A passive and low-complexity Compressed Sensing architecture based on a charge-redistribution SAR ADC

A Passive and Low-complexity Compressed Sensing Architecture Based on a Charge-redistribution SAR ADC

Carmine Paolino[a], Luciano Prono[a], Fabio Pareschi[a,c], Mauro Mangia[b,c], Riccardo Rovatti[b,c], Gianluca Setti[a,c]

[a]*Department of Electronics and Telecommunications - Politecnico di Torino, 10129 Torino, Italy*
[b]*Electric, Electronic and Information Engineering Department - University of Bologna, 40136 Bologna, Italy*
[c]*ARCES Research Center - University of Bologna, 40125 Bologna, Italy*

Abstract

An innovative analog-to-digital converter (ADC) architecture is proposed, with the aim of acquiring an input signal according to the Compressed Sensing (CS) paradigm and without the need for dedicated active analog blocks. Its core is the capacitive array employed in traditional successive-approximation-register (SAR) ADCs. Introducing only a few additional switches, the array can compute the linear combination of consecutive signal samples, as required by the CS encoding.

To manage the presence of leakage currents, which may impair signal reconstruction, a compensation circuit is considered, allowing close-to-ideal performance of the system when properly designed. A neural network-based decoding strategy is also analyzed, with up to 20 dB of additional reconstruction quality with respect to standard algorithms. Synthetic electrocardiogram signals are used to validate optimizations both at the hardware level in the encoding block and at the software level in the decoder.

Keywords: Compressed Sensing, sub-Nyquist sampling, Successive Approximation Register, Analog-to-Digital conversion, Leakage Compensation, Deep Neural Networks

1. Introduction

Compressed Sensing (CS) is a signal processing technique [1, 2] aimed at representing a broad family of signals with fewer scalar quantities than what the Nyquist-Shannon theorem suggests. It effectively allows sub-Nyquist sampling in scenarios where energy and bandwidth are heavily constrained.

Practical applications of the CS theory have emerged in particular in the biomedical field [3, 4]. More recently, circuital implementations of ultra-low power biosensor nodes have been proposed [5–9]. Specifically, in [6] the authors propose an integrated circuit based on CS for multi-lead implantable neural recording, while in [8] CS is exploited for the efficient wireless transmission of neural signals. The circuit in [7] is a low-power CS-based acquisition system for electrocardiograph (ECG) signals and in [9] a CS-based architecture with effective hardware-software optimization is presented. Other applications can be found among radio-frequency (RF) signal receivers [10, 11] and current sensors [12].

The major issue of all the aforementioned solutions is the need for dedicated active circuital elements in the signal processing chain, such as continuous-time [10, 11] or switched-capacitors integrators [6, 7, 9]. As a consequence, the potential energy saving enabled by a sub-Nyquist-rate conversion is hindered by the energy requirements of the additional active elements, which are generally dominant with respect to one of the analog-to-digital converter (ADC).

The main innovative feature of the solution proposed here is based on the observation that the speed and accuracy requirements for the conversion of the measurements generated by a CS system are generally satisfied by a successive-approximation-register (SAR) ADC. In fact, the sub-Nyquist operation typical of CS relaxes the speed constraint, whereas a low resolution is in general sufficient because of the lossy compression mechanism, achieved at the cost of reducing the signal reconstruction quality. Therefore we propose to use the capacitive array already found in traditional charge-redistribution SAR converters (with suitable modifications involving a few additional switches) to enable the CS-based processing. The advantage is to have additional hardware that is *passive*, i.e., it does not require any power-hungry circuit like an active integrator or an operational amplifier. In this way, with the same structure it is possible to: *i*) sample the modulated input signal at different time steps; *ii*) hold several sampled values simultaneously until the end of the acquisition window; *iii*) evaluate the linear combination of the values being held; and *iv*) convert the result into a digital word.

In this paper we describe the working principle of the proposed architecture, we analyze its drawbacks (both at the hardware and system level) and we discuss effective techniques to cope with them. As an example, the limited hold-time allowed by the (small) capacitive cells of the array results in a degradation of the performance for long acquisition windows, poten-

Email addresses: carmine.paolino@polito.it (Carmine Paolino), luciano.prono@polito.it (Luciano Prono), fabio.pareschi@polito.it (Fabio Pareschi), mauro.mangia2@unibo.it (Mauro Mangia), riccardo.rovatti@unibo.it (Riccardo Rovatti), gianluca.setti@polito.it (Gianluca Setti)

tially requiring compensation of the leakage currents.

Some preliminary results on the proposed architecture have already appeared in [13]. With respect to that work, we propose here a better description of the hardware architecture, with particular emphasis on the circuit used to compensate the leakage-induced discharge of the hold capacitances. Furthermore, thanks to a new decoding strategy we are able to introduce an additional level of optimization. The proposed decoder relies on a deep neural network (DNN) and is based on the Trained CS with Support Oracle (TCSSO) technique that has recently appeared in the literature [14]. It has been adapted to the specific features of the hardware architecture discussed here, and shows extremely good results.

The paper is organized as follows. Section 2 introduces the basic theoretical framework of CS. In Section 3 we discuss the main issues of an analog CS implementation and, among the known solutions to mitigate them, we identify the most suitable ones for the proposed architecture. Section 4 describes the proposed circuit, while in Section 5 we focus on the compensation of of leakage currents. In Section 6 we introduce the innovative approach for signal decoding, that reveals to be particularly well suited to work with the proposed circuit. Finally, we draw the conclusion.

2. Compressed Sensing Fundamentals

Compressed Sensing is an encoding/decoding strategy applicable to signals that are compressible, i.e., with an informative content much lower than what their bandwidth would suggest. Specifically, the number of scalars (*measurements*) needed to correctly represent a compressible signal is much smaller than the number of samples required by a straightforward application of the Nyquist-Shannon theorem.

The typical application of a CS based acquisition system is schematized in Figure 1. The input signal x (exemplified by an ECG signal) is encoded and compressed in a set of measurements y, that can be stored in a local memory, transmitted to a receiver, or sent to the cloud for further processing. The main difference with respect to a traditional compression scheme [15] is that, when using CS, encoding and compression are obtained as a single and very simple operation, transferring complexity to the decoding stage. The approach fits perfectly the scenario where signal acquisition has to be performed by a small, battery-powered sensor node, while reconstruction is completed in the cloud.

From a more formal point of view, let us define $x \in \mathbb{R}^n$ as the discrete-time representation of the input signal over a time window made of n consecutive samples. Each signal sample is identified as x_k, with $k = 1, 2, \ldots, n$. Let also $D \in \mathbb{R}^{n \times n}$ be an arbitrary basis, and $\xi \in \mathbb{R}^n$ the representation of x in terms of D, i.e. a vector such that $x = D\xi$. Finally, let the *support* of x be the vector $s \in \{0, 1\}^n$ such that its generic element $s_j = 0$ if $\xi_j = 0$, and $s_j = 1$ otherwise. The assumption on which CS relies is that the input signal is *sparse* in D. A signal is κ-*sparse* if, for any possible x, its representation ξ in terms of D has at most $\kappa \ll n$ non-null elements, i.e., $\|\xi\|_0 \leq \kappa$, or equivalently, the support of x is such that $\|s\|_1 \leq \kappa$, where the notation $\|\cdot\|_p$ refers to the standard ℓ_p norm. The matrix D is also known as *sparsity basis*.

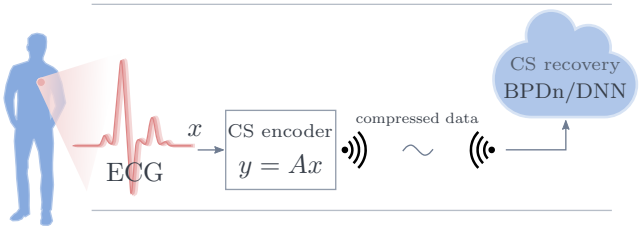


Figure 1: Typical application of a CS based acquisition system.

In the CS framework, a signal is encoded into a set of m measurements, arranged in the vector $y \in \mathbb{R}^m$. Such a vector is obtained by a linear projection of x on the m rows of a sensing matrix $A \in \mathbb{R}^{m \times n}$

$$y = Ax + \nu = AD\xi + \nu, \tag{1}$$

where $\nu \in \mathbb{R}^m$ accounts for noise and non-idealities in the acquisition process. Since $m < n$, the acquisition is *compressive* as expressed quantitatively by the Compression Ratio $\mathrm{CR} = m/n$.

Recovering x from y (i.e., signal reconstruction) is an ill-posed problem, since many vectors ξ exist generating the same measurement vector. According to the standard CS theory [1, 16], correct reconstruction is guaranteed if: *i*) the number of measurements is sufficient, with $m = O(\kappa \log(n/\kappa))$, hence depending both on κ and on n; *ii*) the elements of A are instances of independent and identically distributed (i.i.d.) Gaussian random variables. Given these assumptions, we have theoretical guarantees that the correct solution $\hat{\xi}$ is the sparsest among all possible ξ satisfying equation (1).

Looking for the sparsest solution is indeed a computationally intractable problem, involving the counting of the non-zero components of ξ, as given either by $\|\xi\|_0$ or $\|s\|_1$. However, a nontrivial consequence of the previous assumptions is that sparsity can be promoted by looking at $\|\xi\|_1$ instead, favoring the convergence of the recovery algorithm. Such an approach is called basis pursuit with denoising (BPDn), and is equivalent to solving the optimization problem

$$\hat{\xi} = \underset{\xi \in \mathbb{R}^n}{\operatorname{argmin}} \|\xi\|_1 \qquad \text{s.t. } \|AD\xi - y\|_2 < \varepsilon, \tag{2}$$

where ε accounts for the effects of ν. The input signal is then reconstructed as $\hat{x} = D\hat{\xi}$.

