

RESEARCH

Open Access



An analog hardware solution for compressive sensing reconstruction using gradient-based method

Irena Orović^{1*} , Nedjeljko Lekić¹, Marko Beko² and Srdjan Stanković¹

Abstract

This work proposes an analog implementation of gradient-based algorithm for compressive sensing signal reconstruction. Compressive sensing has appeared as a promising technique for efficient acquisition and reconstruction of sparse signals in many real-world applications. It starts from the assumption that sparse signals can be exactly reconstructed using far less samples than in standard signal processing. In this paper, we consider the gradient-based algorithm as the optimal choice that provides lower complexity and competitive accuracy compared with existing methods. Since the efficient hardware implementations of reconstruction algorithms are still an emerging topic, this work is focused on the design of hardware that will provide fast parallel algorithm execution for real-time applications, overcoming the limitations imposed by the large number of nested iterations during the signal reconstruction. The proposed implementation is simple and fast, executing 400 iterations in 1 ms which is sufficient to obtain highly accurate reconstruction results.

Keywords: Analog hardware, Compressive sensing, Gradient reconstruction method, Signal reconstruction

1 Introduction

In conventional digital signal processing, the sampling frequency needs to be at least the double of the maximal signal frequency in order to be able to get exact reconstruction of signals. Alternatively, in compressive sensing (CS) framework, the signal can be exactly recovered even when the number of available samples is considerably below the conventional requirements [1–5]. Although certain conditions should be fulfilled for efficient CS scenario, it turns out that many signals in real applications are very conducive to CS. The first requirement is related to signal sparsity, which is generally achieved using certain transform domain representation [4–6]. The second one refers to incoherence between the

measurement process and the sparsity basis. In other words, a small set of linear measurements needs to be acquired in a random manner [4]. Under these conditions, the signal reconstruction is observed as a problem of solving undetermined system of linear equations. It is important to emphasize that the CS has been efficiently combined with different time-frequency approaches, either to define improved CS methods based on the time-frequency dictionaries or to define improved time-frequency representations based on the CS reconstruction principles [7]. A reliable instantaneous frequency estimation from a small set of available samples can be achieved by applying the signal reconstruction algorithms on the samples of the local autocorrelation function [8]. Therefore, the CS-based reconstruction methods bring several benefits to time-frequency analysis.

* Correspondence: irenao@ucg.ac.me

¹Faculty of Electrical Engineering, University of Montenegro, Podgorica, Montenegro

Full list of author information is available at the end of the article

Solving the problem of signal reconstruction from a small set of available samples can be a complex and demanding task from both software and hardware perspectives. The existing software solutions [9–14] are mainly based on different iterative convex optimization techniques or their greedy iterative counterparts. These software implementations are generally unsuitable for real-time applications, due to the fact that the algorithms require a large number of iterations to achieve convergence. Moreover, the intrinsic parallelism in the algorithm execution is difficult to achieve in a mono-processor computing system [15]. Therefore, in order to cope efficiently with signal reconstruction problem in compressive sensing, one needs a dedicated parallel hardware implementation of reconstruction algorithm. The hardware realization can be done using analog, digital, or combined analog and digital devices.

In this paper, we propose an analog hardware architecture of the gradient-based signal reconstruction algorithm [14, 16]. The chosen algorithm exhibits optimal performance for different types of signals. It belongs to a group of convex optimization methods prominent for its accuracy and is computationally less demanding than the other methods from this category. The proposed algorithm outperforms the standard ones used for the ℓ_1 -norm-based minimization problems in both the calculation time and accuracy [16]. In huge contrast to greedy methods, the gradient-based algorithm does not require the signal to be strictly sparse in a certain transform domain, rendering it suitable for real-world applications. Furthermore, it is well known the computational burden of digital hardware implementation can be intensive, leading to long execution time [17–19]. Hence, opting for analog implementation instead might bring considerable advantage in terms of complexity and consequently, to reduced execution time.

The paper is organized as follows. Theoretical background on the compressive sensing and gradient-based signal reconstruction algorithm is given in Section 2. Section 2 presents the proposed hardware solution, whose performance is discussed/assessed in Section 3. The concluding remarks are given in Section 4.

