation in C programming language was developed.

II. Click Modulation

Click Modulation was first demonstrated by B. F. Logan in 1982 [4] [5] [6] [7] [3] with a continuous time domain solution proposed. The proposed modulation aims the codification of band limited, $[\Omega_L, \Omega_H]$, analogue signals, $f(t)$, in a train of impulses with the same baseband spectrum. This train of impulses could be then converted in a PWM-like binary signal with the same characteristic. The output spectrum is presented in figure 1, where the spectrum shows the original signal between Ω_L and Ω_H and a guard band between Ω_H and Ω_S separating the message from the noise introduced by the process. Such a signal is suitable for class-D operation as the original signal can be recovered by low-pass filtering with a characteristic similar to $H_A(f)$.

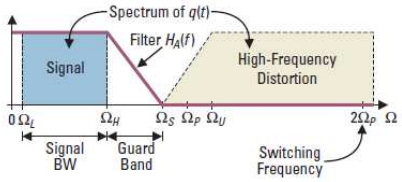

Figure 1: PWM signal spectrum generated by CM. [8].

A CM system follows a set of operations that are shown in figure 2.

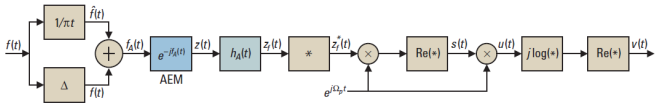

Figure 2: Theoretical scheme of CM operations [8].

For a hardware-like implementation the previous operations are grouped in the blocks presented in figure 3. The description

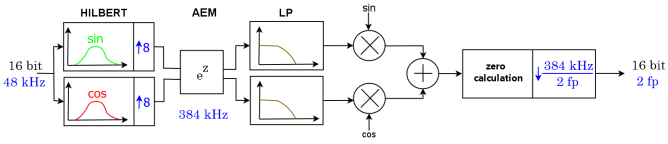

Figure 3: Hardware scheme of CM operations.

of the modulation theory that follows will be based on the hardware block division.

1) Hilbert Transform: In the first part, a Pulse-Coded modulated (PCM) signal with a possible spectrum in figure 4 is fed into a Hilbert transformer that produces a complex analytic signal, whose frequency components are only present in the positive frequency axis.

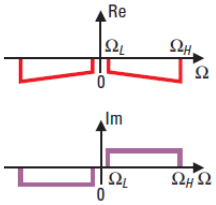

Figure 4: Spectrum of $f(t)$ [8].

The signal produced in this block is

$$f_A(t) = f(t) + j\hat{f}(t), \tag{1}$$

with $\hat{f}(t)$ being the Hilbert Transform of $f(t)$. The Hilbert Transform of a function inverts the sign of its spectrum components in the negative frequency axis. Therefore, when summed with $f(t)$, the resulting spectrum on the negative frequency axis will vanish, producing the mentioned analytic signal $f_A(t)$, solely with positive frequency components (figure 5).

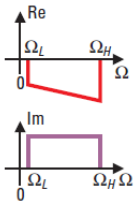

Figure 5: Spectrum of $f_A(t)$ [8].

2) Analytical Exponential Modulator and filtering: After (1), the signal is exponentiated, causing the spectrum to spread along all the positive frequencies, and filtered with a low-pass filter to prevent aliasing. The signal produced is

$$z_f(t) = z(t) * h_A(t) = e^{-jf_A(t)} * h_A(t) = x_f(t) + jy_f(t), \tag{2}$$

where $z(t)$ is the exponential of an analytical function and $h_A(t)$ is the impulse response of the low-pass filter. At this stage, the high frequency components that will form a PWM type wave are created (figure 6).

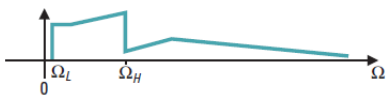

Figure 6: Spectrum of $z(t)$ [8].

The low-pass filter frequency response has a gain of 1 until Ω_S and a transition band until Ω_U, from which it has no gain, as it can be seen on figure 7. This filter is also of key importance since the guard-band of the signal is limited by the corner frequency of this filter [4].

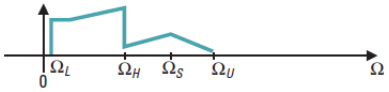

Figure 7: Spectrum of $z_f(t)$ [8].

It is important to refer at this point that an important restriction must be respected in order to guarantee the correct functioning of the modulation: the input signal amplitude must be restricted to a maximum of $\frac{\pi}{2}$ or minimum of $-\frac{\pi}{2}$ [4].