Many algorithms have been proposed to solve problem (2) either in an exact or an approximated way [17–21]. In this paper we refer to the SPGL1 procedure [21], freely available at [22].

3. Reducing Complexity of the CS Acquisition Stage

The CS paradigm was originally introduced to reduce the energetic requirements of the encoder stage in a signal processing chain. Hence, any possible additional reductions in terms of circuital complexity is fundamental.

One of them is to constrain the admissible values of the entries in the sensing matrix A. This method requires less resources to compute (1), and is adopted by almost all hardware implementations of CS-based acquisition systems proposed in the literature [6, 8–11].

As an example, antipodal values, i.e. $A \in \{-1, +1\}^{m \times n}$, reduce multiplications to simple sign inversions, that come almost for free in a differential implementation and shows no performance loss [23].

Binary matrices, i.e. $A \in \{0, 1\}^{m \times n}$, or ternary, i.e., $A \in \{-1, 0, +1\}^{m \times n}$, allow further energy saving since the zero entries of A do not contribute to the whole energy cost. Several classes of matrices A belong to the latter class have been investigated in the literature [24–27]. Authors of [27] explore the trade-off between zeroing at random the entries of A and the CS performance in terms of signal recovery. The same topic is also discussed in [28], where the position of the non zero entries is not anymore randomly drawn, but tuned according to the statistical characterization of the class of acquired signals. In that case, authors did not observe any performance degradation even when 87.5% of the entries in the sensing matrix A are zeroed.

In this section we will go through a few other techniques developed in the literature to ease the hardware implementation of (1) and that are of relevance in the proposed architecture. In particular, we focus on two complementary approaches. The first one is focused on effectively reducing the complexity of the circuitry required to implement the linear projection. In the second one we instead aim at improving the performance of the CS system. This may be interpreted either as increasing reconstruction quality at a given CR, or as increasing CR at a given reconstruction quality. The latter corresponding to a reduction the number of measurements m and therefore of the complexity of the encoder generating y.

3.1. Reducing encoding complexity: short windowing

In a CS-based acquisition, the signal is processed in batches of n consecutive samples. If the total length of the signal is $N > n$, it has to be partitioned into N/n contiguous and non-overlapping time windows, each containing n samples[1]. CS is then applied separately to each window.

To reduce the computational complexity of the encoder, the system should be designed to operate with the smallest possible n. To understand why, we have to consider that the application of (1) to a single time window requires $O(n \cdot m)$ multiply-and-accumulate operations. Extending the computation over all N/n time windows, the total number of operations increases to $O(n \cdot m \cdot N/n) = O(n \cdot N/\mathrm{CR})$.

If CR is chosen to guarantee a target reconstruction quality and assuming, reasonably, that the input signal length N is a constraint, the computational complexity increases linearly with n.

[1]We are implicitly assuming that N is an integer multiple of n in order to keep the description simple. This assumption holds for the rest of the paper.

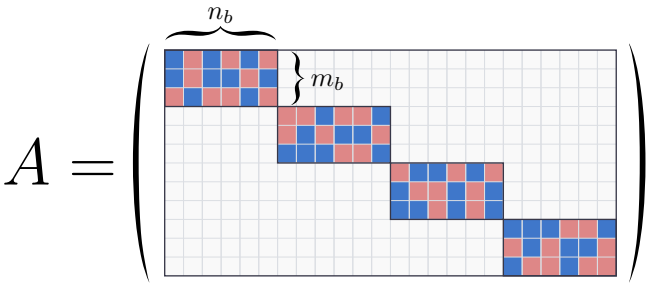


Figure 2: Example of a block-diagonal sensing matrix A, with $n = 24$, $n_b = 6$, $m = 12$ and $m_b = 3$. White blocks correspond to zeroes.

However, some other effects have to be considered as well. Let us reformulate (1) component-wise as

$$y_j = \sum_{k=1}^{n} A_{j,k} x_k + \nu_j, \quad j = 1, \ldots, m \tag{3}$$

where $A_{j,k}$ is the element of A at the intersection of the j-th row and k-th column, and ν_j the j-th component of ν.

First, the hardware resources needed to compute (3) are increasing[2] with m, and a large n implies a large $m = n \cdot \mathrm{CR}$.

At the same time, the noise on y_j increases, since detrimental effects such as clock feedthrough or charge injection cause a degradation that depends linearly on n.

Finally, as the individual x_k are available at different times, they must be sampled and held by the circuit to allow the computation of y_j. For slowly varying signals, leakage currents become a concern, as their effect increases with the hold time, which is proportional to n.

Regrettably, in many CS applications a straightforward reduction of n is not a valid option, since the sparsity properties (hence the ability of correctly retrieving the original signal) are only observed for values of n sufficiently large [14].

A workaround is the design of A as an *antipodal block-diagonal matrix* as in Figure 2, i.e. where the $m_b \times n_b$ blocks lying on the main diagonal of A have antipodal-valued entries, while the rest are set to zero. The aspect ratio of the blocks is the same of the original matrix, with $m_b/m = n_b/n$ or, in terms of compression ratio, $m_b = \mathrm{CR} \cdot n_b$.

With this, there are only n_b non-null elements in every row of A, and they are consecutive. This reduces to n_b the effective number of terms in (3). Furthermore, non-null elements in different blocks do not overlap, so that the corresponding measurements can actually be computed reusing the same hardware. As a consequence, the number of physical channels required to compute all measurements is reduced to m_b.

The effectiveness of this matrix structure is confirmed by the empirical results published in the literature [5, 12]. For an exhaustive discussion on the consequences of employing a block-diagonal sensing matrix, we refer to [29].

[2]This is certainly true when the circuit to compute y_j is replicated m times. A possible alternative to the simple hardware replication is to use the same circuit in an interleaved way to compute all m measurements [6]. In this case, the speed (hence the power consumption) has to be increased by a factor m, leading again to an increase in the required resources.

Table 1: Summary of the tradeoffs involving n or, equivalently, n_b, indicating the section where the topic is discussed. Whenever possible, an estimate of the dependence is given.

Tradeoff	Dependency	Section
Encoder computational complexity	$O(n \cdot m)$	Sec. 3.1
No. of encoder hardware channels	$O(n \cdot N/\mathrm{CR})$	
Noise injected in measurements	$O(n)$	
Extension of the acquisition window	$O(n)$	
Observed signal sparsity	$\uparrow$ with n	
Achievable reconstruction quality	$\uparrow$ with n_b	Sec. 3.4
Leakage-induced degradation	$\uparrow$ with n_b	
Compensator dissipated power	$\uparrow$ with n_b	Sec. 5
Number of leakage compensators	$O(n_b)$	

Yet, this approach actually introduces a tradeoff. Values of n_b too low result in structured sensing matrices having a large number of structured zeroes, violating the requirements of the standard CS theory for some classes of signals [23]. Therefore, even if noise and degradation of the measurements are reduced, also the theoretical performance achievable by the CS reconstruction decreases. An example of this trade-off will be provided in Section 3.4.

As a convenient aid to the reader, Table 1 collects all the parameters and properties having a dependence on n and n_b described in the text.

3.2. Improving performance by adaptation: Rakeness CS

Many optimization techniques have been proposed with the aim of improving CS performance by adapting the statistical properties of A to those of the signals under consideration [30–32]. For an overview of these techniques, we refer the reader to [33] and we focus here on the approach known as *rakeness* [28, 32].

To the best of the authors' knowledge, this technique ensures a large performance improvement in comparison with other optimization approaches when applied in a realistic setting [33]. Due to this, we will consider a rakeness-based CS system as a reference case for all the analyses conducted in this paper.

The rakeness approach exploits an additional prior on the input signal named *localization*. Intuitively speaking, the higher the localization of a signal, the farther is the generating process from being white. Formally, x is localized if the $n \times n$ matrix C_x describing the correlation profile of the instances of x, computed as $\mathbf{E}\left[xx^{\mathrm{T}}\right]$, is not the $n \times n$ identity matrix I_n.

Let us indicate with $a \in \mathbb{R}^n$ the generic row of A. According to the rakeness approach, the elements of a are not drawn as instances of i.i.d. Gaussian or sub-Gaussian (e.g., Bernoulli distributed, as in the antipodal case) random variables, but using instead a multivariate random process defined by a $n \times n$ correlation matrix C_a, computed as follows:

$$C_a = \frac{1}{2}\left(\frac{C_x}{\mathrm{tr}(C_x)} - \frac{I_n}{n}\right),$$

where $tr(\cdot)$ is the trace operator.

The generation of the rows of A according to such a correlation profile maximizes the expected value of the energy collected by each measurement, $\rho = \mathbf{E}[ax]$, as well as the performance of the CS system.

A method to generate rows of A according to the rakeness approach under the constraint of random ternary values can be found in [28]. Both the case of non-structured zeroing (i.e., when the position of the zero elements is a degree of freedom) and of structured zeroing (when the position of the zero elements is constrained a priori) are considered. The latter approach fits perfectly the use of the antipodal block-diagonal matrices depicted in Figure 2.

3.3. Improving performance through DNNs: TCSSO

Joint use of DNNs and CS can be found in many recent works [34–41]. Typically, DNNs replace the classic BPDn approach of reconstructing the input signal either as $\hat{x}$ or as $\hat{\xi}$ directly from the measurement vector y.