2 Compressive sensing—theory and reconstruction method

Compressive sensing framework emerged from the efforts to decrease the amount of data sensed in the modern applications [5], and consequently to decrease the needs for resources such as storage capacity, number of sensors, and consumption. The set

of measured data in traditional approach is usually referred to as full dataset, while the data in CS scenarios are called incomplete dataset. The data samples in the traditional sampling approach are equidistant and uniformly sampled, while in the CS, the data should be sampled randomly to achieve a high incoherence of the dataset. If the measurement process is modeled by a certain measurement matrix Φ , then the measured dataset y of length M can be described as [2]:

$$y = \Phi x \tag{1}$$

where x represents the full dataset of length N , and the measurement matrix is of size $M \times N$, with $M \ll N$. In practical scenarios, the CS dataset y and the measurement matrix Φ are known, and we desire to reconstruct the full dataset x of length N . However, the system of equations in (1) is underdetermined since $M \ll N$. Thus, in order to tackle the problem, one resorts to the assumption that the signal x is sparse when represented in certain transform basis:

$$X = Wx \tag{2}$$

where X is a transform domain representation of full signal dataset and W is the transform matrix, i.e.,

$$W = \begin{bmatrix} W_1(1) & W_2(1) & \dots & W_N(1) \\ W_1(2) & W_2(2) & \dots & W_N(2) \\ \dots & \dots & \dots & \dots \\ W_1(N) & W_2(N) & \dots & W_N(N) \end{bmatrix} \tag{3}$$

where W_i are the basis functions. Depending on the signal, the basic functions could belong to discrete Fourier transform (DFT), discrete Hermite transform (DHT), discrete wavelet transform (DWT), discrete cosine transform (DCT), etc. For the sake of simplicity, we can write W as the coefficients' matrix:

$$W = \begin{bmatrix} w_{11} & w_{21} & \dots & w_{N1} \\ w_{12} & w_{22} & \dots & w_{N2} \\ \dots & \dots & \dots & \dots \\ w_{1N} & w_{2N} & \dots & w_{NN} \end{bmatrix} \tag{4}$$

Sparsity means that the vector X has only K out of N nonzero (or significant) elements, while $K \ll N$ and $K < M$. The system of linear equations becomes:

$$y = \Phi W^{-1} X \tag{5}$$

The reconstructed signal X can be obtained as a sparsest solution of the following optimization problem [7]:

$$\text{minimize } \|X\|_1 \quad \text{subject to } y = \Phi W^{-1} X \quad (6)$$

where $\|X\|_1$ denotes l_1 -norm. Generally, the l_0 -norm is used to measure the signal sparsity, but in practical implementations, using the l_0 -norm can lead to an NP problem; hence, the l_1 -norm is used instead.

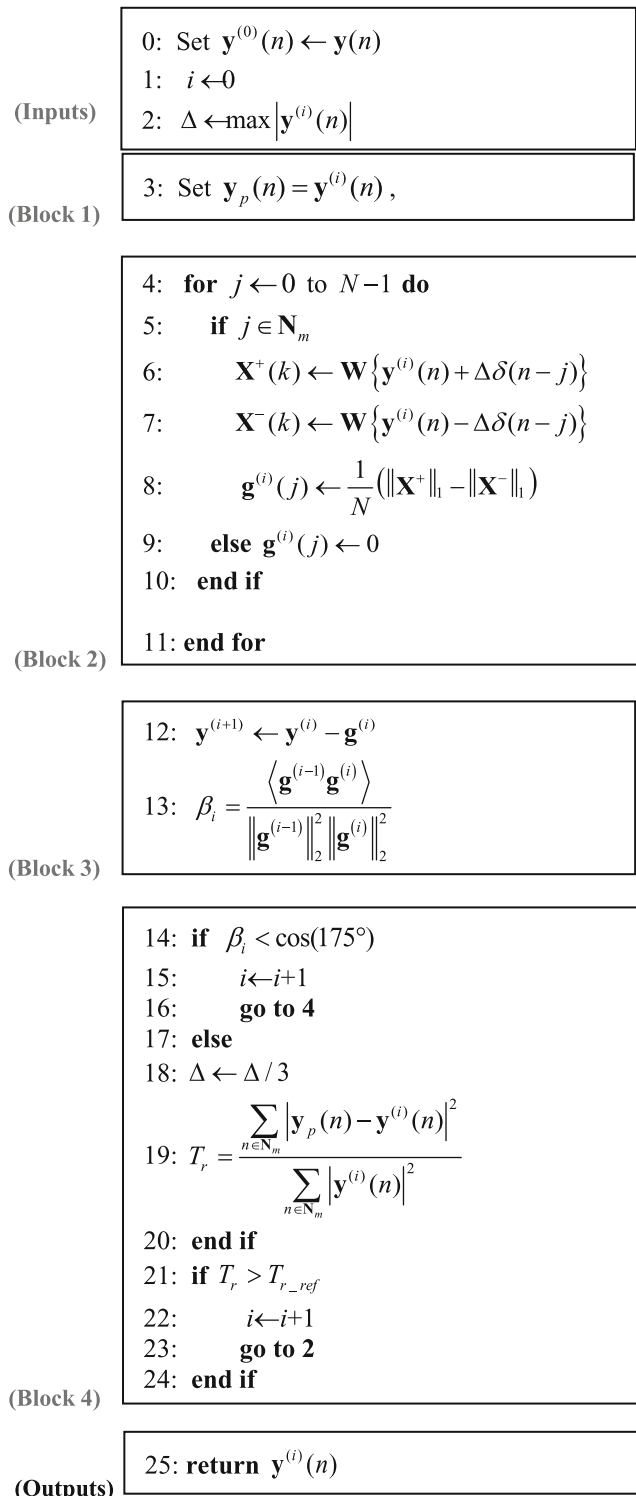
There are few types of algorithms that can be applied for solving this problem and the most commonly used are the convex optimization algorithms and the greedy algorithms [9, 10]. In general, the greedy solutions are simpler, but less reliable. Convex optimization algorithms are more reliable and accurate in general and guaranteed convergence, but their solutions comes at the cost of computational complexity and large number of nested iterations [4, 9]. In this paper, we consider the gradient-based convex optimization algorithm as a representative of the convex optimization group allowing simpler implementation compared with other common solutions.

