



**HAL**  
open science

# A study of companding approaches to alleviate the ADC constraints when demodulating multiple signals

Mathieu Vallérian, Florin Hutu, Guillaume Villemaud, Benoît Miscopain,  
Tanguy Risset

► **To cite this version:**

Mathieu Vallérian, Florin Hutu, Guillaume Villemaud, Benoît Miscopain, Tanguy Risset. A study of companding approaches to alleviate the ADC constraints when demodulating multiple signals. [Research Report] RR-8683, Inria - Research Centre Grenoble – Rhône-Alpes; INRIA. 2015, pp.21. hal-01117302

**HAL Id: hal-01117302**

**<https://inria.hal.science/hal-01117302>**

Submitted on 16 Feb 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# A study of companding approaches to alleviate the ADC constraints when demodulating multiple signals

Mathieu Vallérian, Florin Hutu, Guillaume Villemaud, Benoît Miscopein, Tanguy Risset

**RESEARCH  
REPORT**

**N° 8683**

February 2015

Project-Team Socrate





## A study of companding approaches to alleviate the ADC constraints when demodulating multiple signals

Mathieu Vallérian<sup>\*†</sup>, Florin Hutu<sup>†</sup>, Guillaume Villemaud<sup>†</sup>, Benoît Miscopein<sup>\*</sup>, Tanguy Risset<sup>†</sup>

Project-Team Socrate

Research Report n° 8683 — February 2015 — 21 pages

**Abstract:** In urban sensor networks, the diversity of propagation conditions can lead to the simultaneous reception of signals having very different power levels. Given the diversity of wireless technologies used in this area, implementing gateways using a Software-Defined Radio (SDR) approach seems to be a very practical solution. Overcoming the high dynamic range may however require an Analog-to-Digital Converter (ADC) with a resolution fine enough to digitize the weakest signal with a satisfying precision. In order to relax these specifications, an analog processing, such as companding, should be performed before digitization. The companding technique is usually employed on one signal (and not on multiple signals that are only separated on the frequency domain). This report studies three companding laws to test their efficiency in relaxing the digitization constraints with multiple signals. A  $\mu$ -law, a Piecewise-Linear (PL) law and a Piecewise-Linear, Constant Gain with Offsets (PLCGO) law are tested. It is described how to use a PLCGO approach to reduce ADC's complexity, and two implementations of the compressing law are proposed.

**Key-words:** RF front end, analog processing, multiplicative companding, additive companding, multiple signals digitization, dynamic range reduction

---

<sup>\*</sup> Orange-Labs, 38243, Meylan, France

<sup>†</sup> Université de Lyon, INRIA, INSA-Lyon, CITI-INRIA, F-69621, Villeurbanne, France

RESEARCH CENTRE  
GRENOBLE – RHÔNE-ALPES

Inovallée  
655 avenue de l'Europe Montbonnot  
38334 Saint Ismier Cedex

## **Une étude de lois de compression-extension pour relâcher les contraintes sur le CAN dans le cas d'une démodulation de signaux multiples**

**Résumé :** Dans les réseaux de capteurs urbains, la diversité des conditions de propagation des signaux peut mener à une réception de signaux multiples, avec des niveaux de puissance très différents. De nombreuses technologies de communication sont utilisées dans ce domaine. Sachant cela, la solution la plus adaptée pour implémenter la passerelle collectant les données des nœuds semble être d'utiliser une radio logicielle. Pour pouvoir absorber la grande plage dynamique des signaux, le Convertisseur Analogique-Numérique (CAN) requiert une résolution suffisamment fine pour numériser le signal le plus faible avec une précision satisfaisante. Afin de relâcher cette contrainte, un traitement analogique, comme la compression-extension, doit être appliqué sur les signaux avant numérisation. Habituellement la technique de compression-extension est utilisée sur un seul signal, et pas sur des signaux multiples (qui ne sont séparés que dans le domaine fréquentiel). Ce rapport étudie donc trois lois de compression-extension afin de tester leur capacité à abaisser les contraintes de numérisation de signaux multiples. Une loi en  $\mu$ , une loi linéaire par morceaux, et une loi linéaire par morceaux, avec gain constant et niveaux de tension continue sont testées. Il est décrit dans ce rapport comment une approche avec cette dernière loi permet de réduire la complexité du CAN. Deux implémentations de la loi de compression sont proposées.

**Mots-clés :** systèmes RF, traitement analogique, compression-extension multiplicative, compression-extension additive, numérisation de signaux multiples, réduction de plage dynamique

## 1 Introduction

An urban sensor network is generally built on sensor nodes that collect data from various applications, such as parking management, water and gas metering, pollution monitoring, public lighting, etc. A sink node (also called gateway) collects the data received from the sensors. The sensors are Short Range Devices (SRD) in many actual deployments. Such devices generally emit in the 868 MHz band with a 8 MHz frequency band [1]. Each application having its own constraints, several communication protocols are used to handle them.

These communication protocols are still evolving today, and new ones are susceptible to emerge in a near future. To embed these protocols in the gateway, the existing solution consists in a stack-up of radio receivers, each one dedicated to one communication protocol. This method has a major drawback: if a new technology has to be added to the network, the gateway must be replaced. This problem can be avoided by using a reconfigurable gateway: then the gateway could be able to embed a new technology with a simple software update. As done in Software-Defined Radio (SDR), the transmission frequency band should be digitized at once [2].

Nevertheless another problem appears from this choice. From the diversity of constraints, propagation conditions and signals' features that originate from the diversity of applications, received signals with high dynamic ranges can be encountered. It has been shown in [3] that the dynamic range between two received signals can be up to 100 dB. This high dynamic range is an important constraint from the Analog-to-Digital Converter (ADC) perspective. Indeed the greater the dynamic range is, the higher should be the ADC resolution to properly digitize every signals. The expression of the required ADC resolution to demodulate all the signals in the band with such a dynamic range has been expressed [3], and is plotted in Figure 1. With a 100 dB dynamic range, the required resolution (highlighted with dashed lines on Figure 1) is 21 bits.

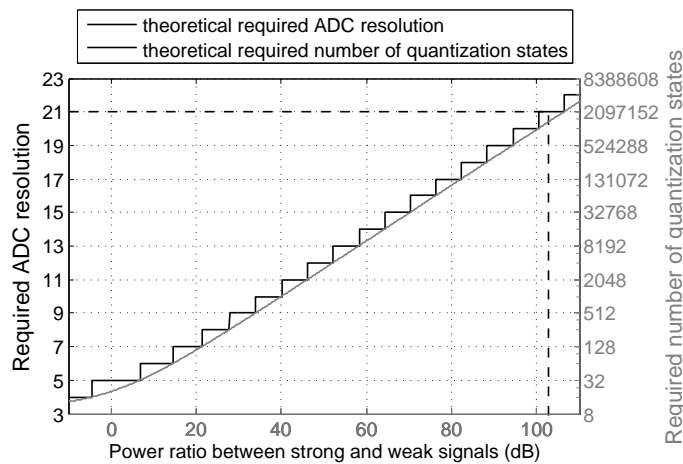


Figure 1: Required resolution regarding the power ratio between the strong and the weak signal

Reminding that a band of 8 MHz must be digitized, a 21 bits resolution appears to be unfeasible with today's ADCs [4]. This presents the ADCs performance evolution to 2010. On figure 2, the evolution of ENOB is given. The ADCs with a sampling frequency of the order of 10 MS/s (grey line) had an ENOB of about 14 bits in 2010. Extrapolating the grey line, an ADC with an ENOB of 16 bits can't be expected in a near future. Then another solution has to be found to digitize the signals. Digitizing sub-bands in parallel would be problematic if signals straddle two sub-bands. Moreover, some communication