3) Quadrature Modulator: The quadrature modulator computes the following expression:

$$s(t) = Re\{z_f^*(t)e^{j\Omega_p t}\} = x_f(t)\cos(\Omega_p t) + y_f(t)\sin(\Omega_p t), \tag{3}$$

In this stage, $z_f(t)$ is conjugated and its real and imaginary parts, $x_f(t)$ and $y_f(t)$, are multiplied by $\cos(\Omega_p t)$ and $\sin(\Omega_p t)$, shifting the center of its spectra to Ω_p. The order of magnitude of this frequency can be similar to Ω_H. The sum in (6) yields $s(t)$, in whose zero crossings are contained the switching instants for the PWM signal [3].

4) Zero Crossing: It is possible to generate a PWM type wave $q(t)$ from both $s(t)$ and $\sin(\Omega_P t)$ [8], from

$$q(t) = -\frac{\pi}{2}\left\{sgn[s(t)]\right\} \cdot \left\{sgn[\sin(\Omega_p t)]\right\}. \tag{4}$$

This expression contains information about the roots of both signals, as the sign of each one changes accordingly with its zero-crossings. Finally, as a result, $f(t)$ can be recovered from $q(t)$ by simply applying to it a filter with a characteristic similar to $H_A(f)$.

III. Comparison with other modulations

A PWM wave generated by CM presents several advantages compared with its competitors, namely UPWM, NPWM and SDM. In UPWM, the input signal is fed through a sample-and-hold unit, whose result is then compared with a triangular or sawtooth wave. The implementation scheme is shown in 8.

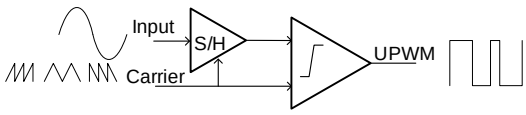

Figure 8: UPWM scheme [2].

The PWM wave generated contains harmonic distortion in integer multiples of the input frequency wave, as it is shown in figure 9, a problem that can only be solved by significanty increasing the input frequency sampling of the S/H unit.

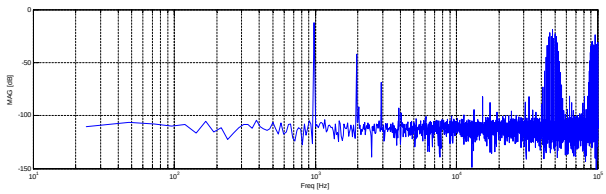

Figure 9: UPWM wave spectrum with input signal at 1 kHz [2].

In the case of NPWM, whose scheme is in figure 10, the type of distortion that degrades the signal the most is caused by aliased components from intermodulation components between the input signal and the carrier frequency that fall into the passband (non-harmonic distortion). This effect is visible in figure 11, and to overcome this issue higher carrier frequencies are used. However, as a direct result, the switching losses in the output stage components will also increase.

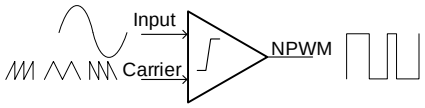

Figure 10: NPWM scheme [2].

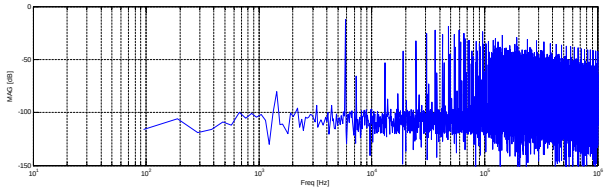

Figure 11: UPWM wave spectrum with input signal at 1 kHz [2].

The 1 bit SDM is the most common modulation (figure 12) technique used to generate a PWM like wave of this type.

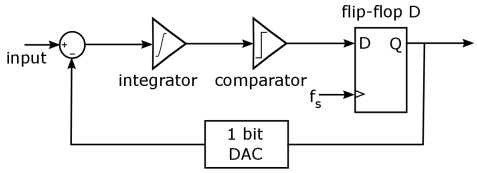

Figure 12: First order SD scheme.

The spectral characteristics of the PWM wave suffer an improvement through the effect of noise shaping at the quantization stage. Nevertheless, SD implementations are not as easy as UPWM or NPWM, not only because of the use of high intermediate sampling frequencies, but also because of possible instability issues. Due to its feedback topology, SD circuits above 2nd order may be unstable, as opposed to CM implementations which, with its open loop layout, have inheritable stability.

IV. State of the Art

A. Click Modulations implementations

There has been essentially two types of digital implementation of CM: offline floating-point in software and real-time fixed-point in hardware.

In the first type, three projects from the same authors [13] [8] [12] were developed at different stages, the latest [12] being the only that covers the whole audio bandwidth, in fact, even more (30 kHz). As no additional information is provided, it is assumed that the operands are 32 bit floating point format. No information is provided as to the implementation of the different blocks, to the exception of the value of $\Omega_p = 2\pi 60$ kHz, and that the zeros of $s(t)$ are computed through a high order polynomial approximation. The spectrum of $q(t)$ generated from an input signal of frequencies 1 kHz, 3 kHz and 6 kHz, is presented in figure 13.