Several examples are focused on compressed images, such as the 3-layer neural network proposed in [38], or the DNN described in [42]. As for other application fields, fast recovery and improved reconstruction quality of videos is obtained in [39], while a joint optimization of the encoding and decoding stages in the CS-based acquisition of EEG signals is presented in [41].

Interestingly, in some works (e.g. [41]) the training process of the DNN also generates the sensing matrix A, ensuring optimal results when applied to the signals contained in the training set, and also when using the DNN for signal reconstruction.

Here, we focus on the innovative approach introduced in [14], where the reconstruction process is split in two consecutive steps. First, a DNN is used to divine the signal support $\hat{s}$ of the reconstructed signal. After that, standard linear algebra is employed to get $\hat{x}$ and $\hat{\xi}$ from $\hat{s}$ and y. The training process of the support oracle network also determines the optimal sensing matrices A to be used for signal acquisition. According to the results presented in [14], the approach ensures a largely improved reconstruction quality. However, the application to antipodal block-diagonal matrix as that in Figure 2 is not straightforward, and requires a major modification of the DNN training algorithm. Details on how to make TCSSO work with antipodal block-diagonal matrices, as well as numerical results on its performance, will be provided in Section 6.

3.4. Case study: synthetic ECG signals

The numerical evaluation of the solutions previously discussed has been carried out on synthetic ECG signals. The generator employed is thoroughly described in [43][3] and the experimental setup is the following.

The heart-beat rate is randomly selected from a uniform distribution between 60 beats/min and 100 beats/min. Signals are generated as noiseless waveforms, sampled at 256 samples/s and split into chunks of length n.

[3]The MATLAB code is freely available for downloaded from the Physionet website at `http://physionet.org/content/ecgsyn/`

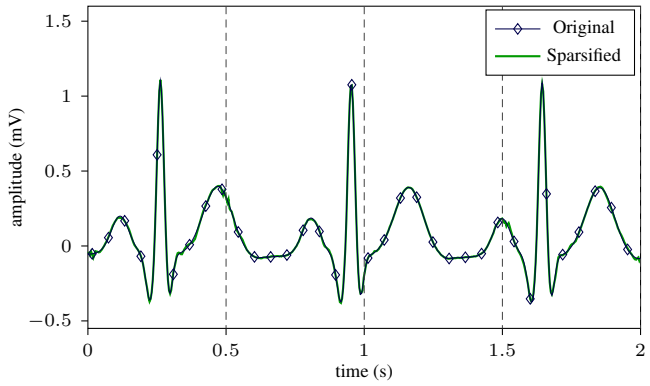


Figure 3: Example of a synthetic ECG with its sparsified version ($\kappa = 24$). The separation in consecutive time windows of length 0.5 s (i.e., $n = 128$ at 256 samples/s) is also highlighted.

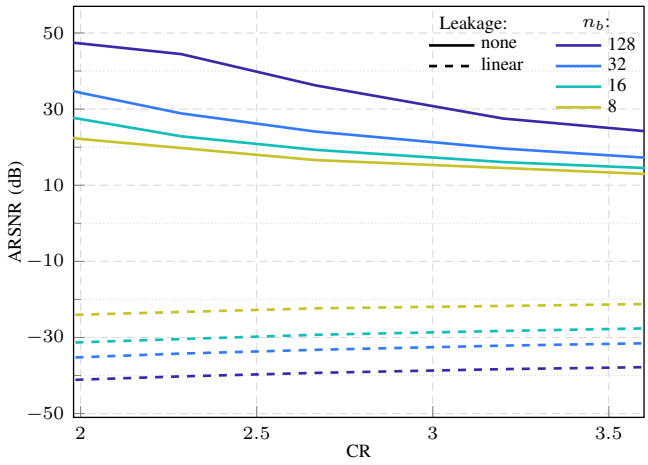


Figure 4: Performance of a CS-based signal processing chain for synthetic ECG signals at approx 60 beats/s, with $f_s = 256$ Hz and $n = 128$. Solid lines refer to an ideal system; dashed lines to measurements corrupted by a constant-current leakage discharge.

Moreover, in order to obtain a better fit of the signal properties with respect to the requirements imposed by CS, each signal window is modified as follows. *i*) The sparsity level is set to $\kappa = 24$ by zeroing the less energetic components of every signal instance, under the assumption of an orthonormal Symlet-6 wavelet sparsity basis D [44]. This operation introduces a sparsification-induced noise between 30 dB and 55 dB. An example is given in Figure 3, where a 32.4 dB signal-to-noise ratio (SNR) is observed. *ii*) In order to define a target for an ideal reconstruction, we superimpose white Gaussian noise to the sparsified signal so that its SNR is 50 dB.

Our benchmark is a CS system with $n_b = n = 128$, i.e. full antipodal sensing matrix, in which A is drawn according to the rakeness approach and where the signal is reconstructed by a standard BPDn technique through the SPGL1 algorithm as in [22]. In Figure 4 we compare the benchmark results with those obtained for block-diagonal matrices A with different values of n_b, as a function of the desired CR. The figure of merit under consideration is the reconstruction SNR (RSNR), defined as

$$\mathrm{RSNR[dB]} = 20\log_{10}\left(\frac{\|x\|_2}{\|\hat{x} - x\|_2}\right)$$

The plot shows the average value of the RSNR (ARSNR) observed over 1000 Monte-Carlo trials.

According to the data depicted in the figure, a full sensing matrix (i.e., $n_b = 128$), for values of CR ≈ 2 achieves performance close to the theoretical limit of 50 dB. As n_b is decreased, the number of zeroes in the matrix grows and less energy is collected from the signal. As a consequence, performance noticeably drops.

A significant decrease is also observed when the degradation problems addressed in Section 3.1 concerning the analog implementation of the CS encoder are considered. Targeting the sensing of low-frequency biomedical signals, leakage currents are indeed the predominant source of noise. In the example, using $n_b = 128$ with a sampling frequency $f_s = 256$ Hz implies that the value of $A_{j,1}x_1$ in (3) has to be sampled and preserved for a time period of almost 0.5 s, a hold time typically unaffordable even for pF-range hold capacitances.

In Figure 4, the solid lines obtained by an ideal system where no measurement degradation is considered are compared with the dashed curves, representing an acquisition process affected by a constant-current discharge of the hold capacitors due to leakage currents. The discharge model is based on actual data from a 180 nm CMOS technology. It considers a realistic configuration of four minimum size switches for each hold capacitor, leading to a 300 pA discharge current in the unfavorable condition of 85 °C. Moreover, the total sampling capacitance C_{tot} of the SAR array is kept constant to emulate equal area occupation and conversion power. Each hold capacitance is therefore equal to $C_h = C_{tot}/n_b$. For a fair comparison with respect to the solution proposed in [9], the value of C_{tot} has been set to 16 pF.

It is clear from the observation of the dashed curves in Figure 4 that block size has an opposite effect on reconstruction quality as compared to the ideal setup: a lower n_b shortens the acquisition window, reducing the degradation of measurements, with a positive effect on the reconstruction performance. However, it is also evident that performance is not sufficient for a decent reconstruction, requiring ad-hoc measures to counter the leakage-induced discharge (Section 5).

4. Proposed Architecture

The proposed architecture is based on a traditional charge-redistribution SAR ADC [45], suitably modified to hold different signal samples at the same time. Both the original SAR and the proposed modifications are illustrated in Figure 5.

A straightforward implementation of a l-bit SAR converter requires capacitors ranging from C_u to $2^{l-1}C_u$. In more convenient realizations, the binary-weighted array is used only for the most significant bits (MSBs), while the least significant ones (LSBs) are computed exploiting either a secondary binary-weighted array, with an attenuation capacitor in between, or a C-2C sub-array [46]. The figure refers to an implementation