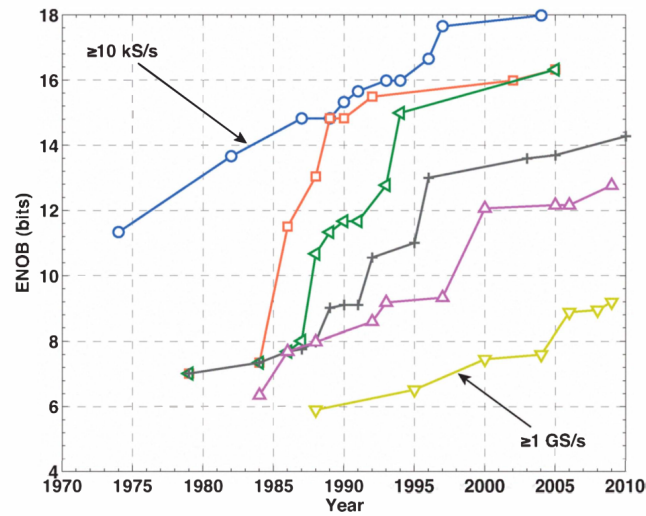


Figure 2: ADC performance evolution, from [4]. The sampling rates are 10 kS/s, 100kS/s, 1MS/s, 10MS/s, 100MS/s, and 1 GS/s. The grey line represents the 10 MS/s ADCs. By extrapolation, no ENOB higher than 16 bits can be expected in a near future.

protocols use spread spectrum techniques, so the signal could be digitized in several sub-bands: then it would be disrupted after reconstitution. Another way to reduce the required resolution is to apply on the signals an analog processing, such as companding, in order to reduce their dynamic range.

The goal of this work is to explore how companding laws are able to improve or not the required resolution to properly digitize the multiple signals present on the band. Three companding laws are studied: the  $\mu$ -law [5], a Piecewise-Linear (PL) law [6] and a Piecewise-Linear, Constant Gain law with Offsets (PLCGO) [7]. It is shown that whereas the  $\mu$ -law brings no improvement with multiple signals with a high dynamic range, the PL law shows promising results. The PLCGO law is similar to the PL law although offsets are added to the signal to reduce its dynamic range instead of changing the compressing gain. This feature allows to reach a far better digitization performance than the PL law.

The section 2 introduces the companding technique and gives common examples of companding laws. Three companding laws are also introduced, and are studied latter in the report. The  $\mu$ -law is described and studied in section 3. The Bit Error Rate (BER) of a weak signal in presence of a stronger signal and the quantization noise are used to quantify the digitization quality, which is compared with and without companding. The section 4 is dedicated to the study of PL law, which is described there. The digitization performance is still quantified by the BER of a weak signal and by the quantization noise. The section 5 introduces the PLCGO law. Its efficiency to reduce the required resolution with multiple signals is studied, and its performance in terms of BER and of quantization noise is compared with the PL law and  $\mu$ -law. The relevance of the PLCGO law in our case being proved, the section 6 proposes two implementations for this law, and discusses their performance. The section 7 gives the conclusions of this work and gives some directions for future work.

## 2 The companding technique

Companding is a signal processing technique aiming at reducing the dynamic range of the input signal [5]. It has been patented in 1928 for the transmission of analog pictures [8] and the technique had been

widely used in analog telephony. Another application is the transmission of speech messages in digital wireless systems like cellular telephony. The technique is widely described in [9], where Tsividis studies externally-linear, time-invariant systems and demonstrates their interest in companding implementation by improving the dynamic range and power dissipation. In the case of digitization, the signal is first compressed to reduce its dynamic range, the compressed signal is then digitized and eventually expanded to recover the original signal. The expansion is performed by means of the expanding function, which is the compression reciprocal function.

Comanding can be instantaneous or syllabic. With instantaneous companding, the compressing gain varies with the instantaneous amplitude value of the input signal [10]. With syllabic companding, the input signal is divided in small time segments called *syllables* [9]. The measurement of a mean (*e.g.* Root Mean Square: RMS) amplitude value on the syllable is used to set up the compressing gain. In both cases the compressing gain depends on the input signal amplitude, so the compressing and expanding functions are always non-linear.

The companding law can follow various functions. Some commonly used functions are the logarithmic function ( $\log$ ), the hyperbolic tangent ( $\tanh$ ) function, the hyperbolic sinus function ( $\sinh$ ) or the square root function ( $\sqrt{\cdot}$ ). They are usually implemented through  $\log$ ,  $\tanh$ ,  $\sinh$  or  $\sqrt{\cdot}$ -domain filters. Examples of implementations of these laws through this way can be found in [11] (proposing a new implementation of the  $\sinh$  law and making simpler to analyze the circuit imperfections), in [12] (proposing a  $\tanh$  implementation for syllabic companding in order to reduce power dissipation) or in [13] (whose primary goal is to demonstrate that an external non-linear behaviour of logarithmic laws can be extended to  $\sqrt{\cdot}$  laws). These implementations are usually dedicated to syllabic companding, but [14] showed that instantaneous companding can be implemented through a translinear filter, using the analog floating point technique. The logarithmic function is the most commonly used, and is for example used in the well-known  $\mu$ -law [5]. The compression functions of these laws are similarly shaped, and have the same behavior when compressing a signal. Thus, in this report, only the logarithmic law is considered and results are extrapolated to other similarly shaped laws.

Another class of companding laws exists. The described above laws are all bijective functions. But the compressing law can be only surjective (not injective), *i.e.* the original signal can't be directly recovered from the compressed signal. Then one compressed signal level corresponds to several input signal levels. This input signal level is eventually recovered thanks to the state variable. An example of a surjective law where a state variable is used can be found in [6].

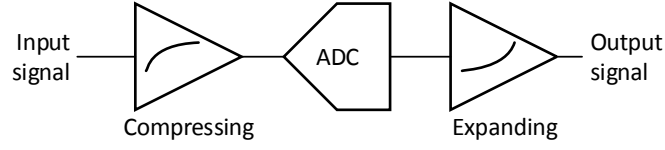
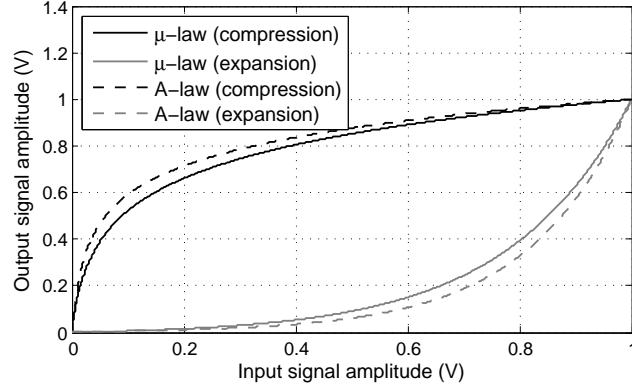
In this report, only instantaneous companding is considered and three compression laws are described, one being bijective and the other two being only surjective. The first one relies on a logarithmic function whereas the second one is built on a PL function. The third one is defined by a PLCGO function. It is important to note that companding is usually used to compress signals that are separated in the time domain. Given this report explores how to digitize two signals which occur simultaneously but separated in the frequency domain, the following sections show how these three techniques can meet this requirement or not.

## 3 Logarithmic compressing function

### 3.1 Introduction to the logarithmic compressing law

Applying a logarithmic gain on the received signal is a common means to get a logarithmic compression. This can be implemented with a logarithmic amplifier for example [5]. This kind of function was widely used in analog telephony, when the audio signal dynamic range is bigger than the one supported by the transmission protocol [5]. Another application is the transmission of speech messages in digital wireless systems like cellular telephony. In practice, the most widely used functions were the  $\mu$ -law in North



Figure 3: Block diagram of the  $\mu$ -lawFigure 4: Gains of the  $\mu$ -law and of the A-law with a compression factor of 100

America and the A-law in Europe. The Figure 3 presents the block diagram of companding using a logarithmic law.