2.1 Gradient-based signal reconstruction algorithm

Assume that the missing samples positions are defined by the set $N_m = \{n_1, n_2, \dots, n_{N-M}\}$. Now the measurement vector is extended to the length of full dataset such that we embed zeros on the positions of missing samples, i.e., $y(n) = 0$, for $n \in N_m$. The gradient-based approach starts from some initial values of unavailable samples (initial state) which are changed through the iterations in a way to constantly improve the concentration in sparsity domain. In general, it does not require the signal to be strictly sparse in certain transform domain, which is an advantage over other methods. In particular, the missing samples in the signal domain can be considered as zero values. In each iteration, the missing samples are changed for $+\Delta$ and for $-\Delta$. Then, for both changes, the concentration is measured as l_1 -norm of the transform domain vectors X^+ (for $+\Delta$ change) and X^- (for $-\Delta$ change), while the gradient is determined using their difference. Finally, the gradient is used to update the values of missing samples. Here, it is important to note that each sample is observed separately and a single iteration is terminated when all samples are processed.

In the sequel, we provide a modified version of the algorithm adapted for an efficient hardware implementation avoiding complex functions and calculations.

Modified Algorithm



The algorithm steps are combined within the blocks according to the parallelization principle. The steps within the same block can be implemented in parallel.