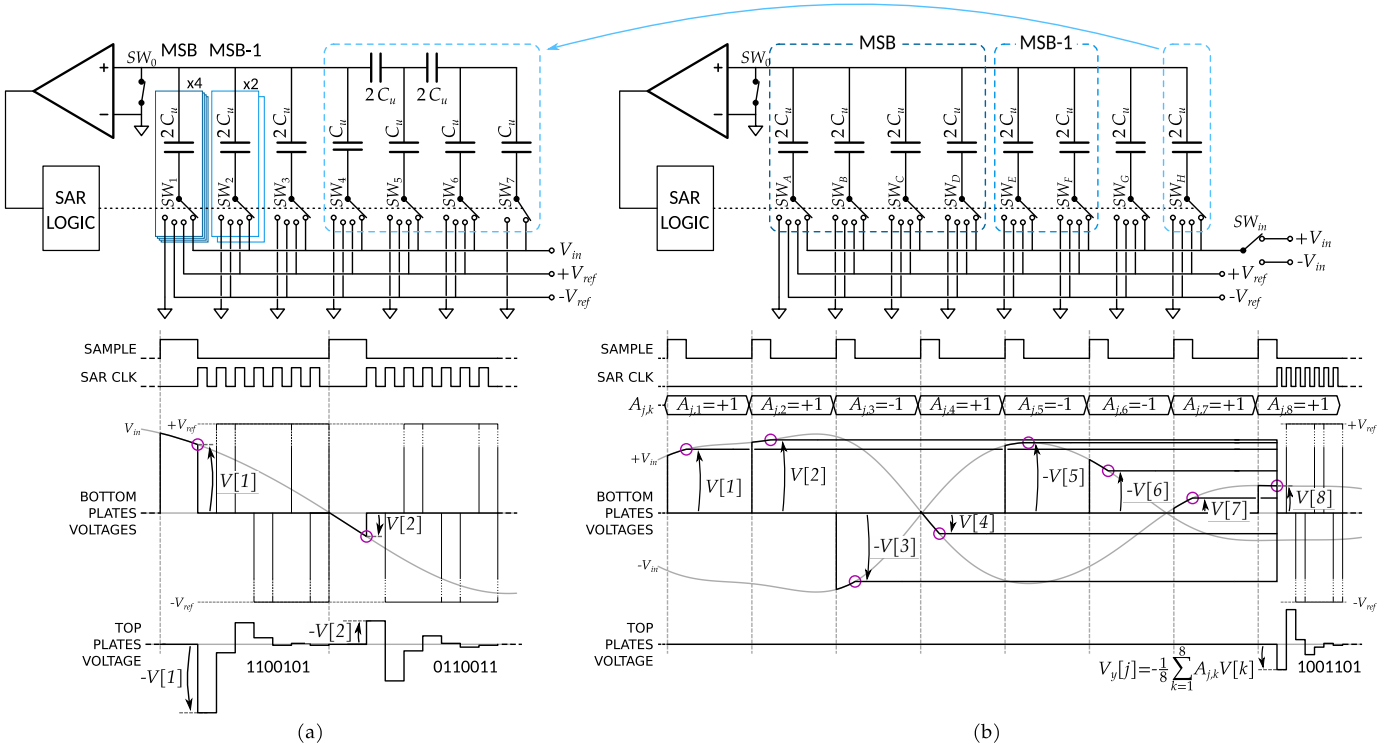


Figure 5: (a) Schematic and working principle of a traditional charge-redistribution SAR ADC with a 4-bit binary-weighted array, and a 2-bit C-2C array. Notice that, having the comparator reference in the middle of the range $[-V_{ref}, +V_{ref}]$, the overall ADC gains an extra bit, reaching a total of 7 bits. (b) Schematic and working principle of the CS-based acquisition system, based on a charge-redistribution SAR ADC, proposed in this paper. The structure has $C_{tot} = 16C_u$, $n_b = 8$ and $C_h = 2C_u$. The $+V_{ref}$ and $-V_{ref}$ levels in the timing diagrams are not to scale.

where the binary-weighted array and the C-2C structure compute 4 bits and 2 bits, respectively.

The behavior of the architecture, either when working as a SAR converter or as a CS-based acquisition system, is described in the following.

4.1. Traditional SAR converter

The schematic of a traditional switched-capacitor SAR ADC is illustrated in Figure 5(a). The figure also includes a timing-diagram highlighting the behavior of the converter. Note that the time-domain dynamics of a SAR (hence the timing-diagram) are, to first order, independent of the array implementation, and would be unchanged if considering a binary-weighted secondary array, or even a single main array exclusively.

To sample the input, the top plates of the capacitors are grounded by SW_0, while the bottom plates are *all* connected to the input signal V_{in}, tracking it. Then, SW_0 opens, so that the input voltage is sampled, and the total charge accumulated in the capacitors is fixed and remains constant until the end of the conversion. Also, all other switches move to the ground position and the opposite of the sampled voltage is observed at the non-inverting input of the comparator. At this point, all switches from SW_1 to SW_6 move sequentially either to $+V_{ref}$ or $-V_{ref}$, depending on the output of the comparator. By means of capacitive voltage division, the variations observed on the top plates become increasingly smaller, either because the capacitors involved are inherently smaller, or, in the case of the C-2C subarray, because the attenuation from each input depends on the position within the array. Over time, the voltage at the non-inverting input of the comparator tends towards its reference voltage. This way, the converter finds the best digital approximation of the sampled input voltage.

4.2. CS-based converter

The proposed architecture modifies the behavior of the traditional converter when sampling occurs, as shown in Figure 5(b). Through additional switches, allowing a finer-grained control of the largest array capacitors, and the presence of SW_{in} at the input, equation (3) can be implemented directly in the analog domain. The structure depicted in the figure corresponds to one row of the sensing matrix A and is used to compute the j-th measurement y_j.

Let us consider the case $n_b = 8$, as in Figure 5(b), i.e., it is necessary to acquire 8 samples from the input, each modulated by a ± 1 coefficient. Their values have to be stored until the end of the acquisition window, and then summed to compute y_j.

The modulation is achieved by the switch SW_{in}, which selects either $+V_{in}$ or $-V_{in}$, inherently available in differential implementations. At the same time, the original capacitive array can be used as n_b identical sampling capacitors, each of size $C_h = C_{tot}/n_b$, so that n_b samples of the modulated input can be stored independently (in the figure, with $C_h = 2C_u$ we are able to obtain 8 hold cells). This operation requires the decomposi-

tion of the largest capacitors in Figure 5(a), i.e., $8C_u = 4 \times 2C_u$ and $4C_u = 2 \times 2C_u$, into smaller elements, hence the requirement for additional switches. At the same time, the smallest capacitors have to be driven simultaneously, so that their combined sampling capacitance is equal to C_h.

Before each sampling instant, only one among the switches SW_A to SW_H connects the input to a sampling capacitor. By the end of the acquisition window, each of them will hold a value equal to $A_{j,k}V[k]$, with $k = 1, 2, \ldots, 8$, in agreement with the notation of the figure.

Finally, SW_0 opens, and all other switches move to the ground position. All the sampling capacitors are thus connected in parallel, sharing the accumulated charge and generating a voltage level at the input of the comparator equal to

$$V_y[j] = -\sum_{k=1}^{n_b} \frac{C_h A_{j,k} V[k]}{n_b C_h} = -\frac{1}{n_b} \sum_{k=1}^{n_b} A_{j,k} V[k]$$

Apart from the scaling factor $-1/n_b$, this is equivalent to the measurement y_j described by (3).

The acquisition phase is now complete and the A/D conversion can start by logically reconfiguring the capacitive array in its original shape. The largest of the binary-weighted capacitors, initially split into smaller elements, can be re-obtained by driving them simultaneously. Equivalently, the smallest capacitors, jointly driven during acquisition, can be controlled independently.

Note that, without entering into details, the C-2C sub-array requires particular care because of its sensitivity to parasitic loading of the internal isolated nodes. Hence it is preferable not to change its topology during the acquisition phase, using it as a whole. This determines a minimum value for C_h, which has to be $C_h \geq C_u$.

4.3. Comparison with previous solutions

The main difference with respect to previously proposed integrated CS solutions is the lack of additional active devices such as operational amplifiers or integrators, or any other block requiring active biasing, with respect to a Nyquist-rate converter. The proposed circuit, in fact, only requires additional switches to drive independently the elements of the largest capacitors.

A summary comparing integrated solutions recently proposed in the literature is included in Table 2. For each architecture considered, a simplified schematic highlights the additional hardware blocks required with respect to a straightforward Nyquist-rate approach, given by the direct A/D conversion of the input signal samples.

In [10] a sub-Nyquist rate receiver for radar pulse signal is presented. Authors of [11] presented a data acquisition front-end for RF communication assuming a multi-tone input signal. Both solutions embed a passive mixer that exploits $A_{j,k} \in \{-1, 1\}$ by exchanging the two wires of the differential input signal. However, they also require a power-hungry continuous-time integrator.

Solutions designed for lower bandwidths typically rely on a switched-capacitors integrator architecture. In [7] an analog front-end for ECG signals is presented, with a passive mixer designed for approximating $A_{j,k} \in \mathbb{R}$. It exploits the differential architecture to implement the sign change, and a 6-bit multiplying DAC embedded in the integrator. In [6] the target application is given by intracranial EEG signals, and also in this case a passive mixer is adopted, implementing $A_{j,k} \in \{0, 1\}$ by means of simple pass-transistors. The last considered architecture is that described in [9], where a passive mixer is obtained constraining $A_{j,k} \in \{-1, +1\}$ and exploiting the fully differential architecture for sign inversion via pass-transistors. Anyway, all these solutions require an operational amplifier as additional active circuit to execute the integration.

In Table 2, we included the energy required by the additional active elements to compute a single measurement. We have also indicated the signal bandwidth and the circuit noise performance (estimated as the resolution of the ADC used to sample the signal). Of course, a fair comparison would look also at other CS-related capabilities (mainly, the possibility to work at different levels of compression, controlling m, that indeed depends on the architecture). The aim of this comparison is only to identify, in several architectures, the additional power required with respect to the that of a mere A/D converter circuit.

The last row in the table highlights the fact that the architecture only requires the presence of the ADC, with the addition of a few switches and their driving logic. The advantage is the implementation not only of the mixer, but also of the integrator using exclusively passive circuitry, leading to negligible additional energy.

As a reference case, a 10-bit SAR converter [47] in a 90 nm technology, employing the conversion technique described in the previous section (also known in the literature as *VCM-based method*), shows a power consumption of 3 mW at 100 MS/s, equivalent to 30 pJ per conversion, with more than 9 effective bits. This energy is almost negligible when compared with the additional energy required by all previous solutions in Table 2.

Since, in the proposed architecture, the logic of the conversion is unchanged and the only difference is that of a counting mechanism to keep track of the samples to be acquired in a window, it is reasonable to assume that the extra energy per conversion is negligible with respect to that of the original ADC.