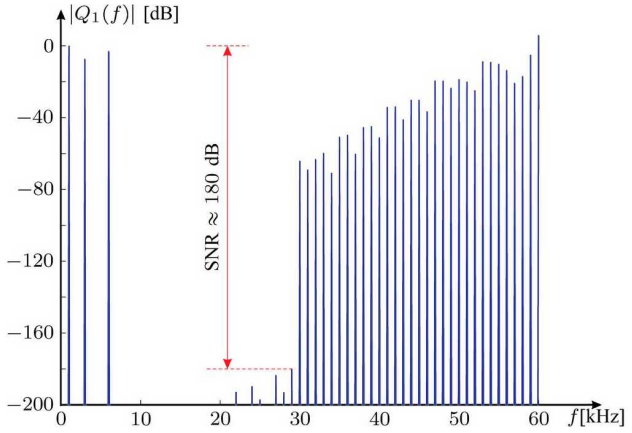

Figure 13: Frequency spectrum of $q_1(t)$ [12].

As to hardware implementations, in the available literature, two groups of authors developed two different projects. One group produced a real-time, hardware floating point implementation [11] in a DSP specific hardware description language similar to C programming. The Hilbert transformer, that is also an interpolator (see subsection IV-B), was implemented with a bandpass filter of 127 coefficients. No information about the AEM block or the zero calculation is provided, and the carrier frequency of the quadrature modulator is 24 kHz. The PWM output is produced with a resolution of 13 bit and the maximum signal bandwidth is 12 kHz. The resulting PWM wave spectrum for a 1 kHz input signal is presented in figure 14.

Another group presented two implementations [15] [14] with similar characteristics: a real-time algorithm, operands of fixed-point 24 bit and maximum input signal bandwidth of 20 kHz. The architecture of the two projects is also very similar, with the second project presenting better results. In both cases the Hilbert transform is implemented with a bandpass 51 coefficient filter, the AEM with an approximation of the exponential function through Look-Up Tables (10 bit in the first project, 12 bit in the second) and the LP filter after AEM with a 171 tap FIR. The carrier frequency of the quadrature modulator is 48.8 kHz and the zero finding

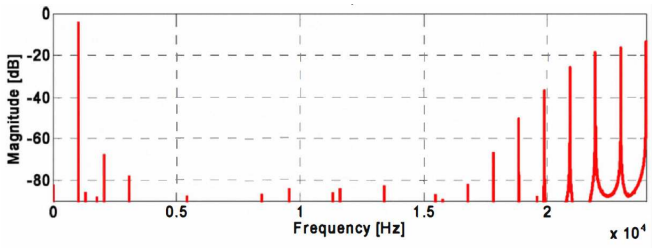

Figure 14: Frequency spectrum of $q(t)$ [11].

is obtained through the Newton method applied to a 3rd order polynomial interpolation, with a resolution of 11 bit in the first case and 10 bit in the second. In the first case, 5 DSPs, 1 FPGA and 1 upsampler (with factor 8) were used, whereas in the second project only 3 DSPs, 2 FPGAs and 1 upsampler with the same factor were used.

The spectrum of the PWM wave generated by the second project is presented in figure 15. Since no information is provided, the input signal is presumed to be the sum of two pure sinusoidals at the frequencies 18 kHz and 20 kHz.

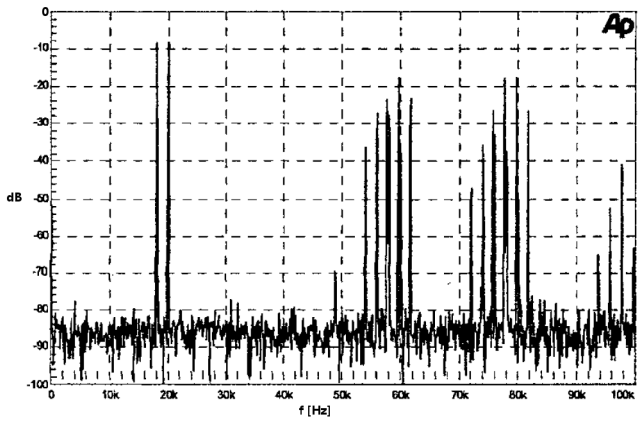

Figure 15: Frequency spectrum of $q(t)$ [11].