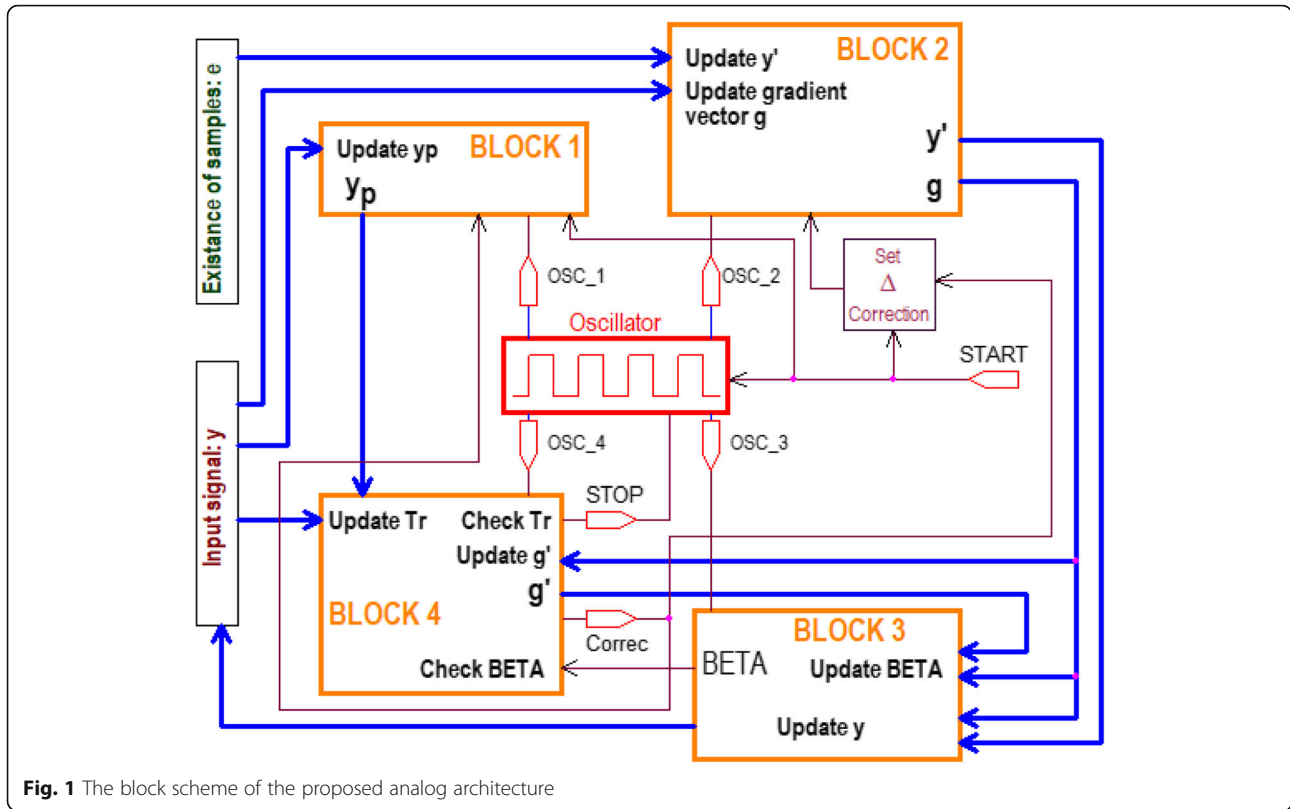


Fig. 1 The block scheme of the proposed analog architecture

The steps from the succeeding block are performed after the current block is processed. When the algorithm comes close to the minimum of the sparsity measure, the gradient changes direction for almost 180 degrees, meaning that the step Δ needs to be decreased (line 18). The precision of the result in this iterative algorithm is estimated based on the change of the result in the last iteration.

3 Analog hardware implementation of modified gradient-based algorithm

From the modified version of the algorithm, we can identify following steps that need to be implemented within analog hardware architecture:

- Set the signal samples at the input of analog architecture (line 0)
- Set the digital signal identifying the positions of available and missing samples at the input of the architecture: positions of available samples are marked by value "1" (high voltage) and missing samples are marked by value "0" (low voltage) values (line 0 in the modified algorithm).
- Set the value for the gradient step Δ as in line 2 in the modified algorithm
- Update the values of the gradient \mathbf{g} for input samples (lines 4 to 11 in the modified algorithm)
- Update the values of the missing samples and β (lines 12 and 13 in the modified algorithm)
- Check the condition for changing Δ (line 14 of the modified algorithm):