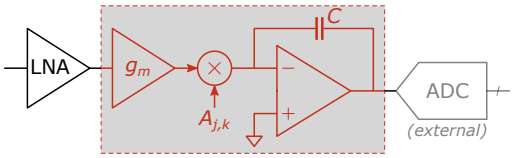
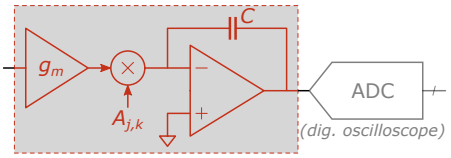
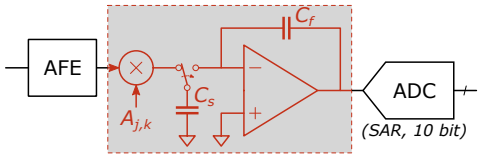
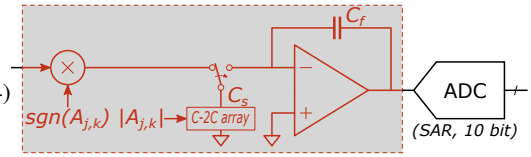
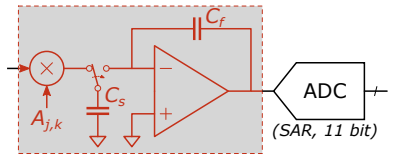
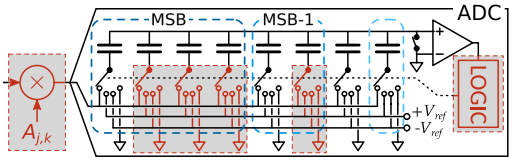
5. Compensation of Leakage Currents

When the target application involves low-frequency signals, a major source of noise in (3) is given by leakage currents discharging the hold capacitors. Since the intermediate modulated samples have to be stored until the end of the acquisition window this leads to a degradation of their linear combination, i.e., the measurement. Components of such currents origin both from the reverse-biased drain/source diffusions of the transistors used as switches, and from subthreshold conduction.

Indeed, this drawback has to be considered in any analog implementation of a CS-based acquisition chain, especially for the proposed architecture where the goal is to have the capacitive array as small as that embedded in a typical SAR ADC.

Coping with the discharge by a straightforward increase of C_h is unfeasible, as highlighted in Table 3. The values show the total array capacitance that, for a given n_b (hence a given du-

Table 2: Summary of solutions proposed to implement a CS encoder exploiting (3), highlighting the additional hardware required with respect to a straightforward Nyquist approach (i.e., A/D conversion of the input signal), along with the power consumption of this additional hardware only. To allow a fair comparison, the energy per measurement and the energy per measurement normalized for n (or n_b) has been also indicated. The resolution has been estimated by the effective resolution of the connected ADC.

Ref.	Schematic	Brief description	Figures of Merit	Res.	Tech.
[10] (2012)		Continuous-time g_mC integrator with passive mixer ($A_{j,k} \in \{-1, 1\}$ exploiting fully differential implementation)	2 GHz BW, 506.4 mW ($n = 100$, $m = 8$) 1.58 nJ/conv 15.8 pJ/conv/n	–[1]	90 nm
[11] (2012)		Continuous-time g_mC integrator with passive mixer ($A_{j,k} \in \{-1, 1\}$ exploiting fully differential implementation)	500 MHz BW, 30 mW 0.083 nJ/conv ($n_b = 22$, $m_b = 8$)[2] 3.75 pJ/conv/n_b	8 bit[2]	90 nm
[6] (2014)		Switched-capacitors integrator with passive mixer ($A_{j,k} \in \{0, 1\}$ with simple pass-transistors implementation)	10 kHz/m BW[3], 8.4 µW ($n = 16$) 0.42 nJ/conv	9.2 bit	180 nm
[7] (2014)		Switched-capacitors integrator with passive mixer ($A_{j,k} \in \mathbb{R}$, sign of $A_{j,k}$ implemented by exploiting differential implementation, module via C-$2C$ modulation of sampling capacitance)	1 kHz BW, 1.8 µW ($n = 256$, $m = 64$) 3.58 nJ/conv 14 pJ/conv/n	6.5 bit	130 nm
[9] (2016)		Switched-capacitors integrator with passive mixer ($A_{j,k} \in \{-1, 1\}$ exploiting fully differential implementation)	60 kHz BW, 430 µW ($n = 128$, $m = 16$) 13.2 nJ/conv 216 pJ/conv/n	9.0 bit	180 nm
This work		Integration by passive charge redistribution in the capacitive array of the ADC. Passive mixer ($A_{j,k} \in \{-1, 1\}$ exploiting fully differential implementation)	Negligible extra energy	-	-

[1] An external ADC is used, whose resolution is not indicated.

[2] A block-diagonal approach is used. Furthermore, a digital oscilloscope is used for sampling data, so that a standard 8 bit resolution is assumed.

[3] Due to resource sharing, the maximum bandwidth of the input signal decrease with the number m of measurements. The architecture do not integrate over time but over space, so the energy of the measurement do not depend on n.

[4] Includes the power consumption of the ADC (power consumption of the analog block only not declared by authors).

Table 3: Total array capacitance C_{tot} guaranteeing a degradation of the ARSNR of less than 3 dB with respect to ideal sensing (solid lines in Figure 4), when a constant-current leakage discharge is considered. The values are computed for CR=2 and the leakage parameters are the same of Figure 4. The capacitance is approximately proportional to $(n/n_b)^{-2.5}$.

n_b	n/n_b	C_{tot}
128	1	500 µF
32	4	25 µF
16	8	3 µF
8	16	350 nF

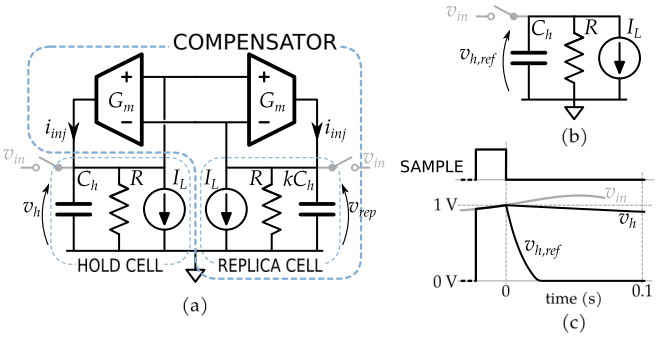


Figure 6: (a) Leakage-compensation circuit, to be placed around every hold capacitor in Figure 5(b). The resistor and current source model the effect of turned-off switches and do not correspond to real components. (b) Uncompensated hold cell. (c) Waveforms obtained using the parameters indicated in Section 5.

ration of a single measurement), guarantees 3 dB of loss with respect to the corresponding ideal ARSNR (where no degradation is considered in the signal acquisition). The values are indeed impractical and ask for some other means to limit the loss of charge.

A robust solution for the compensation of leakage currents is proposed in [48]. Therein the authors describe and validate the topology depicted in Figure 6, achieving a 9.5 mV/s residual drift of the voltage sampled on a 0.5 pF capacitor with a total current absorption of the active blocks around 38 nA.

The circuit should be placed around every hold capacitor of the array, with the exclusion of the one made by the simultaneous driving of the LSB capacitors, since A/D conversion starts immediately after their sample is acquired. As a result, $n_b - 1$ replicas are required.

Without compensation, once a sample has been stored in a capacitor, its bottom-plate switch is opened. Being it implemented by several MOS transistors, the accumulated charge starts to flow through the node because of their leakage, with a direction which depends on the exact topology of the switch. The discharge is modeled by placing a current source I_L and the switch off-resistance R in parallel to every hold capacitor.

The compensator works by recreating the same switch topology around a down-scaled replica of the hold capacitor, having value kC_h. Being smaller, the rate-of-change of its voltage will be higher. The resulting difference $v_h - v_{rep}$ determines a current, injected equally in both capacitors by means of a couple of identical transconductors. Such a compensating current reduces

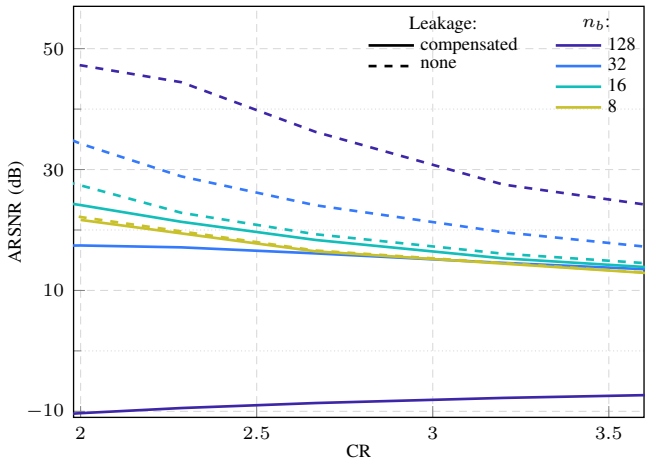


Figure 7: Performance of a CS-based signal processing chain for the considered synthetic ECG signals, when leakage compensation is included. Solid lines refer to the proposed architecture with leakage compensation; dashed lines to an ideal system with no leakage.

the leakage-induced variation.

Neglecting for a moment the resistive component, the behavior of the compensator can be described as follows. With reference to Figure 6(a), let us consider positive leakage currents, discharging both capacitors over time. Since both cells sample the same voltage, their voltage difference is initially null. As this difference increases linearly with time, the injected compensation current increases as well. The net current flowing in each capacitor is therefore decreased until an equilibrium condition is reached. At this point the leakage currents are canceled exactly by the action of feedback and the sampled voltage is held indefinetly.