Denoting the input signal, the compressed signal and the expanded signal by respectively  $x(t)$ ,  $w(t)$  and  $y(t)$ , and  $\mu$  and  $A$  being the respective compression factors of the  $\mu$  and A-laws, their compressing and expanding laws are defined respectively in (1) for the  $\mu$ -law and in (2) for the A-law.

$$\begin{aligned}
 w(t) &= \operatorname{sgn}(x(t)) \cdot \frac{\ln(1 + \mu|x(t)|)}{\ln(1 + \mu)} \quad (\text{Compression}) \\
 y(t) &= \operatorname{sgn}(w(t)) \frac{((1 + \mu)^{|w(t)|} - 1)}{\mu} \quad (\text{Expansion})
 \end{aligned} \tag{1}$$

$$\begin{aligned}
 w(t) &= \begin{cases} \operatorname{sgn}(x(t)) \cdot \frac{A|x(t)|}{1 + \ln(A)} & \text{if } |x(t)| < \frac{1}{A} \\ \operatorname{sgn}(x(t)) \cdot \frac{1 + \ln(A|x(t)|)}{1 + \ln(A)} & \text{if } \frac{1}{A} \leq |x(t)| \leq 1 \end{cases} \quad (\text{Compression}) \\
 y(t) &= \begin{cases} \operatorname{sgn}(w(t)) \cdot \frac{|w(t)|(1 + \ln(A))}{A} & \text{if } |w(t)| < \frac{1}{1 + \ln(A)} \\ \operatorname{sgn}(w(t)) \cdot \frac{\exp(|w(t)|(1 + \ln(A)) - 1)}{A} & \text{if } \frac{1}{1 + \ln(A)} \leq |w(t)| \leq 1 \end{cases} \quad (\text{Expansion})
 \end{aligned} \tag{2}$$

Both laws have the same behavior, as one can see on Figure 4. There the  $\mu$ -law and the A-law have been plotted with the same compression factor of 100. As the compressing and expanding functions are similarly shaped, only the  $\mu$ -law has been studied.

In [3], the required resolution to demodulate two simultaneous signals has been found with respect to their power ratio. This required resolution is shown on Figure 1. One can see that the required resolution to demodulate the signals with a 100 dB dynamic range is 21 bits. The goal of this section is to assess how this ADC requirement is impacted when a logarithmic compression law is used.

### 3.2 Simulation of the $\mu$ -law's performance

Using the Agilent ADS software, a strong and a weak Binary Phase Shift Keying (BPSK)-modulated signals (whose amplitudes are respectively denoted  $S_s$  and  $S_w$ ) are generated together with an Additive White Gaussian Noise (AWGN), both with a bitrate of 200 kbps. The resulting signal is compressed, digitized and then expanded using the  $\mu$ -law with a compression factor of 100. The digitization is performed with a sampling frequency of 16 MHz. After demodulation, a BER test is performed through the Monte-Carlo method (the simulation ends when the desired relative variance is met) to evaluate the companding impact and the digitization quality. Simulations are done by varying the ADC resolution and the power ratio between the strong and weak signals ( $S_s/S_w$ ), respectively from 1 to 25 bits and from 0 to 100 dB. The noise level is defined by the energy per bit to noise power spectral density ratio ( $E_b/N_0$ ) of the weakest signal. The lowest required resolution to demodulate the weakest signal only depends on  $S_s/S_w$  [3]. Since it does not depend on the weakest signal's  $E_b/N_0$ , this latter value has been arbitrarily set to 7 dB in all the simulations. As the  $S_s/S_w$  ratio can be up to 100 dB the BER test is performed only on the weak signal since it would take too much simulation time if done on the strongest one. Moreover, the Signal-to-Noise Ratio (SNR) of the strong signal (which is basically the sum of the weakest signal's SNR and  $S_s/S_w$ ) is high enough to guarantee a demodulation with a significantly low BER.

The Figure 5 shows the results of these simulations with a relative variance of 0.01. For a given value of  $S_s/S_w$ , the minimum required resolution is determined when the BER comes close to the  $10^{-3}$  floor, corresponding to the expected BER for a BPSK modulation when using an AWGN channel with  $E_b/N_0 = 7$  dB. The variation range of  $S_s/S_w$  has been chosen to be similar to the case of [3] (*i.e.* up to 100 dB).

When  $S_s/S_w$  is 0 dB, a positive effect on the BER is observed since it is lightly lessened. If  $S_s/S_w$  is a little higher, but remains low (here 20 dB) the BER is the same with and without companding. When the  $S_s/S_w$  ratio is getting higher, companding with a  $\mu$ -law leads to a BER degradation. The simulation results on Figure 5 are presented with a compressing factor  $\mu = 100$ , but other values have been tested, giving similar results.

The effect of the logarithmic companding on BER can be explained by considering the time-based representation of quantization noise. The quantization noise can be measured by performing the compression, digitization and expansion steps on a sine input signal. If the compression and expansion functions' implementations are the exact mathematical functions, the difference between the expanded and original signals is the quantization noise. This representation is given in Figure 6 and has been obtained through Matlab simulations. It shows the quantization noise of a 20-bit ADC obtained on a sinusoidal input signal (Figure 6a), using a  $\mu$ -law compression (Figure 6b) and a Direct Digitization (DD) (Figure 6c). When using a  $\mu$ -law compression, the quantization noise level depends on the input signal level: the higher the input signal, the higher the quantization noise. This effect is not observed with a DD.

When two signals are combined in time, the quantization noise with the  $\mu$ -law is driven by the strongest signal's amplitude. This issue is shown on Figure 7, where the equivalent quantization step is highlighted when using a DD and a  $\mu$ -law on two signals represented by sinusoids with different amplitudes and frequencies. The quantization step is higher when the input signal level is high, leading to a higher quantization noise. It means that when the input signal level is high the weakest signal's SNR may be severely affected, leading to a BER degradation. This effect is amplified as  $S_s/S_w$  is getting higher. When  $S_s/S_w = 0$  dB, the signals' amplitudes are equal: then the total signal amplitude is driven by both signals. The SNR then depends on the amplitude of both signals, leading to a BER improvement. All this

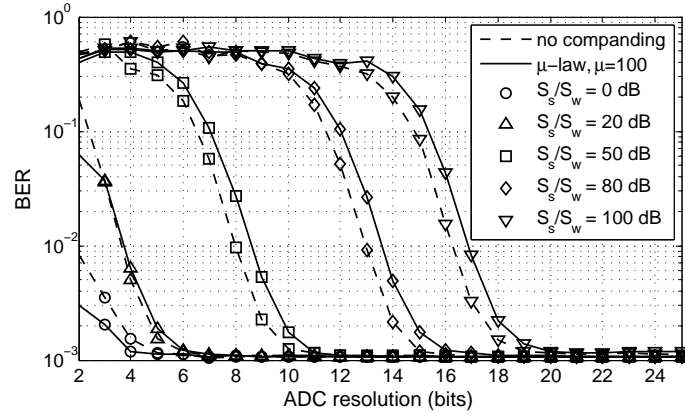


Figure 5: Simulated BER as a function of  $S_s/S_w$  and of the ADC resolution, with and without companding using a  $\mu$ -law

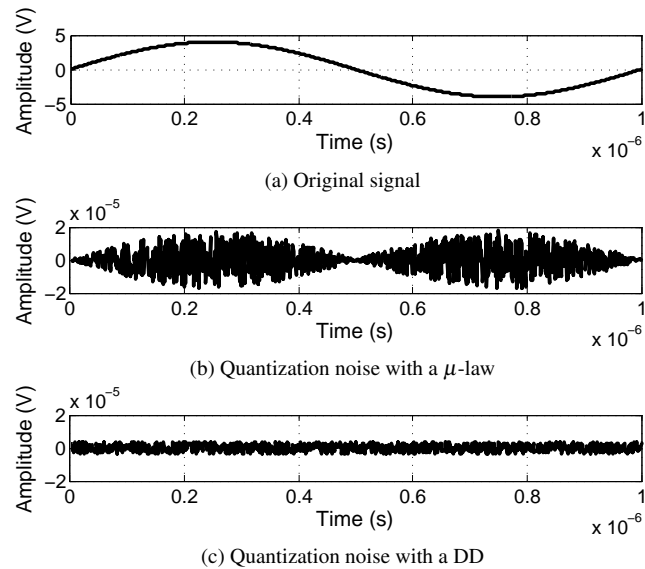


Figure 6: Quantization noise with a  $\mu$ -law, a PL law and a DD

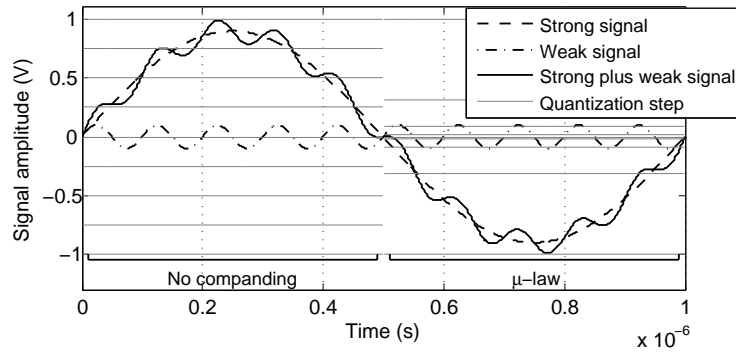
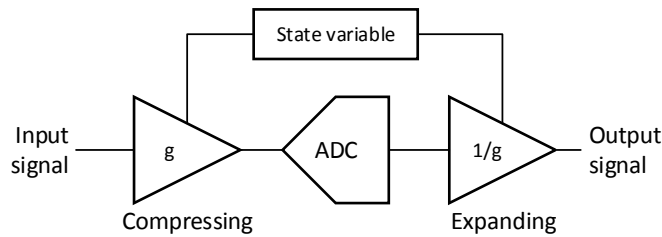