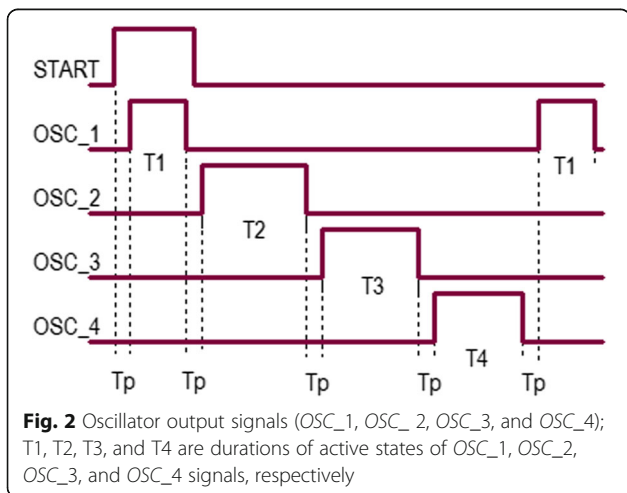
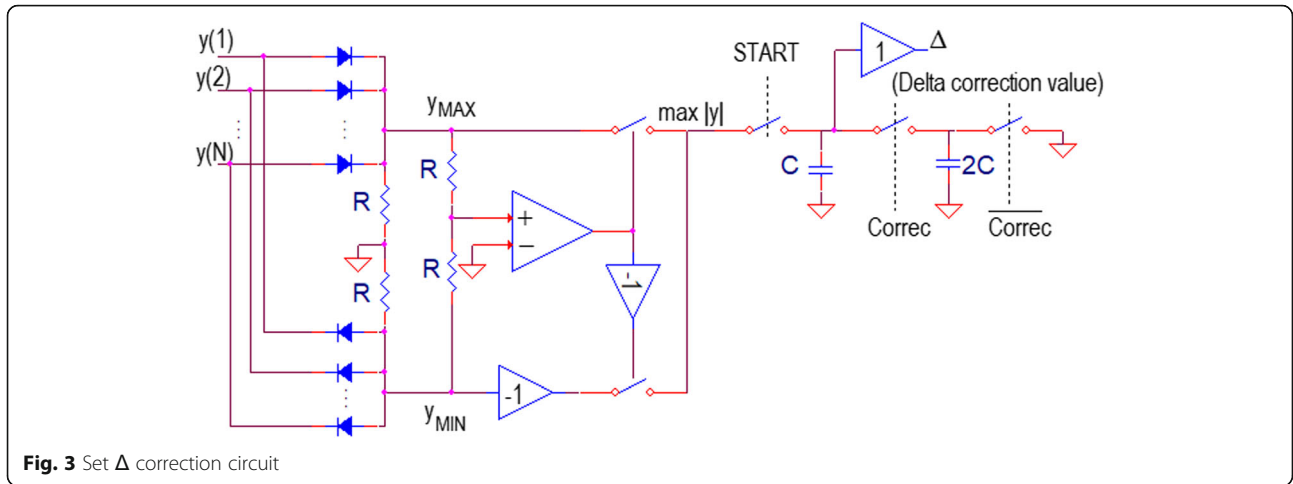


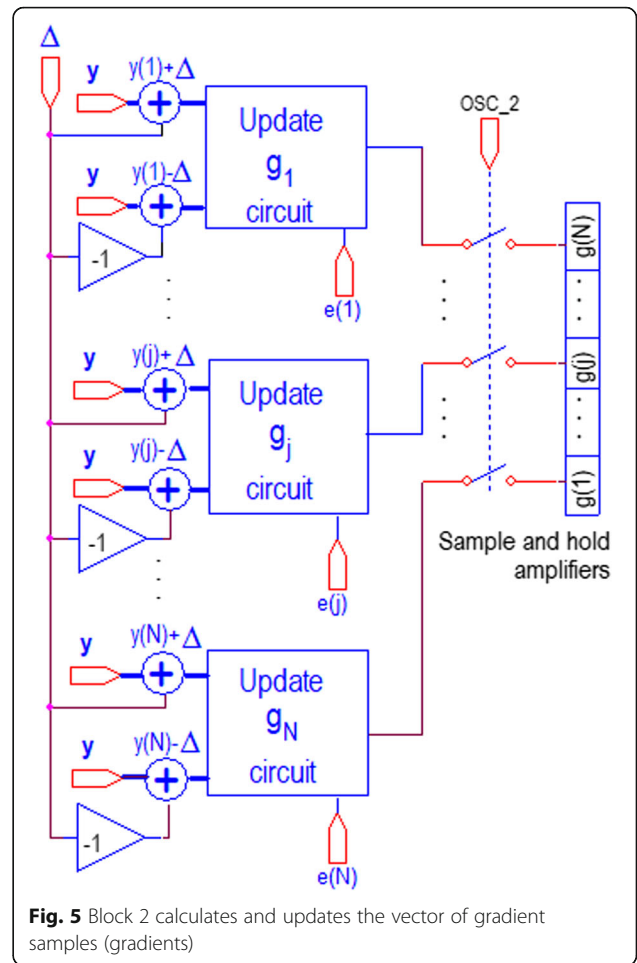
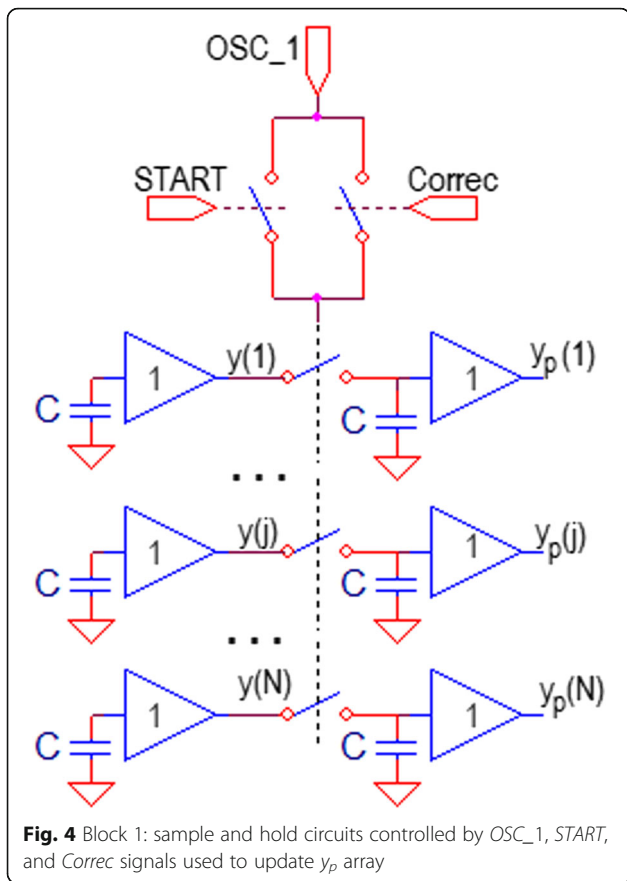
Fig. 2 Oscillator output signals (OSC_1 , OSC_2 , OSC_3 , and OSC_4); $T1$, $T2$, $T3$, and $T4$ are durations of active states of OSC_1 , OSC_2 , OSC_3 , and OSC_4 signals, respectively