The complete model accounting for the effects of R actually shows that v_h, the voltage across the hold cell, decays over time:

$$v_h(t) = v_h(0)\exp\left(-\frac{t}{\tau}\right)u(t) - RI_L\left[1-\exp\left(-\frac{t}{\tau}\right)u(t)\right] \quad (4)$$

with $\tau = G_m R^2 C_h(1-k)$ and $u(t)$ the unitary step function. Since the initial voltage in any case cannot be preserved, the elements of the feedback loop have to be sized to constrain the drop to an acceptable value. The discharge time constant depends in fact on the gain factor $G_m R(1-k)$, which can be designed to slow down the variation as much as required, as exemplified in Figure 6(c). Considering that the hold time is not equal among the capacitors, the values of G_m can be tuned individually, leading to some power savings.

As a final comment, like in any feedback loop, stability has to be guaranteed. The complete analysis will not be presented here, suffice it to say that the hold and replica cells behave as a low-pass filter whose pole frequency depends on $1/k$. Considering the singularities in the transconductor transfer function, a small k, desirable to minimize the overhead introduced by the replica cell, may indeed lead to instability.

Figure 7 compares the reconstruction quality obtained for an ideal sensing to the one where the acquisition process suffers

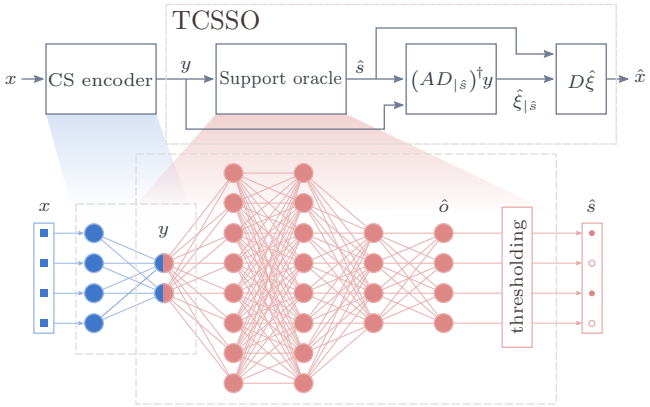


Figure 8: Architecture the TCSSO framework, including the CS encoder since it is optimized along with the support oracle DNN during the training phase. The recovered signal $\hat{x}$ is computed using the estimated $\hat{s}$.

from leakage, with the presence of the compensators. The discharge is modeled as in (4), with parameters $G_m = 100\,\mu\text{S}$, $R = 1\,\text{G}\Omega$, $k = 0.1$ and $I_L = 300\,\text{pA}$. Notice that, for simplicity, the transconductance is equal for all compensators.

Considering a constant total array capacitance of 16 pF, indeed C_h for $n_b = 128$ is too small to preserve the sampled values even in presence of the compensator. As n_b is decreased, the structure can better manage the signal window and performance matches more closely the ideal one. In particular, the curves for $n_b = 8$ and $n_b = 16$ are quite close to the reference ones, with the latter obtaining slightly better results. Such values of n_b are indeed valid options for the design of the capacitive array.

6. Decoding by TCSSO with Short Windowing

The application of the TCSSO approach proposed in [14] to the architecture considered in this paper is not straightforward. In this section we first introduce the original method, then we show how to modify it to handle antipodal block-diagonal sensing matrices. Finally we apply the modified TCSSO to the case study introduced in section 3.4 and provide numerical results.

6.1. The TCSSO approach

The architecture of the TCSSO approach is depicted in Figure 8 and is described in the following.

The computation of the reconstructed signal $\hat{x}$ starts from a set of measurements y coming from the CS encoder. The vector y is fed into a DNN, named "support oracle", trained to predict the support $\hat{s}$ of the input signal that generated y. Once the support $\hat{s}$ has been estimated, it is possible to reconstruct the original input signal from y ,i.e., to invert (1), using standard linear algebra.

Consider $\xi_{|\hat{s}} \in \mathbb{R}^{\kappa}$ as the vector containing only the non-null elements of ξ, whose positions are identified by $\hat{s}$. Similarly, we can define $D_{|\hat{s}} \in \mathbb{R}^{n\times\kappa}$ as the matrix containing only the columns of D selected by $\hat{s}$. Neglecting the noise term ν, the sensing equation (1) can be rewritten as

$$y = AD_{|\hat{s}}\,\xi_{|\hat{s}} \tag{5}$$

While the inversion of (1) gives rise to an underdetermined system of equations whose solution is a complex task involving a minimization problem (2), the inversion of (5) corresponds to an overdetermined system. Several known methods, such as the least-squares based ones, can be used to find its approximate solution. In [14] the authors consider the Moore-Penrose pseudoinverse operator $(\cdot)^{\dagger}$, so that

$$\hat{\xi}_{|\hat{s}} = (AD_{|\hat{s}})^{\dagger}\,y \tag{6}$$

and the original signal is finally reconstructed either as $\hat{x} = D_{|\hat{s}}\,\hat{\xi}_{|\hat{s}}$ or as $\hat{x} = D\hat{\xi}$.

Figure 8 also details the internal structure of both the CS encoder and the support oracle DNN. The encoder is considered as part of the neural network only during the training phase, to generate the optimal A. It *emulates* the linear projection $y = Ax$, having n inputs (i.e., the dimensionality of x) and m outputs (dimensionality of y). With no bias, using as interconnection weights the actual elements of A, and employing a linear activation function, its behavior is equivalent to (1).

The actual oracle starts from the second layer of nodes, which are shared with the encoder during training. It is built with m inputs and three hidden layers with $2n$, $2n$ and n neurons each, and ReLU activation functions. The n outputs use a sigmoid activation function $\alpha(a) = 1/(1 + e^{-a})$ and generate the output vector $\hat{o} \in \mathbb{R}^n$, which can be interpreted as the probabilities of the coefficients of ξ being non-zero. The divined support $\hat{s}$ is obtained from $\hat{o}$ by simply thresholding it element-wise, so that (ideally) its κ largest elements are set to 1 and the rest to 0 [4].

6.2. TCSSO with antipodal block-diagonal matrices

In the TCSSO approach, as well as in many other frameworks where the CS reconstruction relies on a DNN which is jointly trained with the sensing matrix A [38, 41], the key point is to model the matrix-vector product required by the CS acquisition in (1) as an extra layer of the DNN.

It is evident that any constraint on A (e.g., forcing it to be antipodal, ternary or block-diagonal) corresponds to an equivalent constraint on such a layer. In reality, when the neural network is used for inference, sensing matrices of any kind can be used with the support oracle. However, during the training phase, obtaining an optimized A with a specific structure requires particular care.

In the original TCSSO framework [14], a full antipodal sensing matrix $A^{\pm} \in \{-1, +1\}^{m\times n}$ is considered. The algorithm used for the optimization of the neural network is the Stochastic Gradient Descent (SGD), which consists of two phases, namely forward- and back-propagation. The convergence of the SGD algorithm requires tiny variations of A, which are not possible if training is performed directly on the antipodal matrix $A^{\pm}$, whose values are either -1 or 1. Therefore a real valued $A^{\mathbb{R}} \in \mathbb{R}^{m\times n}$ sensing matrix is employed since, during back-propagation, its values can be finely adjusted to minimize the

[4] In practice the number of ones in the recovered support is on average no less than κ.

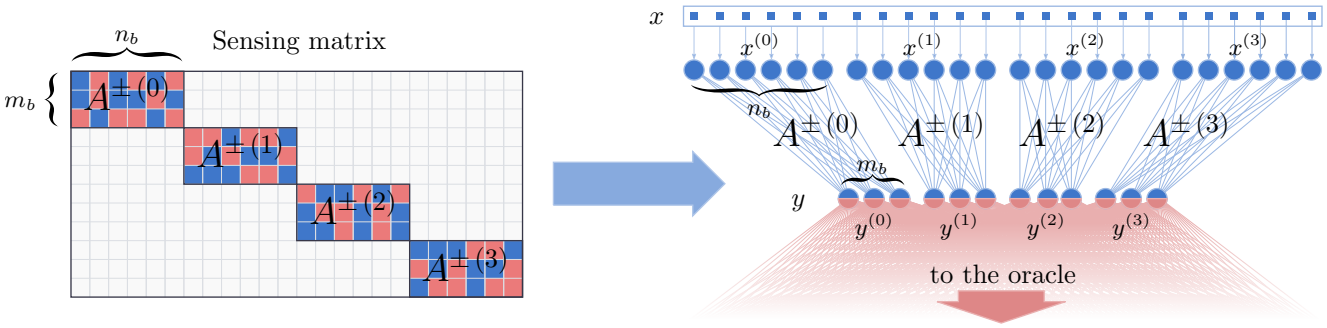


Figure 9: Equivalent views of the CS encoder in the case of block-diagonal sensing. (left) Antipodal block-diagonal sensing matrix A. (right) DNN layer modeled as several parallel, independent, fully-connected sub-layers. Each block $A^{\pm(l)}$ is mapped to the weight matrix of a sub-layer. The latter description is used for the optimization of the sensing matrix during the training phase of the support oracle. In the example, $n_b = 6$ and $m_b = 3$, with CR = 2.

error at the output. In the forward-propagation phase, the corresponding antipodal matrix $A^{\pm}$ is extracted from $A^{\mathbb{R}}$ by evaluating the sign of each matrix element.