 Figure 7: Equivalent quantization states when using a DD and a  $\mu$ -law compression


Figure 8: Block diagram of the PL law

can be extrapolated to non-logarithmic, but similarly shaped functions, such as a tanh function.

The simulation results reported on Figure 6b show that companding using a logarithmic (or similarly shaped) law is not appropriate to reduce the dynamic range between two simultaneous signals, due to the aforementioned impact of the quantization noise. In the next section the PL compressing law is studied, as it may provide a different behavior depending on the compressing function shape.

## 4 PL compressing function

### 4.1 Introduction to the PL compressing function

With a PL compressing function, the gain is still dependent on the input signal amplitude; nevertheless it remains constant in predefined amplitude ranges. These amplitude ranges, or functioning states, are delimited by voltage thresholds. Thus within one state the compression law is linear irrespective to the input signal level, but overall the law is still non-linear (*i.e.* the compressing gain varies with time). The block diagram is given in Figure 8. With  $x(t)$ ,  $w(t)$ ,  $g(t)$  and  $y(t)$  the input signal, the compressed signal, the compressing gain and the expanded signal, the compressing law can be expressed as:

$$\begin{aligned} w(t) &= g(t) \cdot x(t) && \text{(Compressing)} \\ y(t) &= \frac{w(t)}{g(t)} && \text{(Expanding)} \end{aligned} \quad (3)$$

To expand the signal the inverse gains are applied to the compressed signal. For that, the state must be transmitted from compression to expansion through a dedicated state variable. An example of such a law can be seen on Figure 9, where three states (0, 1 and 2) are defined with the related thresholds  $V_{th1}$

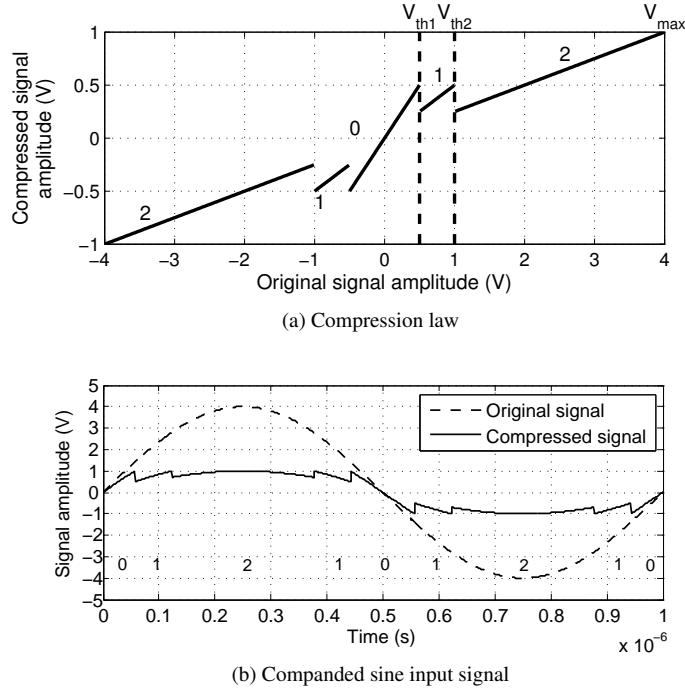


Figure 9: Example of a PL companding technique

and  $V_{th2}$ . The gain of the law is plotted in Figure 9a and the compression step is represented on Figure 9b with a sinusoidal input for illustration. The law example comes from [6], where an implementation using switched capacitor filters is presented. The aim of this work is to propose an implementation that can coexist with digital blocks on die, allowing to reduce the ADC power consumption. One can note that the dynamic range is effectively compressed; however the compressed signal amplitude range is not uniform from one state to another, due to the different gains applied on the input signal.

The goal of the following is to demonstrate how the ADC resolution requirements are affected when a PL compressing law is used.

## 4.2 Simulations of the law's performance

As in section 3, the BER of a weak signal is simulated in presence of a stronger signal using the Agilent ADS software. The only differences with the simulation schematic described at the beginning of the subsection 3.2 are the compressing and the expanding laws, which are cosimulated using the MatLab software. The laws were simulated using three states and as previously, the BER of the weak signal is simulated regarding the ADC resolution and  $S_s/S_w$ .  $E_b/N_0$  is still 7 dB, leading to the same BER floor of  $10^{-3}$ .

The simulation results are shown on Figure 10, still with a relative variance of 0.01. Contrary to the  $\mu$ -law, the PL law has always a positive effect on the BER by lessening it regardless of the  $S_s/S_w$  value. The benefit in terms of quantization bits is still dependent on  $S_s/S_w$  for the same reason as previously: the total signal amplitude being driven by the strong signal amplitude, the quantization noise do likewise.

This effect can be seen on Figure 11 where the quantization noise is given assuming a sinusoidal signal as an input signal (Figure 11a) for a  $\mu$ -law compression (Figure 11b), a PL law compression (Figure 11c) and a DD (Figure 11d). The same ADC features as in section 3 were used. The PL law's performance

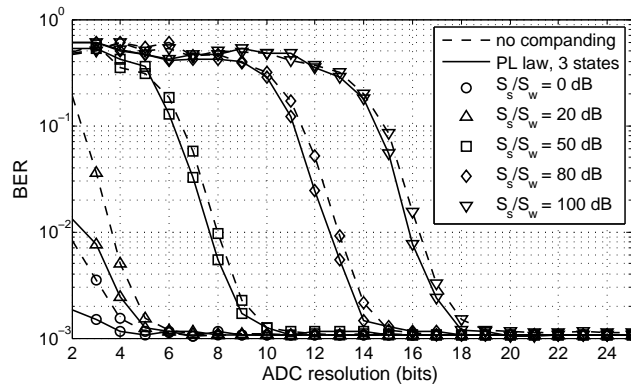


Figure 10: Simulated BER as a function of  $S_s/S_w$  and of the ADC resolution, with and without companding using a PL law

should be compared with the one obtained without companding and with a logarithmic compression function.

The analysis of Figure 11 indicates that PL compression seems a better way to reduce the required ADC resolution to digitize both low and strong levels signals than logarithmic compression. Indeed it can be easily shown that the quantization noise with a PL law is at worst equivalent to the one obtained with DD: when the signal is in the higher state (in the state 2 in our example) the compressing gain is 1 and then the compressed signal is equal to the input signal. In the other states the gain is higher and the input signal is amplified. The quantization noise is then drastically reduced in comparison with the  $\mu$ -law case: it is at its maximum close to the one obtained with a DD, whereas with the  $\mu$ -law it was higher at its maximum than the one with a DD.

The Figure 12 shows the equivalent quantization step when using a PL compressing law. More quantization states are used with the PL law than with DD, thanks to the use of a state variable. When the input signal is low, more quantization states are used, leading to a lower quantization noise. When the input signal is high, the same number of quantization states as with DD are used. When comparing with the  $\mu$ -law's performance, less quantization states are used when the signal is low, but more quantization states are used when the signal is high. As the quantization noise was mainly present when the signal was high with the  $\mu$ -law, the quantization noise is globally reduced.