By inspection of figures 14 and 15 it is possible to conclude that both spectra present a guard band that separates the baseband from the high frequency components, one of the key characteristics of the CM. Regarding the quality of both spectra, in the latter quite a number of distortion components are visible, both harmonic and non-harmonic, whereas the former project resulting FFT presents a better spectral content. It is also interesting to note that the results are achieved, as foreseen in the theory, with a low switching rate.

B. MATLAB model

The work developed in this thesis was based in a real-time MATLAB model of CM with 64 bit floating point operands [1]. The scheme of this implementation is presented in figure 16.

The Hilbert block was designed as an 8 stage interpolator filter with 128 coefficients per stage. The AEM block was

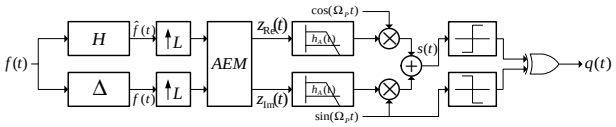

Figure 16: MATLAB implementation of CM [1].

implemented as an approximation of the exponential function through a Maclaurin series with 8 coefficients:

$$z(t) = e^{-jf_A(t)} = \sum_{k=0}^{7} \frac{(-jf_A(t))^k}{k!} = 1 - jf_A(t) - \frac{[f_A(t)]^2}{2!} + \ldots \tag{5}$$

As a result of this operation, the spectrum of $z(t)$ is increased by a factor of 8. Therefore, in order to avoid aliasing, the signal must be interpolated prior to the approximation of the exponential. That is the reason why the Hilbert transform FIR, that form the block immediately before the AEM are also interpolators. The low-pass filter after the Maclaurin approximation of $e^{-jf_A(t)}$ is also a FIR, with cut-off frequency of 22 kHz and an attenuation band starting at 24 kHz. This filter, as mentioned before, defines the extent of the guard band.

The quadrature modulator output is calculated through (6):

$$s(t) = Re\{z_f^*(t)e^{j\Omega_p t}\} = x_f(t)\cos(\Omega_p t) + y_f(t)\sin(\Omega_p t), \tag{6}$$

with $\Omega_p = 2\pi 24$ rad/s.

Finally, the zero crossing instants of $s(t)$ are computed through the bissection method, after a 3rd order polynomial evaluation around each root, bracketed through a change in the samples sign. The spectrum of the PWM wave generated by this model with a sinusoidal at 1 kHz as the input signal is presented in figure 17.

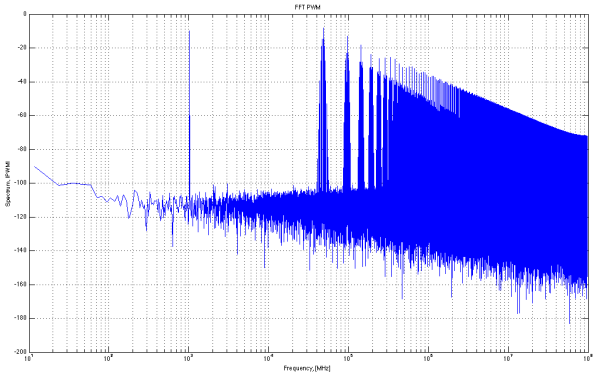

Figure 17: Frequency spectrum of $q(t)$ with 1 kHz input[1].

The MATLAB model results are indicative of a good implementation of CM, since the spectrum of the PWM signal contains all the expected characteristics. The seeming bad

performance for lower frequencies is due to the logarithmic scale of the frequency axis. Since a uniform number of points per frequency interval is used in the making of the FFT signal, as we go through the frequency axis from higher to lower frequencies, the number of points used for the FFT line presented decreases exponentially, thus generating a distorted figure.

V. Proposed Implementation

The realization of a real-time 16 bit fixed point implementation of CM in C programming language has a structure similar to that of the MATLAB model, as it can be seen in figure 18.

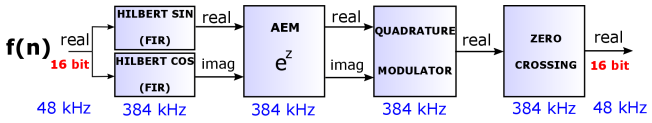

Figure 18: 16 bit fixed point scheme of CM implementation.

The main goal of this model compared to the others, particularly those with floating point operators, is to implement the CM algorithm with division only by integer powers of 2, to prevent long and costly hardware operations. Additionally, care must be taken when adjusting the formats of each operation's results, for the possibility of underflow or overflow exists.

A. Hilbert Transform and Interpolation

The scheme of this block is shown in figure 18.

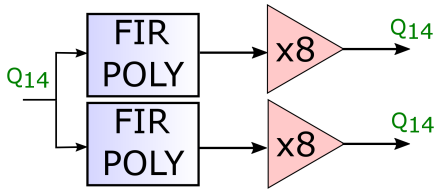

Figure 19: Hilbert Transform and Interpolation block scheme.