At the same time, if the sensing matrix has to be block-diagonal, we may actually observe that such a matrix can be split into smaller antipodal matrices $A^{\pm(l)} \in \{-1, 1\}^{m_b \times n_b}$ as in Figure 9, where l is the index of the l-th block. Each elementary block acts only on a portion of the input signal $x^{(l)} \in \mathbb{R}^{n_b}$, contributing to a subset $y^{(l)} \in \mathbb{R}^{m_b}$ of the measurements vector y. Hence, the sensing operation becomes

$$y_j^{(l)} = \sum_{k=1}^{n_b} A_{j,k}^{\pm(l)} \, x_k^{(l)}. \quad \text{for} \begin{cases} l = 1, \ldots, n/n_b \\ j = 1, \ldots, m_b \end{cases}$$

In other words, we have decomposed the encoding process (3) into n/n_b independent and parallel operations, each of them defined by an antipodal matrix $A^{\pm(l)}$. This is illustrated in Figure 9. From the point of view of the DNN, the input layer is no more a single, fully-connected layer, but it is the composition of n/n_b mutually independent, fully-connected sub-layers. All the zeroes of the sensing matrix and the corresponding interconnections between neurons are thus neglected altogether.

Equivalently to what is done in the original framework, here the SGD algorithm is applied to multiple full-precision matrices $A^{\mathbb{R}(l)} \in \mathbb{R}^{m \times n}$. The corresponding antipodal sensing matrices $A^{\pm(l)}$ are obtained by extracting the sign from each element of every $A^{\mathbb{R}(l)}$. Finally, the desired antipodal block-diagonal matrix $A^{\pm}$ is composed by a proper arrangement of the individual blocks on the main diagonal.

6.3. Numerical results with the modified TCSSO

Simulations using the same input signals defined in Section 3.4 prove that the TCSSO approach is extremely effective. When properly modified to work with antipodal block-diagonal sensing matrices, it ensures better performance with respect to the BPDn approach boosted by the rakeness optimization. A time-domain comparison of the two decoding is shown in Figure 10, where the techniques are applied to the signals depicted in Figure 3. The behavior is noticeably improved at the window boundaries, which are the largest noise contributors.

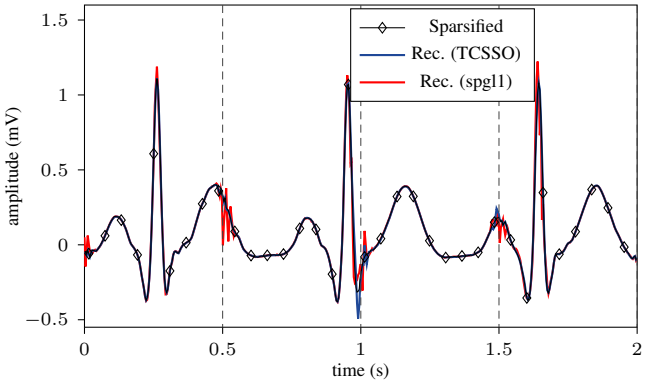


Figure 10: Reconstruction of the sparsified ECG waveform depicted in Figure 3, using an ideal setup (no leakage) with $n_b = 8$ and CR = 2.7. The different decoders result in a 16 dB RSNR for SPGL1 and 31 dB for the TCSSO.

The ideal setup without the effects of the leakage currents is considered in details in Figure 11(a). Similarly to what is shown in Figure 4 for the BPDn reconstruction, a higher value of n_b results in an improved performance. Furthermore, for all considered values of CR, reconstruction quality using the TCSSO is higher with respect to the reference case given by the SPGL1 algorithm with rakeness-optimized sensing matrices.

In Figure 11(b) measurements are degraded by leakage. It can be readily observed that TCSSO achieves up to 20 dB of increased ARSNR with respect to the reference case. As already described for the curves Figure 7, the value of n_b sets a trade-off. However, whereas for a BPDn-based reconstruction the optimal performance is obtained for $n_b = 16$, using TCSSO the optimum is found for $n_b = 8$. This is extremely important from the hardware point of view, since it allows at the same time an improvement in performance and a reduction of the complexity of the architecture, requiring a coarser decomposition of the SAR capacitive array (reduced number of hold capacitors C_h).

7. Conclusion

An innovative switched-capacitor SAR architecture for CS-based acquisitions in the analog domain has been presented. Two techniques to make it practical have been analyzed, namely

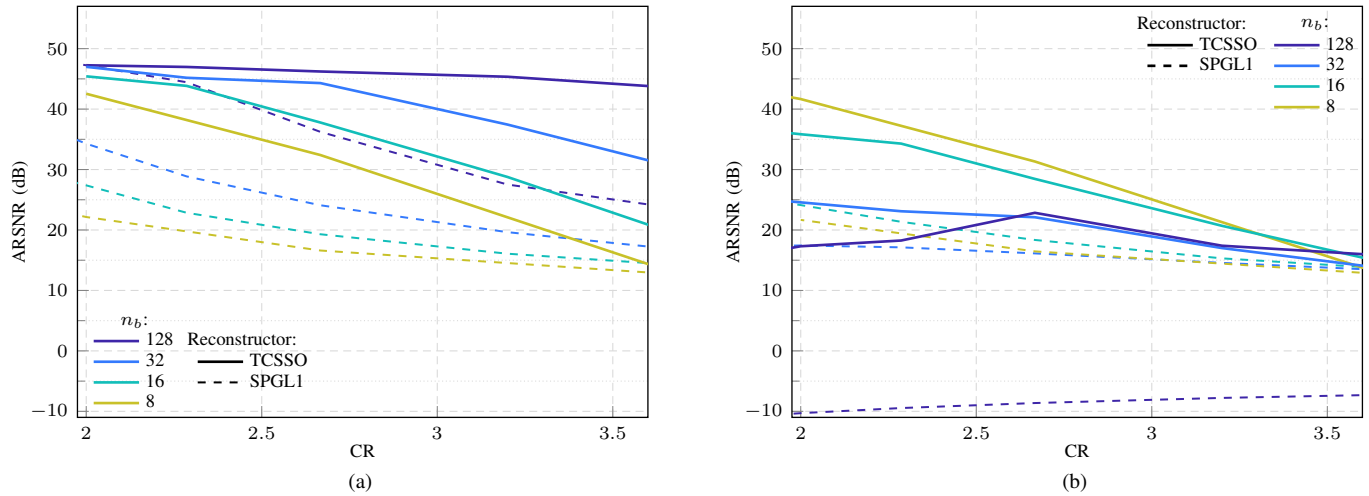


Figure 11: Performance of a CS-based signal processing chain expressed in terms of ARSNR as a function of CR, for synthetic ECG signals at approx 60 beats/s, with $f_s = 256\,\text{Hz}$ and $n = 128$. Solid lines refer to the TCSSO reconstruction, dashed lines to the SPGL1 algorithm. (a) Ideal sensing (b) Sensing corrupted by leakage, with the presence of the compensator introduced in Section 5 and the same parameters used to obtain Figure 7.

using an antipodal block-diagonal sensing matrix and introducing a hardware leakage compensator. Performance gains of such techniques have been validated through algorithmic simulations with real-world parameters, highlighting their importance for the feasibility of the architecture. Furthermore, it has been shown that the recently proposed TCSSO technique allows a significant boost in reconstruction performance and a reduction of the hardware complexity.

References

[1] E. J. Candes, J. Romberg, T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information, IEEE Trans. on Inf. Theory 52 (2) (2006) 489–509. doi:10.1109/TIT.2005.862083.

[2] D. L. Donoho, Compressed Sensing, IEEE Trans. on Inf. Theory 52 (4) (2006) 1289–1306. doi:10.1109/TIT.2006.871582.

[3] M. Lustig, D. L. Donoho, J. M. Santos, J. M. Pauly, Compressed sensing MRI, IEEE Signal Processing Magazine 25 (2) (2008) 72–82. doi:10.1109/MSP.2007.914728.

[4] E. J. Candès, M. B. Wakin, An introduction to compressive sampling, IEEE Signal Processing Magazine 25 (2) (2008) 21–30. doi:10.1109/MSP.2007.914731.

[5] F. Chen, A. P. Chandrakasan, V. M. Stojanović, Design and Analysis of a Hardware-Efficient Compressed Sensing Architecture for Data Compression in Wireless Sensors, IEEE J. Solid-State Circuits 47 (3) (2012) 744–756. doi:10.1109/JSSC.2011.2179451.

[6] M. Shoaran, M. H. Kamal, C. Pollo, P. Vandergheynst, A. Schmid, Compact low-power cortical recording architecture for compressive multichannel data acquisition, IEEE Trans. Biomed. Circuits Syst. 8 (6) (2014) 857–870. doi:10.1109/TBCAS.2014.2304582.

[7] D. Gangopadhyay, E. G. Allstot, A. M. R. Dixon, K. Natarajan, S. Gupta, D. J. Allstot, Compressed sensing analog front-end for biosensor applications, IEEE J. Solid-State Circuits 49 (2) (2014) 426–438. doi:10.1109/JSSC.2013.2284673.

[8] J. Zhang, Y. Suo, S. Mitra, S. P. Chin, S. Hsiao, R. F. Yazicioglu, T. D. Tran, R. Etienne-Cummings, An efficient and compact compressed sensing microsystem for implantable neural recordings, IEEE Trans. Biomed. Circuits Syst. 8 (4) (2014) 485–496. doi:10.1109/TBCAS.2013.2284254.

[9] F. Pareschi, P. Albertini, G. Frattini, M. Mangia, R. Rovatti, G. Setti, Hardware-Algorithms Co-Design and Implementation of an Analog-to-Information Converter for Biosignals Based on Compressed Sensing, IEEE Trans. Biomed. Circuits Syst. 10 (1) (2016) 149–162. doi:10.1109/TBCAS.2015.2444276.

[10] J. Yoo, S. Becker, M. Loh, M. Monge, E. Candès, A. Emami-Neyestanak, A 100MHz-2GHz 12.5x sub-Nyquist rate receiver in 90nm CMOS, in: 2012 IEEE radio frequency integrated circuits symposium, 2012, pp. 31–34. doi:10.1109/RFIC.2012.6242225.