The fact that the quantization noise with the PL law is globally lower than the one with a DD proves that the PL law improves the digitization quality with regard to the DD. However, it can be seen on Figure 9b that most of time only a part of the ADC's dynamic range is occupied by the compressed signal. Especially in the states 1 and 2 the signal occupies a small part of the total ADC's dynamic range. The next section presents the PLCGO law, that maximizes the signal's dynamic range occupancy.

## 5 Compression law using offsets

### 5.1 Motivation

The general idea of the additive companding law is to maximize the dynamic range occupied by the signal after compression (*i.e.* before digitization) as an improvement of the PL law. A way to get this improvement is to add offsets on the input signal to center it around the zero level and to set the compressing gain in order to maximize the signal occupancy of the ADC's dynamic range. To do this, voltage thresholds are

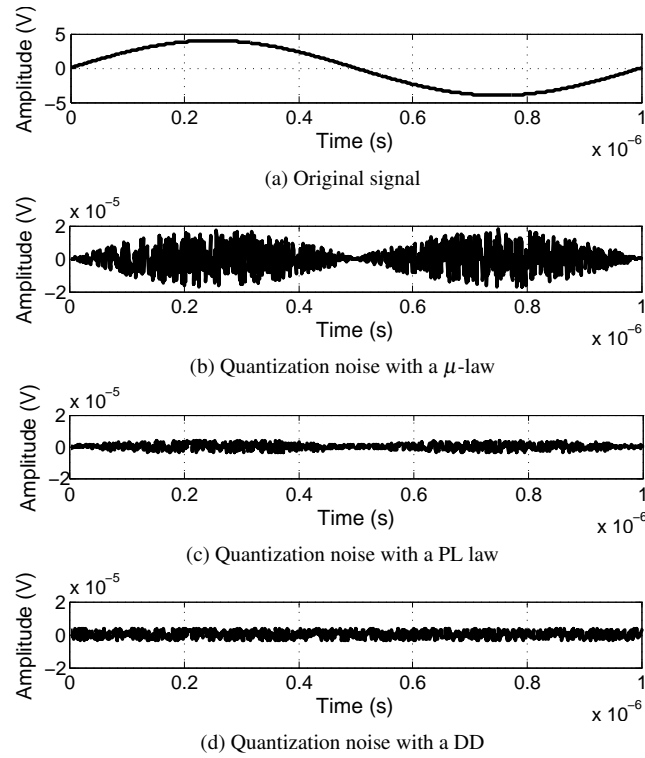


Figure 11: Quantization noise with a  $\mu$ -law, PL law and without companding

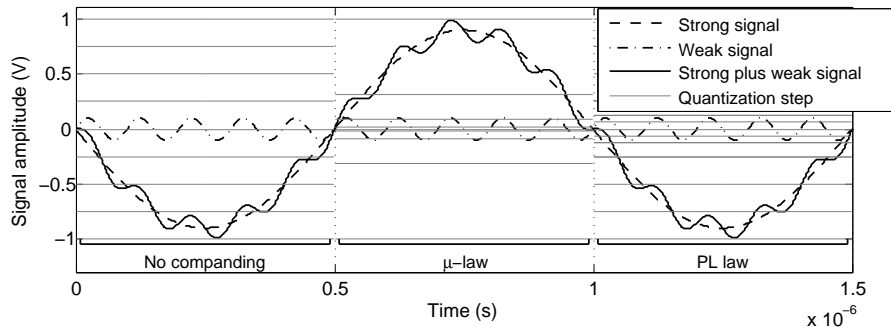


Figure 12: Equivalent quantization states when using a PL law compression

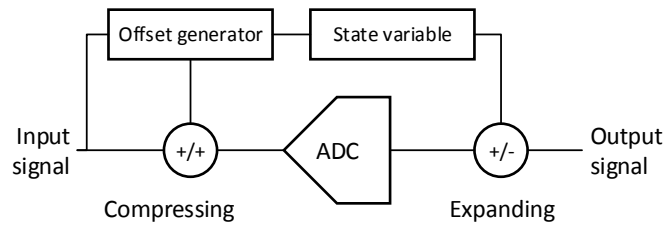


Figure 13: Block diagram of the PLCGO law

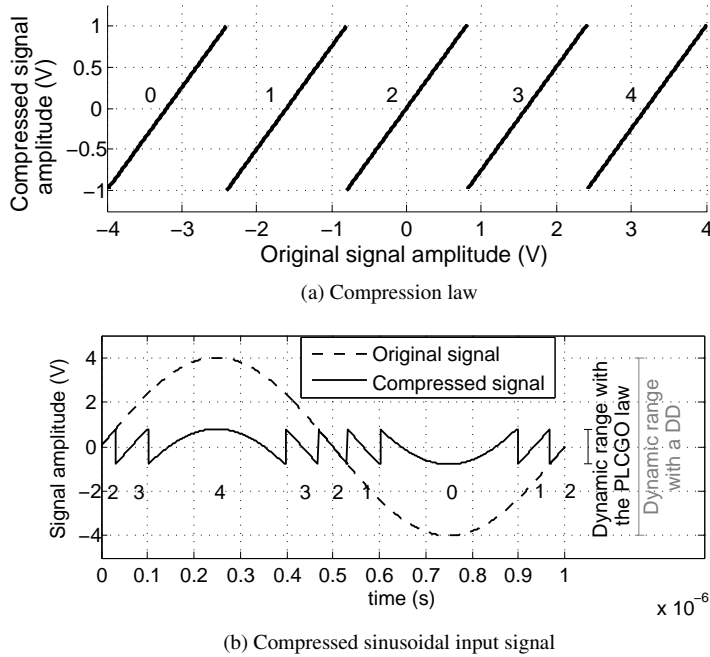


Figure 14: The PLCGO companding technique

defined on the input signal, delimiting functioning states. The offset value depends on the state variable. The principle of the law is shown on Figure 14, with a sinusoidal input signal for illustration, and the block diagram is shown on Figure 13. With  $x(t)$ ,  $w(t)$ ,  $off(t)$  and  $y(t)$  the input signal, the compressed signal, the compressing offset and the expanded signal, the compressing law can be expressed as:

$$\begin{aligned} w(t) &= x(t) + off(t) \quad (\text{Compressing}) \\ y(t) &= w(t) - off(t) \quad (\text{Expanding}) \end{aligned} \quad (4)$$

An example of the law is given in Figure 14. The gain of the law is shown in Figure 14a whereas the compression step is represented in Figure 14b, with a sinusoidal input signal for illustration. One can see that the dynamic range is reduced by the use of this compressing law.

The most similar approach is the PL law [6]. Indeed this law uses functioning states too, but the compressing gain is modified instead of adding an offset regarding the state value. The PL law is thus the most relevant law to compare the additive companding law's performance with. In this report the additive companding law is referred to as a PLCGO to keep the same notation convention as with the PL law.

The PLCGO law has already been implemented in [7] for a different application (*i.e.* cochlear implants). However, their implementation has the drawback of requiring to save the path that have led to the current state variable value (*i.e.* the list of the states the signal crossed before arriving in the current state), and not the state variable itself. This can lead to memory issues, especially if a lot of states are used with fast-varying signals. Another implementation is proposed latter in this report.