First, the input signal is divided between the two blocks that implement the Hilbert Transform and the interpolation, which are implemented as a polyphase FIR filter with 8 stages, each having 128 coefficients. As a result of the interpolation, the signal's power is divided by 8 and, as such, to compensate that energy loss, the final stage of this block is a 3 bit arithmetic shift left to perform a multiplication by 8. The input signal amplitude can oscillate between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$, thus the adequate fixed point format to represent it is Q_{14}. As the coefficients of the Hilbert Transform filter are in Q_{15}, the final result format is Q_{14}.

B. Analytical Exponential modulator and Filtering

The scheme of this block is shown in figure 18.

The approximation of the complex exponential is grouped as follows:

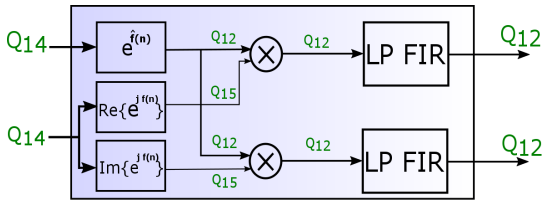

Figure 20: AEM and filtering block scheme.

$$z^*(n) = \left\{e^{-jf_A(n)}\right\}^* = \left\{e^{-j[f(n)+j\hat{f}(n)]}\right\}^* = \\ = \left\{e^{\hat{f}(n)-jf(n)}\right\}^* = e^{\hat{f}(n)+jf(n)} = e^{\hat{f}(n)} \cdot e^{jf(n)} = \\ = \underbrace{e^{\hat{f}(n)}}_{e_1} \cdot \underbrace{\left(Re\{e^{jf(n)}\} + jIm\{e^{jf(n)}\}\right)}_{e_2}. \quad (7)$$

The value of e_1 is obtained through a Maclaurin approximation with 8 coefficients grouped in Horner scheme:

$$e^{\hat{f}(n)} \approx 1 + \hat{f}(n) + \left[\frac{1}{2!} + \frac{1}{3!}\hat{f}(n)\right][\hat{f}(n)]^2 + \\ \left[\left(\frac{1}{4!} + \frac{1}{5!}\hat{f}(n)\right) + \left(\frac{1}{6!} + \frac{1}{7!}\hat{f}(n)\right)[\hat{f}(n)]^2\right][\hat{f}(n)]^4. \quad (8)$$

The calculation of e_2 has the same approach, but given the fact that the argument is complex (and so is its result), this block is split into real and imaginary coefficient calculation, which leads to

$$e^{jf(n)} = \underbrace{1 - \frac{[f(n)]^2}{2!} + \frac{[f(n)]^4}{4!} - \frac{[f(n)]^6}{6!}}_{Re\{e^{jf(n)}\}=\cos f(n)} + \\ + j\underbrace{\left(f(n) - \frac{[f(n)]^3}{3!} + \frac{[f(n)]^5}{5!} - \frac{[f(n)]^7}{7!}\right)}_{Im\{e^{jf(n)}\}=\sin f(n)}. \quad (9)$$

After the multiplication of e_1 for the real and imaginary part of e_2, both results are filtered with a low-pass FIR filter whose cutoff frequency of $\approx$ 22 kHz will define the corner frequency of the guard band.

The input signals of this block are in Q_{14}. As such, for the real exponential the correct fixed point format is Q_{12} ($\approx [0.2; 4.81]$) and for both parts of the complex exponential Q_{15} [$\sin(\cdot)$ and $\cos(\cdot)$]. Therefore, the multiplication of e_1 by e_2 should be stored as Q_{12}, and the same format is suitable for both outputs, since the absolute value of all filter coefficients does not exceed 1.

C. Quadrature Modulator

The scheme of this block is shown in figure 21. It consists of 2 multipliers, 16 samples of a sin signal period, with

frequency $f_p = 24$ kHz (sampled at a rate of 384 kHz) and an adder. The cos signal is obtained from the same table with a different index, starting 4 positions ahead of sin' s index.

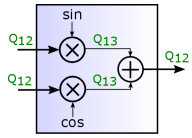

Figure 21: Quadrature modulator block scheme.

The sin samples are between -1 and 1 (Q_{15}), and the input samples are in Q_{12}. As they multiply by the trigonometric functions, its energy is halved. As such, when summed to form the real $s(n)$ signal, the adequate format is Q_{12}.

D. Zero Crossing

The scheme of this block is shown in figure 22.