[11] X. Chen, E. A. Sobhy, Z. Yu, S. Hoyos, J. Silva-Martinez, S. Palermo, B. M. Sadler, A sub-nyquist rate compressive sensing data acquisition front-end, IEEE Journal on Emerging and Selected Topics in Circuits and Systems 2 (3) (2012) 542–551. doi:10.1109/JETCAS.2012.2221531.

[12] D. Bellasi, M. Crescentini, D. Cristaudo, A. Romani, M. Tartagni, L. Benini, A broadband multi-mode compressive sensing current sensor SoC in 0.16 μ m CMOS, IEEE Trans. Circuits Syst. I, Reg. Papers 66 (1) (2019) 105–118. doi:10.1109/TCSI.2018.2846573.

[13] C. Paolino, F. Pareschi, M. Mangia, R. Rovatti, G. Setti, A Practical Architecture for SAR-based ADCs with Embedded Compressed Sensing Capabilities, in: 15th conference on Ph.D. research in microelectronics and electronics (PRIME), 2019, pp. 133–136. doi:10.1109/PRIME.2019.8787816.

[14] M. Mangia, L. Prono, A. Marchioni, F. Pareschi, R. Rovatti, G. Setti, Deep neural oracles for short-window optimized compressed sensing of biosignals, IEEE Trans. Biomed. Circuits Syst. (2020). doi:10.1109/TBCAS.2020.2982824.

[15] T. M. Cover, J. A. Thomas, Elements of information theory, 2nd Edition, wiley series in telecommunications and signal processing, Wiley-Interscience, New York, NY, USA, 2006.

[16] E. J. Candes, T. Tao, Decoding by linear programming, IEEE Trans. on Inf. Theory 51 (12) (2005) 4203–4215. doi:10.1109/TIT.2005.858979.

[17] J. A. Tropp, A. C. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit, IEEE Trans. on Inf. Theory 53 (12) (2007) 4655–4666. doi:10.1109/TIT.2007.909108.

[18] T. Blumensath, M. E. Davies, Iterative thresholding for sparse approximations, Journal of Fourier Analysis and Applications 14 (5) (2008) 629–654. doi:10.1007/s00041-008-9035-z.

[19] D. Needell, J. A. Tropp, CoSaMP: Iterative signal recovery from incomplete and inaccurate samples, Applied and Computational Harmonic Analysis 26 (3) (2009) 301–321. doi:10.1016/j.acha.2008.07.002.

[20] S. Becker, J. Bobin, E. Candès, NESTA: A fast and accurate first-order method for sparse recovery, SIAM Journal on Imaging Sciences 4 (1) (2011) 1–39. doi:10.1137/090756855.

[21] E. Van den Berg, M. P. Friedlander, Sparse optimization with least-squares constraints, SIAM J. on Optimization 21 (4) (2011) 1201–1229. doi:10.1137/100785028.

[22] E. van den Berg, M. P. Friedlander, SPGL1: A solver for large-scale sparse reconstruction (Apr. 2015).

URL https://www.cs.ubc.ca/~mpf/spgl1/

[23] J. Haboba, M. Mangia, F. Pareschi, R. Rovatti, G. Setti, A pragmatic look at some compressive sensing architectures with saturation and quantization, IEEE Journal on Emerging and Selected Topics in Circuits and Systems 2 (3) (2012) 443–459. doi:10.1109/JETCAS.2012.2220392.

[24] J. Romberg, Compressive sensing by random convolution, SIAM J. Imaging Sci. 2 (4) (2009) 1098–1128. doi:10.1137/08072975X.

[25] A. Harms, W. U. Bajwa, R. Calderbank, A constrained random demodulator for sub-nyquist sampling, IEEE Trans. Signal Process. 61 (3) (2013) 707–723. doi:10.1109/TSP.2012.2231077.

[26] A. Ravelomanantsoa, H. Rabah, A. Rouane, Simple and efficient compressed sensing encoder for wireless body area network, IEEE Trans. Instrum. Meas. 63 (12) (2014) 2973–2982, tex.ids: Ravelomanantsoa_TIM2014. doi:10.1109/TIM.2014.2320393.

[27] M. Mangia, D. Bortolotti, F. Pareschi, A. Bartolini, L. Benini, R. Rovatti, G. Setti, Zeroing for HW-efficient compressed sensing architectures targeting data compression in wireless sensor networks, Microprocessors and Microsystems 48 (2017) 69–79. doi:10.1016/j.micpro.2016.09.007.

[28] M. Mangia, F. Pareschi, V. Cambareri, R. Rovatti, G. Setti, Rakeness-based design of low-complexity compressed sensing, IEEE Trans. Circuits Syst. I, Reg. Papers 64 (5) (2017) 1201–1213. doi:10.1109/TCSI.2017.2649572.

[29] M. Mangia, F. Pareschi, V. Cambareri, R. Rovatti, G. Setti, Adapted compressed sensing for effective hardware implementations: A design flow for signal-level optimization of compressed sensing stages, Springer International Publishing, 2018.

[30] N. Cleju, Optimized projections for compressed sensing via rank-constrained nearest correlation matrix, Applied and Computational Harmonic Analysis 36 (3) (2014) 495–507. doi:10.1016/j.acha.2013.08.005.

[31] J. Xu, Y. Pi, Z. Cao, Optimized projection matrix for compressive sensing, EURASIP Journal on Advances in Signal Processing (2010) 560349doi:10.1155/2010/560349.

[32] M. Mangia, R. Rovatti, G. Setti, Rakeness in the design of analog-to-information conversion of sparse and localized signals, IEEE Trans. Circuits Syst. I, Reg. Papers 59 (5) (2012) 1001–1014. doi:10.1109/TCSI.2012.2191312.

[33] A. Marchioni, C. H. Pimentel-Romero, F. Pareschi, M. Mangia, R. Rovatti, G. Setti, Resource redistribution in internet of things applications by compressed sensing: A survey, in: IEEE international symposium on circuits and systems (ISCAS), 2018, pp. 1–5. doi:10.1109/ISCAS.2018.8351891.

[34] A. Mirrashid, A. A. Beheshti, Compressed remote sensing by using deep learning, in: 9th international symposium on telecommunications (IST), 2018, pp. 549–552. doi:10.1109/ISTEL.2018.8661112.

[35] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, A. Ashok, ReconNet: non-iterative reconstruction of images from compressively sensed measurements, in: IEEE conference on computer vision and pattern recognition (CVPR), 2016, pp. 449–458. doi:10.1109/CVPR.2016.55.

[36] A. Mousavi, R. G. Baraniuk, Learning to invert: signal recovery via deep convolutional networks, in: IEEE international conference on acoustics, speech and signal processing (ICASSP), 2017, pp. 2272–2276. doi:10.1109/ICASSP.2017.7952561.

[37] W. Shi, F. Jiang, S. Zhang, D. Zhao, Deep networks for compressed image sensing, in: IEEE international conference on multimedia and expo (ICME), 2017, pp. 877–882. doi:10.1109/ICME.2017.8019428.

[38] A. Mousavi, A. B. Patel, R. G. Baraniuk, A deep learning approach to structured signal recovery, in: 3rd annual allerton conference on communication, control, and computing (allerton), 2015, pp. 1336–1343. doi:10.1109/ALLERTON.2015.7447163.

[39] M. Iliadis, L. Spinoulas, A. K. Katsaggelos, Deep fully-connected networks for video compressive sensing, Digital Signal Processing 72 (2018) 9 – 18. doi:10.1016/j.dsp.2017.09.010.

[40] J. Zhang, B. Ghanem, ISTA-Net: interpretable optimization-inspired deep network for image compressive sensing, in: IEEE/CVF conference on computer vision and pattern recognition, 2018, pp. 1828–1837. doi:10.1109/CVPR.2018.00196.

[41] B. Sun, H. Feng, K. Chen, X. Zhu, A deep learning framework of quantized compressed sensing for wireless neural recording, IEEE Access 4 (2016) 5169–5178. doi:10.1109/ACCESS.2016.2604397.

[42] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM Journal on Imaging Sciences 2 (1) (2009) 183–202. doi:10.1137/080716542.

[43] P. E. McSharry, G. D. Clifford, L. Tarassenko, L. A. Smith, A dynamical model for generating synthetic electrocardiogram signals, IEEE Trans. on Biom. Eng. 50 (3) (2003) 289–294. doi:10.1109/TBME.2003.808805.

[44] S. Mallat, A wavelet tour of signal processing: the sparse way, Access Online via Elsevier, 2008.

[45] F. Maloberti, Data Converters, Springer Electronics & Electrical Engineering, 2007.

[46] A. Agnes, E. Bonizzoni, P. Malcovati, F. Maloberti, A 9.4-ENOB 1V 3.8μW 100kS/s SAR ADC with time-domain comparator, in: IEEE international solid-state circuits conference - digest of technical papers, 2008, pp. 246–610. doi:10.1109/ISSCC.2008.4523149.

[47] Y. Zhu, C.-H. Chan, U.-F. Chio, S.-W. Sin, S.-P. U, R. P. Martins, F. Maloberti, A 10-bit 100-MS/s Reference-Free SAR ADC in 90 nm CMOS, IEEE J. Solid-State Circuits 45 (6) (2010) 1111–1121. doi:10.1109/JSSC.2010.2048498.

[48] L. S. Y. Wong, S. Hossain, A. Walker, Leakage current cancellation technique for low power switched-capacitor circuits, in: 2001 Int. Symp. on Low Power Electronics and Design, 2001, pp. 310–315. doi:10.1145/383082.383175.