The Figure 15 outlines the benefit of the PLCGO law (Figure 15c) with regard to the PL law (Fig 15b) with a sinusoidal input (Figure 15a): the quantization noise was dependent on the input signal amplitude with the PL law, leading to a fall of performance because of the presence of the strong signal. This effect has disappeared with the PLCGO law. By comparison with the quantization noise with DD (Figure 15d), one can see that the quantization noise is lessened with the PLCGO law. Moreover, the quantization noise



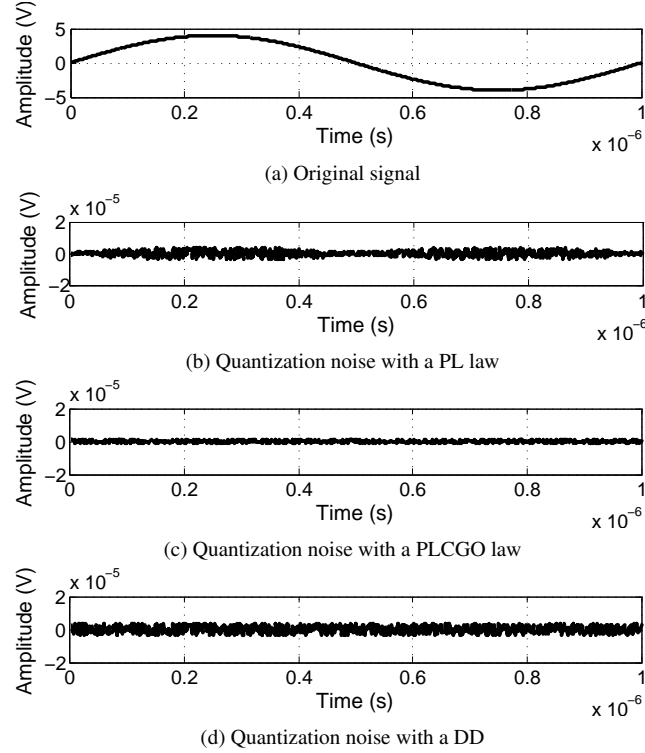


Figure 15: Quantization noise with a PL law, PLCGO law and a DD

level is independent of the  $S_s/S_w$  value, as with a DD.

On Figure 16, the equivalent quantization step of the PLCGO and PL companding laws have been plotted. With the PL law, the quantization step was lower when the input signal was high, but this effect has disappeared with the PLCGO law: the quantization step is uniform whatever the input signal level is. Then the weak signal is properly quantized regardless of the total signal amplitude.

## 5.2 Analytical impact of PLCGO companding

Assuming the input uncompressed signal varies between  $-V_M$  and  $V_M$ , the  $i^{th}$  threshold will take the value:

$$V_{th,i} = -V_M + \frac{2 \cdot V_M}{N_{th} + 1} \cdot i \quad (5)$$

with  $N_{th}$  being the total number of thresholds. In a given state, the gain  $G$  that should be applied to the signal depends on the ADC's dynamic range  $V_{ADC}$  and on the number of states  $N_{th} + 1$ . As the thresholds are equidistributed, the gain does not depend on  $i$ . It can be expressed as the ratio of the ADC full-scale range and the difference of two voltage thresholds:

$$G = \frac{V_{ADC}}{V_{th,i+1} - V_{th,i}} = (N_{th} + 1) \cdot \frac{V_{ADC}}{2V_M} \quad (6)$$

The impact of this law can be evaluated through the quantization noise. On Figure 14b, it can be seen that after compressing, the total signal's dynamic range is reduced by a factor equal to the number of states. The quantization step  $\Delta$  is reduced by the same factor (assuming the signal is scaled to the ADC

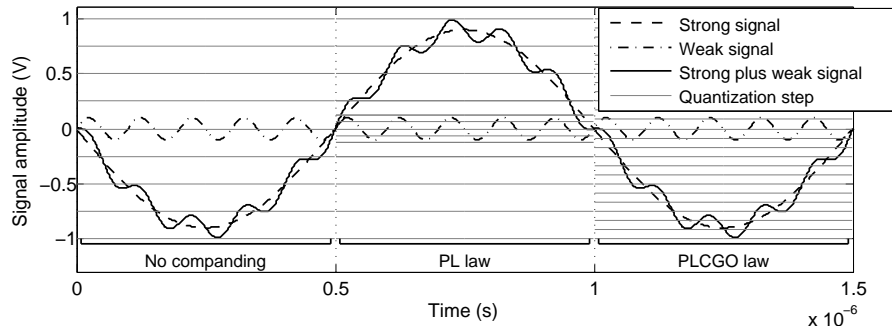


Figure 16: Equivalent quantization states when using a PLCGO law compression

input), since it is the ratio between the dynamic range and the number of quantization states. So,  $\Delta_{PLCGO}$  and  $\Delta_{DD}$  denoting the equivalent quantization step with the PLCGO compressing law and the quantization step with a DD:

$$\Delta_{PLCGO} = \frac{\Delta_{DD}}{N_{th} + 1} \quad (7)$$

$N_{Q,PLCGO}$ ,  $N_{Q,DD}$  are respectively the quantization noise with the PLCGO law and with a DD. The quantization noise being  $\Delta/\sqrt{12}$  [15], the benefit of the law in terms of quantization noise can be expressed from (7) as:

$$\frac{N_{Q,PLCGO}}{N_{Q,DD}} = \frac{\Delta_{PLCGO}}{\Delta_{DD}} = \frac{1}{N_{th} + 1} \quad (8)$$

The quantization step is the ADC full scale voltage divided by the number of quantization states  $2^n$  of the ADC ( $n$  being the ADC resolution).  $n_{PLCGO}$  and  $n_{DD}$  being respectively the equivalent resolution when using the PLCGO compressing law and the real resolution of the ADC, the gain in terms of number of bits of resolution can be expressed from (8) as:

$$\begin{aligned} \frac{\Delta_{PLCGO}}{\Delta_{DD}} &= \frac{2^{n_{DD}}}{2^{n_{PLCGO}}} = \frac{1}{N_{th} + 1} \\ \Rightarrow n_{PLCGO} &= n_{DD} + \log_2(N_{th} + 1) \end{aligned} \quad (9)$$

The great interest of this approach is highlighted in (9), since PLCGO companding allows to save a number of bits equal to the binary logarithm of the number of states. We remind here that 21 bits of resolution are required to digitize the set of received signals in the 8 MHz band. Implementing PLCGO companding with 32 states would require an ADC resolution of 16 bits. Such a resolution is still unaffordable today, but recent work performance [16] allows to expect such ADCs will be available in a foreseeable future.

### 5.3 Theoretical performance of the law

The PLCGO law has been first cosimulated to assess the relevance of the approach in a multiple signals case. As previously, a strong and a weak signals are simulated using the Agilent ADS software with a BPSK modulation, a bitrate of 200 kbps and a AWGN. The simulation schematic remains nearly the same as in sections 3 and 4: the only difference is the companding law. The compressing and expanding functions are cosimulated through the MatLab software. The BER on the weak signal is simulated with  $E_b/N_0 = 7$  dB while  $S_s/S_w$  and the ADC resolution are varying. The sampling frequency is still of

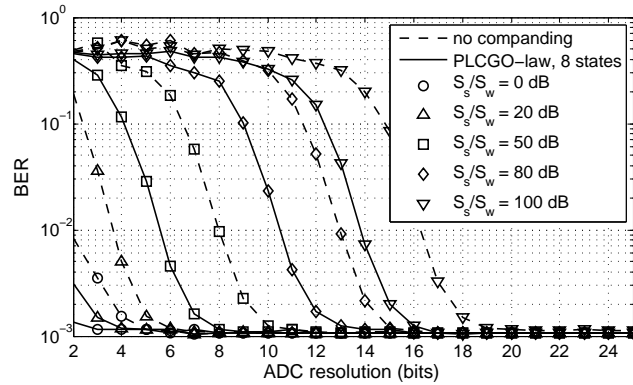


Figure 17: Simulated BER as a function of  $S_s/S_w$  and of the ADC resolution, with and without companding using a PLCGO law

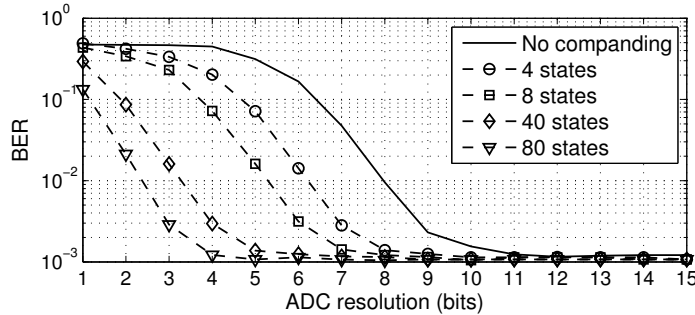


Figure 18: BER vs the ADC resolution for  $S_s/S_w = 50$  dB, depending on the number of thresholds used with the PLCGO law

16 MHz, and the PLCGO companding is performed using 8 states. During the simulations, The ADC resolution and  $S_s/S_w$  are varying from 1 to 25 bits and from 0 to 100 dB. The results are shown on Figure 17. When comparing with the  $\mu$ -law (Figure 5) and PL law (Figure 10) performance, one can see that the BER does not depend anymore on  $S_s/S_w$ . Moreover, the BER is significantly improved with the PLCGO law, whereas it was only lightly improved with the PL law and degraded with the  $\mu$ -law.