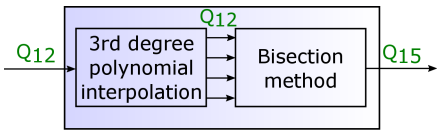

Figure 22: Zero Crossing block scheme.

When the samples of $s(n)$ arrive at this block, they are scanned for a sign change in groups of four. As a change is detected, four consecutive samples, $s_{-1} = s(n_{-1})$, $s_0 = s(n_0)$, $s_1 = s(n_1)$ and $s_2 = s(n_2)$, two positive and two negative, are used to produce a 3rd order polynomial through Lagrange Polynomial Interpolation, which serves the purpose of providing intermediate values for the approximation of the $s(n)$ signal between samples. As this CM algorithm is intended to be implemented in real-time, the PWM pulse former does not need to provided a specific absolute time instant directly related to the input signal itself. It needs to be fed with a value (16 bit in this case) that corresponds to a PWM wave duty cycle with a certain switching frequency. That switching frequency is given by the sin function of the quadrature modulator, and corresponds to $2f_p$, since there are 2 instants by period in which the sin function takes the value 0. These instants define the leading or the trailing edge of the PWM wave, whether we are dealing with a Leading-Edge or a Trailing-Edge PWM wave. The other instants, the roots of $s(n)$, define the trailing-edge on the former case (or the leading edge, in the latter). As the interval between samples is constant, one may stipulate $n_{-1} = -1$, $n_0 = 0$, $n_1 = 1$ and $n_2 = 2$, since each root of $s(n)$ is relative to the $\sin(\Omega_p n)$ root immediately before, and each one's calculation is independent of any other. Consequently, one may determine

$$p(x) = \sum_{k=-1}^{2} s_k l'_k(x) = s_{-1}l'_{-1} + s_0 l'_0(x) + s_1 l'_1(x) + s_2 l'_2(x), \tag{10}$$

with each l_k being

$$l'_{-1}(x) = \frac{(x-n_0)(x-n_1)(x-n_2)}{(x_{-1}-n_0)(x_{-1}-n_1)(x_{-1}-n_2)} = \\ = K_{-1}\underbrace{(x-1)(x^2-2x)}_{l_{-1}(x)}, K_{-1} = -\frac{1}{6}, \tag{11a}$$

$$l'_0(x) = \frac{(x-n_{-1})(x-n_1)(x-n_2)}{(x_0-n_{-1})(x_0-n_1)(x_0-n_2)} = \\ = K_0\underbrace{(x+1)(x-1)(x-2)}_{l_0(x)}, K_0 = \frac{1}{2}, \tag{11b}$$

$$l'_1(x) = \frac{(x-n_{-1})(x-n_0)(x-n_2)}{(x_1-n_{-1})(x_1-n_0)(x_1-n_2)} = \\ = K_1\underbrace{(x+1)(x^2+2x)}_{l_1(x)}, K_1 = -\frac{1}{2}, \tag{11c}$$

$$l'_2(x) = \frac{(x-n_{-1})(x-n_0)(x-n_1)}{(x_2-n_{-1})(x_2-n_0)(x_2-n_0)} = \\ = K_2\underbrace{x(x^2-1)}_{l_2(x)}, K_2 = \frac{1}{6}, \tag{11d}$$

later multiplied by the respective samples according with (10) to formulate the desirable approximation.

The bisection method is used afterwards to find the relative instant (between 0 and 1, Q_{15}) at which the zero crossing occurs. Although this method has a relatively low convergence order (linear), it was chosen because it is the only one (from the most common root-finding algorithms) that only has a division by 2, unlike many others that have division in which the two operands are unknown, namely the Secant, Newton, Ridders, Muller and Brent-Dekker methods [9]. Additionally, in the bisection method, convergence towards the root is always guaranteed, with each step gaining at least 1 bit of resolution, thus limiting the number of iterations to a maximum of 16 per zero crossing detection. As a result of this block, when all the $s(n)$ samples are through it, a set of duty cycles were produced that fully represent the PWM type signal $q(t)$ generated by CM.

VI. Results

A number of spectra were produced to demonstrate the correct implementation of the algorithm. In figure 23 is presented the spectrum of a PWM wave generated by the C model and in figure 24 is presented the spectrum of a PWM wave generated by the MATLAB model for comparison. Both have as input a 1 kHz sine wave.

As observed in previous figures, a clear separation between the baseband ($\approx$ 22 kHz) and the high frequency components ($\geq$ 24 kHz) is clearly visible. This key feature of CM allows a relaxed specification of the demodulating low-pass filter.