Other simulations has been done to evaluate the required resolution gain regarding the number of states used in the PLCGO law compression.  $E_b/N_0$  still being 7 dB,  $S_s/S_w$  is arbitrary fixed to 50 dB. This value is not of prime importance since the ADC resolution improvement only depends on the number of states. The simulations are done by varying the ADC resolution and the number of states. Assumption is made that there is no delay between the time the signal cross a threshold and the time the appropriate offset is added to the signal.

For a given number of states, the minimum resolution is obtained when the BER value is close to the expected value of  $10^{-3}$  (which is the theoretical BER when using a BPSK modulation with  $E_b/N_0 = 7$  dB). It can be seen on Figure 18 that rising the number of states allows to reach the expected BER floor with a smaller resolution. The use of 40 states reduces the required resolution by 5 bits, showing the interest of the method.

The relevance of the PLCGO compressing law to reduce the required ADC resolution to digitize multiple signals being proved, the next section's purpose is to propose an implementation of this law.

Indeed with the implementation proposed in [7], the complete list of states crossed by the signal are saved instead of only saving the state variable itself. In the case of urban sensor networks the signals are fast varying, and thus this approach is not suitable.

## 6 PLCGO law architecture proposals and implementation

### 6.1 Impact of the delay of updating of the state variable

With the PLCGO law as well as with the PL law, the state variable must be updated when the signal reaches a threshold level to apply respectively the proper offset and gain. The threshold crossing can be detected both in analog or digital domains.

It has been previously assumed that the state variable update was instantaneous, but this assumption does not stand if the threshold crossing detection is performed after digitization. In that case, the delay would be at least one sampling period  $T_s$ . The existence of this delay of updating is a problem, as during this delay a wrong offset is added to the signal. This can lead to an ADC clipping and thus to the loss of the signal's information during this time interval. This problem cannot be addressed by using a scaling factor between the compression and digitization steps, since the number of thresholds that the signal can cross during the delay of updating cannot be predicted on a real signal.

The impact on a single signal's BER of the state variable delay of updating has been evaluated by performing the compression with a delay set to  $T_s$ . If ideal companding (*i.e.* without delay) would allow to properly digitize the signal, the BER remains equal to 0.5 when a delay of  $T_s$  is added, regardless of the ADC resolution: the delay is so big that the expanding law is almost never applied using the proper state variable. Thus the law must be implemented analogically, to guarantee the delay of updating to be negligible compared to  $T_s$ .

### 6.2 Implementation proposal

A simple implementation would be to use a Sample and Hold (S&H) device to generate the offsets. But this solution would then rely on time-based discretization, although PLCGO is based on an amplitude-based discretization. The biggest issue of this option is the required digitization of the S&H output signal, which has the same dynamic range as the original signal. It turns out that the digitization of the S&H output implies the same resolution as without companding, meaning 21 bits. Another solution must be found to implement this law.

One can use comparators to generate the offsets. A parallelized and a cascaded implementations are proposed, whose diagrams are shown on Figure 19. In the parallelized implementation (Figure 19a), the comparators compare the signal with pre-defined thresholds that can be generated through bandgap voltage references. Their outputs are summed, giving a proportional value to the offset that must be added to the signal. This value is scaled to generate the real offset and is subtracted from the signal. The resulting signal is then scaled to the ADC input. On Figure 19a, a differential amplifier performs all these operations. Another digitization (not shown on Figure 19a) is still required to get the offset in a digital form. To do this, the comparators outputs are summed, giving the state variable and digitized with a resolution equal to the binary logarithm of the number of thresholds. As it was found through (9) that 32 thresholds should be enough to properly digitize the signal with a 16 bits resolution (with  $S_s/S_w = 100$  dB), the offsets can be digitized with a 5 bits ADC. So this solution could allow in theory (*i.e.* without considering the implementation imperfections) to replace one ADC with a 21 bits resolution with two ADCs with resolutions of 16 and 5 bits.

With the cascaded implementation (Figure 19b), the offset value is found iteratively. The first comparator gives the most significant bit of the offset value and this value ( $V_{thi}$ ) is scaled to the input signal and subtracted from it through a differential amplifier. The resulting signal is the input signal compressed

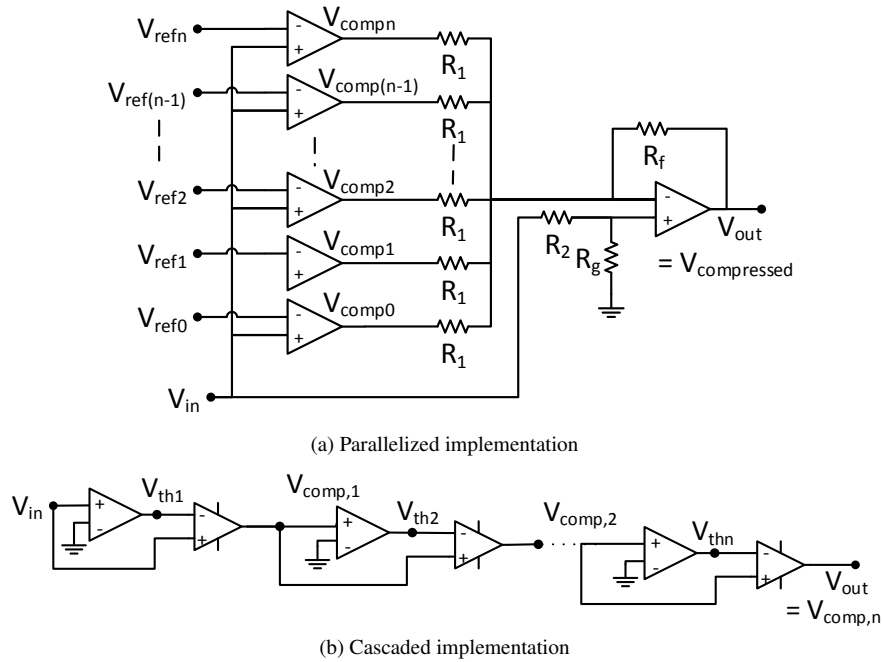


Figure 19: Proposed implementations of the PLCGO law

with one threshold. Then the second comparator gives the next bit of the offset value, and the corresponding value is subtracted from the signal. This operation is repeated until the total offset has been subtracted from the signal.

Given that the application of this digitization method targets sensors network gateways the main constraint to design a suitable implementation of the companding technique is the demodulation performance. The power consumption is not a key point, as it is assumed that the gateway is powered by the mains supply.

These architectures are compared through simulations with a  $S_s/S_w$  ratio of 50 dB.  $E_b/N_0$  is still 7 dB in all the simulations, and only the BER considering the weak signal is simulated. With both architectures, 8 states are used in order to compress the signal, allowing in theory (from (9)) to save 3 bits. Simulations have been made using ADC resolutions set to 25, 8 and 5 bits. It should be noted that a minimum resolution of 11 bits is required to digitize the signal without companding (see Figure 18), meaning that any resolution higher than 11 bits will provide the same BER performance. A 25-bit resolution therefore represents the expected satisfying level of performance that solutions implementing companding should be compared with.

Results reported in Table 1 are given with a relative variance of 0.01. They show that both architectures provide good results: the expected gain of 3 bits provided by the PLCGO law is confirmed. Indeed both architectures with 8-bit resolution provide the same BER value as the one provided by the minimum required 11-bit resolution (represented by the 25-bit resolution row). Along the same line, the BER levels obtained with companding with a 5-bit resolution are comparable (even if slightly above) to the BER value of a 8-bit resolution without companding.