The -120 dB noise floor for both MATLAB and C model suggests that the accuracy of the zero finding method is key in the final resolution of the PWM signal. Although they are not

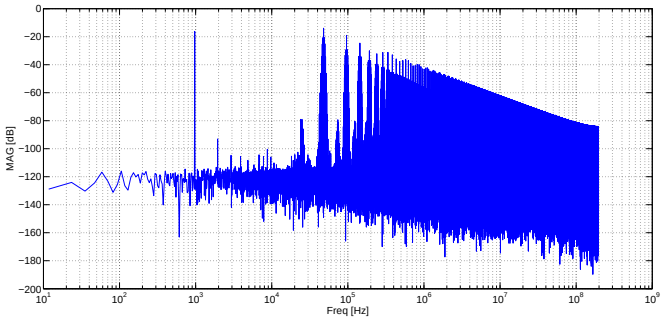

Figure 23: Spectrum of a PWM wave (C model, 1 kHz input).

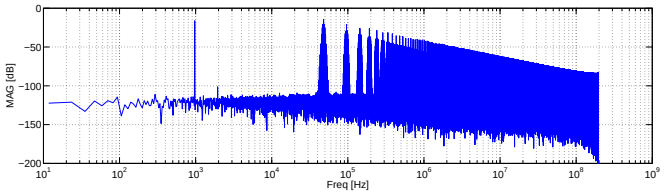

Figure 24: Spectrum of a PWM wave (MATLAB model, 1 kHz input).

shown in this article, throughout the modulation chain the noise floor of the MATLAB model was consistently lower than that of the C model due to the resolution of the operands: floating point double precision in the latter as opposed to fixed point 16 bit in the former. However, the zero crossing instants produced by both models have a 16 bit resolution, and as it can be seen by the analysis of table I, the C model presents better results than those of MATLAB model.

Additionally, its low switching frequency increases the efficiency of the output power stage, since the great majority of class D losses correspond to simultaneous conducting devices caused by high switching rates.

A few more results were produced, with both C and MATLAB models, with different input signals. Figures 25 and 26 have as input signal a 10 kHz sin and figures 27 and 28 have as input a multi-tone signal with 1 and 2 kHz.

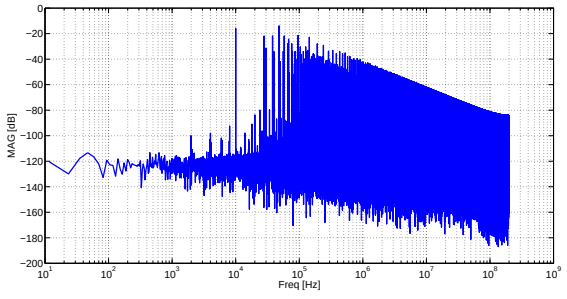

Figure 25: Spectrum of a PWM wave (C model, 10 kHz input).

A closer analysis to the previous spectra reveals several extra-signal components above the -100 dB noise floor, namely

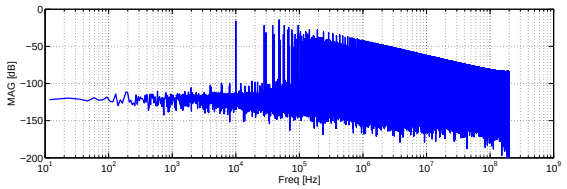

Figure 26: Spectrum of a PWM wave (MATLAB model, 10 kHz input).

at 2, 4, 8 and 16 kHz. These are result of aliased inter-modulation products between the input signal and the carrier. However, as shown in table I, its effect in the quality of signal is almost negligible.

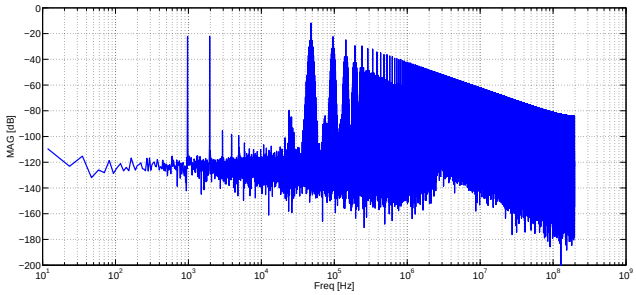

Figure 27: Spectrum of a PWM wave (C model, 1+2 kHz input).

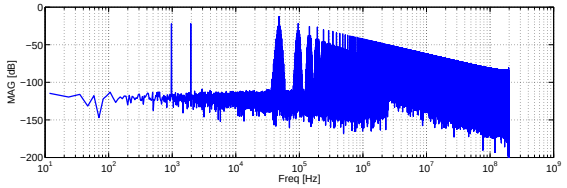

Figure 28: Spectrum of a PWM wave (MATLAB model, 1+2 kHz input).

In the case of multi-tone signals, both in MATLAB and C models, a few harmonics appear slightly above the -100 dB level. As it is with the case of the 10 kHz input signal, these components appear as a result of non-harmonic distortion previously mentioned.

	SNR [dB]		THD [dB]	
	MATLAB	C	MATLAB	C
1 kHz	104.00	104.90	-95.49	$-86,12$
10 kHz	103.51	105.18	-82.13	-67.73
1+2 kHz	104.00	104.90	-95.15	-93.33

Table I: SNR and THD from C and MATLAB models.

From the analysis of the values present in table I, it is possible to conclude that the C model designed to implement CM is proved to work correctly and has very good results, since a magnitude in the order of 100 dB for SNR is desired

for audio applications. Although lower in absolute value, THD evaluation confirms a good algorithm performance.

VII. Conclusion

A proposed method for a real-time 16 bit fixed point CM implementation was described in this paper. Its results, namely the SNR and THD, confirm that the developed algorithm is suitable for high-definition audio applications, marking a step towards the realization of CM in integrated circuit. A series of PWM signals were produced that match the spectral characteristics of CM.

One of the key aspects of the developed algorithm is the capability of performing advanced and complex mathematical operations, such as interpolation and root-finding methods, without requiring the use of a division operation with 2 unknown operands. In addition, another complex operations, such as Hilbert Transform, were implemented through the use of much used and stable structures, such as FIR filters. This feature allows for much faster hardware performance.

Although C language is not a hardware description language, its similarities with both MATLAB (software) and Verilog (hardware description) provide a versatile, generic, binary operator algorithm for CM implementation, capable of being directly translated to either Verilog, VHDL, other hardware description language, or even used directly by a DSP or an FPGA capable or decoding fixed point C.

References

[1] Tiago Domingues, Marcelino Santos, and Goncalo Tavares. Low frequency pwm modulation for high efficiency class-d audio driving. In *Design of Circuits and Integrated Circuits (DCIS), 2014 Conference on*, pages 1–5. IEEE, 2014.

[2] Tiago Domingues, Marcelino Santos, and Goncalo Tavares. A click modulation audio player. *Design of Circuits and Integrated Circuits (DCIS), 2015 Conference on*, November 2015.

[3] B. F. Logan. Information in the zero crossings of bandpass signals. *Bell System Technical Journal*, 56(4):487–510, 1977.

[4] B. F. Logan. Click modulation. *AT&T Bell Laboratories Technical Journal*, 63(3):401–423, 1984.

[5] B. F. Logan. Signals designed for recovery after clippingi. localization of infinite products. *AT&T Bell Laboratories Technical Journal*, 63(2):261–285, 1984.

[6] B. F. Logan. Signals designed for recovery after clippingii. fourier transform theory of recovery. *AT&T Bell Laboratories Technical Journal*, 63(2):287–306, 1984.

[7] B. F. Logan. Signals designed for recovery after clippingiii: Generalizations. *AT&T Bell Laboratories Technical Journal*, 63(3):379–399, 1984.

[8] Alejandro Oliva, Eduardo Paolini, and Simon S. Ang. A new audio file format for low-cost, high-fidelity, portable digital audio amplifiers. 2005.

[9] William H Press, Saul A Teukolsky, William T Vetterling, and Brian P Flannery. *Numerical recipes in C*, volume 2. Cambridge university press Cambridge, 1996.

[10] Adel S. Sedra and Kenneth C. Smith. *Microelectronic Circuits International Student Edition*, chapter 14. Oxford University Press, Inc., 5th edition, 2004.

[11] Krzysztof Piotr Sozański. A digital click modulator for a class-d audio power amplifier. *Elektronika: konstrukcje, technologie, zastosowania*, 51(3):84–88, 2010.

[12] Leandro Stefanazzi, Alejandro R Oliva, and Eduardo E Paolini. Alias-free digital click modulator. *Industrial Informatics, IEEE Transactions*, 9(2):1074–1083, 2013.

[13] Leandro Stefanazzi, Eduardo Paolini, and Alejandro Oliva. Click modulation: an off-line implementation. In *Circuits and Systems, 2008. MWSCAS 2008. 51st Midwest Symposium on*, pages 946–949. IEEE, 2008.

[14] M. Streitenberger, F. Felgenhauer, H. Bresch, and W. Mathis. Class-d audio amplifiers with separated baseband for low-power mobile applications. In *Circuits and Systems for Communications, 2002. Proceedings. ICCSC '02. 1st IEEE International Conference on*, pages 186–189, 2002.

[15] M. Streitenberger and W. Mathis. A novel coding topology for digital class-d audio power amplifiers with very low pulse-repetition rate. In *Solid-State Circuits Conference, 2002. ESSCIRC 2002. Proceedings of the 28th European*, pages 515–518, Sept 2002.