ADC resolution	Simulated BER		
	parallelized architecture	cascaded architecture	without companding
5	$1.7 \cdot 10^{-2}$	$1.4 \cdot 10^{-2}$	$3.7 \cdot 10^{-1}$
8	$1.2 \cdot 10^{-3}$	$1.3 \cdot 10^{-3}$	$8.6 \cdot 10^{-3}$
25 ( $11 \rightarrow \infty$ )	$1.3 \cdot 10^{-3}$	$1.1 \cdot 10^{-3}$	$1.2 \cdot 10^{-3}$

Table 1: Simulated BER depending on the PLCGO compressing architecture

### 6.3 Discussion on the parallelized and cascaded implementations

From the BER performance perspective, one cannot prefer the cascaded or the parallel implementation. But both of them has some pros and cons: on one hand the parallelized architecture ensures that an error on one offset does not propagate to the other offsets, contrary to the cascaded architecture. On the other hand, a problem exists on the state transition: when the signal crosses a threshold, the comparators output does not switch from the low to the high state immediately, as the comparator amplifier's gain is not high enough. During this transition time, the offset takes a voltage value ranging between two predefined values. If this happened at an ADC sampling time (assuming the clocks of the offset's ADC and of the compressed signal's ADC are aligned), the signal would be compressed and expanded with a different offset value. The probability of this to happen is smaller with the cascaded architecture, as the amplifiers' gains cumulate from one stage to the other.

The selection of a given architecture can also be led from an implementation complexity perspective. The parallelized architecture uses more amplifier stages: a  $k$  states compression requires  $k$  amplifier stages if they are in parallel, whereas  $2 \cdot \log_2(k)$  amplifier stages are enough if cascaded. To digitize the state variable, the parallelized implementation is more advantageous as it sums the comparators outputs. So it can be digitized using a summing amplifier. In the cascaded implementation case, each comparator output is one bit of the final offset value. The all comparators outputs should be digitized separately, with a one-bit ADC.

## 7 Conclusion

In the special case of urban sensor networks, the high dynamic range of received signals makes the digitization of the full band impossible with today's ADCs. A way to tackle this problem is to use the companding technique. Three existing companding laws have been studied: the  $\mu$ -law, the PL law and the PLCGO law. Only two signals have been considered, one with a low power level (weak signal) and the other one with a high power level (strong signal). The  $\mu$ -law is considered to be representative of other similarly-shaped companding functions, such as the  $\sinh$ ,  $\tanh$  or  $\sqrt{\cdot}$  functions. The  $\mu$ -law represents bijective companding functions, whereas the PL law represents the non-bijective functions, which require the use of a state variable. The PLCGO law (which is also non-bijective) is an additive companding technique, and therefore behaves differently than the  $\mu$ -law and the PL law, which are multiplicative companding techniques.

For both the PL and the  $\mu$ -laws, the performance in terms of BER when demodulating the weak signal and of quantization noise has been studied. The  $\mu$ -law presents a disappointing performance: the BER is degraded in most of cases, and the quantization noise is globally amplified. The PL law performance is better since it lightly decrease the quantization noise: the BER is then improved. The BER performance and the quantization noise brought by the use of the PLCGO law have also been studied, highlighting the great interest of this approach with multiple signals. The PLCGO law performance is far better than the PL law's one: the quantization noise is drastically reduced by the use of this law. It has been shown that

the required resolution to digitize multiple signals can be reduced in theory by five bits by the use of 32 states with this law.

The implementation of the PLCGO law proposed in [7] may not be relevant with the signals considered in urban sensor networks, which are fast-varying signals. For that two implementations of the PLCGO law are proposed, that facilitate the digitization of the state variable. These implementations are based on cascaded or on parallelized amplifiers stages. Both architectures have been simulated using the Agilent ADS software and confirm their expected theoretical performances. The main advantage of the cascaded architecture is its low complexity, whereas the parallelized architecture allows more precision when offsets are generated since an error on one comparator does not propagate to the other ones. However, the cascaded architecture cumulative gain is an advantage, as the comparators saturation is faster, leading to a smaller probability of error when digitizing the state variable.

Both architectures have limits, since the offset must be generated with the same precision than the target equivalent resolution, and its AC component must be less than one LSB. Future work will focus on a physical implementation of these architectures and tests on real systems in order to evaluate this method's feasibility in practice.

## References

- [1] ETSI, "Electromagnetic compatibility and Radio spectrum Matters (ERM); System Reference document (SRdoc): Spectrum Requirements for Short Range Device, Metropolitan Mesh Machine Networks (M3N) and Smart Metering (SM) applications," Sept. 2011, TR 103 055.
- [2] M. Dardaillon, K. Marquet, J. Martin, T. Risset, and H.-P. Charles, "Cognitive radio programming: Existing solutions and open issues," Inria, Tech. Rep. 8358, Sep. 2013. [Online]. Available: <http://hal.inria.fr/hal-00859467/>
- [3] M. Vallerian, G. Villemaud, B. Miscopein, T. Risset, and F. Hutu, "SDR for SRD: ADC specifications for reconfigurable gateways in urban sensor networks," in *Radio and Wireless Symposium (RWS), 2014 IEEE*, Jan 2014, pp. 178–180.
- [4] B. Jonsson, "A survey of A/D-Converter performance evolution," in *Electronics, Circuits, and Systems (ICECS), 2010 17th IEEE International Conference on*, 2010, pp. 766–769.
- [5] D. Van Ess, *Algorithm - Logarithmic Signal Companding - Not just a good idea - it is  $\mu$ -Law*, Cypress, december 2013, AN2095.
- [6] V. Maheshwari, W. Serdijn, and J. Long, "Companding Baseband Switched Capacitor Filters and ADCs for WLAN Applications," in *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, 2007, pp. 749–752.
- [7] C. Bes, C. Sawigun, and W. Serdijn, "An additive instantaneously companding readout system for cochlear implants," in *Biomedical Circuits and Systems Conference (BioCAS), 2010 IEEE*, Nov 2010, pp. 126–129.
- [8] A. B. Clark, H. Nyquist, and D. K. Gannett, "Electrical Picture-Transmitting System," US Patent 1,691,147, 1928.
- [9] Y. Tsividis, "Externally linear, time-invariant systems and their application to companding signal processors," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 44, no. 2, pp. 65–85, 1997.

- 
- [10] C. O. Mallinckrodt, "Instantaneous Companders," *Bell System Technical Journal*, vol. 30, no. 3, pp. 706–720, 1951.
- [11] A. Katsiamis, H. Ip, and E. Drakakis, "A Practical CMOS Companding Sinh Lossy Integrator," in *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, May 2007, pp. 3303–3306.
- [12] P. Chatchawaree and A. Thanachayanont, "A 3rd-order elliptic syllabic-companding tanh filter," in *TENCON 2004. 2004 IEEE Region 10 Conference*, vol. D, Nov 2004, pp. 376–379 Vol. 4.
- [13] C. de La Cruz-Blas and O. Feely, "Limit cycle behavior in a class-AB second-order square root domain filter," in *Electronics, Circuits and Systems, 2008. ICECS 2008. 15th IEEE International Conference on*, Aug 2008, pp. 117–120.
- [14] J. Mulder, W. Serdijn, A. van Der Woerd, and A. H. M. Van Roermund, "An instantaneous and syllabic companding translinear filter," *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, vol. 45, no. 2, pp. 150–154, Feb 1998.
- [15] S. Gupta and A. Phatak, *ADC Guide, Part 1 – The Ideal ADC*, Cypress Semiconductor, Janvier 2012.
- [16] C. Hurrell, C. Lyden, D. Laing, D. Hummerston, and M. Vickery, "An 18 b 12.5 MS/s ADC With 93 dB SNR," *Solid-State Circuits, IEEE Journal of*, vol. 45, no. 12, pp. 2647–2654, 2010.





**RESEARCH CENTRE  
GRENOBLE – RHÔNE-ALPES**

Inovallée  
655 avenue de l'Europe Montbonnot  
38334 Saint Ismier Cedex

Publisher  
Inria  
Domaine de Voluceau - Rocquencourt  
BP 105 - 78153 Le Chesnay Cedex  
[inria.fr](http://inria.fr)

ISSN 0249-6399