- If the condition is not satisfied, repeat the steps d) and e) (lines 4 to 13 in the modified algorithm)



– If the condition is satisfied, Δ is decreased, for example, to $\Delta/3$ (line 18 in the modified algorithm). It is noteworthy that in order to achieve high precision, the step Δ should be decreased when approaching the stationary oscillations zone.

g) In parallel with changing the step Δ , check the stopping criterion. The stopping criterion T , is defined as the mean value of changes applied on the missing samples using the previous value of gradient step Δ (line 19: in the modified algorithm). When the applied changes are less than the



predefined minimum, the updated value of Δ has no significant influence on the signal quality and thus, the procedure should be stopped. In such a case, the current values of input signal represent the reconstruction result. If the stopping criterion is not met, the procedure is repeated with new updated value Δ .

The block scheme for the proposed analog architecture is shown in Fig. 1.

The architecture is composed of the following segments:

- Input array holding the analog samples y ,
- Input array e holding the indicators of available and missing samples positions,
- Set Δ correction circuit,
- Four architecture blocks: *Block 1*, *Block 2*, *Block 3*, and *Block 4*.

The proposed architecture blocks (*Block 1*, *Block 2*, *Block 3*, and *Block 4*) have been driven by the output signals from the oscillator, namely *OSC_1*, *OSC_2*, *OSC_3*, and *OSC_4*, respectively. The sequence of active states of the oscillator's signal controls the activation order of the four architecture blocks. Before the *START* signal, all outputs are set to low voltage level. After activating *START* signal, the oscillator's output signals are shown in Fig. 2.

Prior to activating the *START* signal, the values of input signals y are loaded to the sample and hold amplifiers, and e values are set. During the *START* impulse, the initial value of Δ is set to the output of *set Δ correction* circuit, whose electrical diagram is shown in Fig. 3. Let us briefly discuss the *set Δ correction* circuit. By activating *START* signal, the capacitor C is loaded to the approximately maximal absolute value of the input signal samples. It is the initial value of the Δ correction.

Further update of Δ is controlled by the *Correc* signal: when *Correc* is high, Δ is changed to $\Delta/3$ (as in line 18 in the modified algorithm).

The architecture *Block 1* is used to temporarily store values of y within the array y_p (for the current value of Δ), Fig. 4. *Block 1* is active when *OSC_1* (T1 impulse) is high and either *START* or *Correc* signal is active. At the beginning, during active state of *START* signal and first T1 impulse, the initial value of array y is loaded into the array y_p . Further update of y_p is controlled by active state of *OSC_1* signal and active state of the *Correc* signal. It is important to note that y_p values are updated each time when a new correction value Δ is set to $\Delta/3$. Therefore, y_p represents the version of reconstructed signal at the moment of applying a new Δ value. It is used later to calculate T_r (line 19 in the modified algorithm) as the mean square error between y signal before and y signal after applying current Δ correction value.

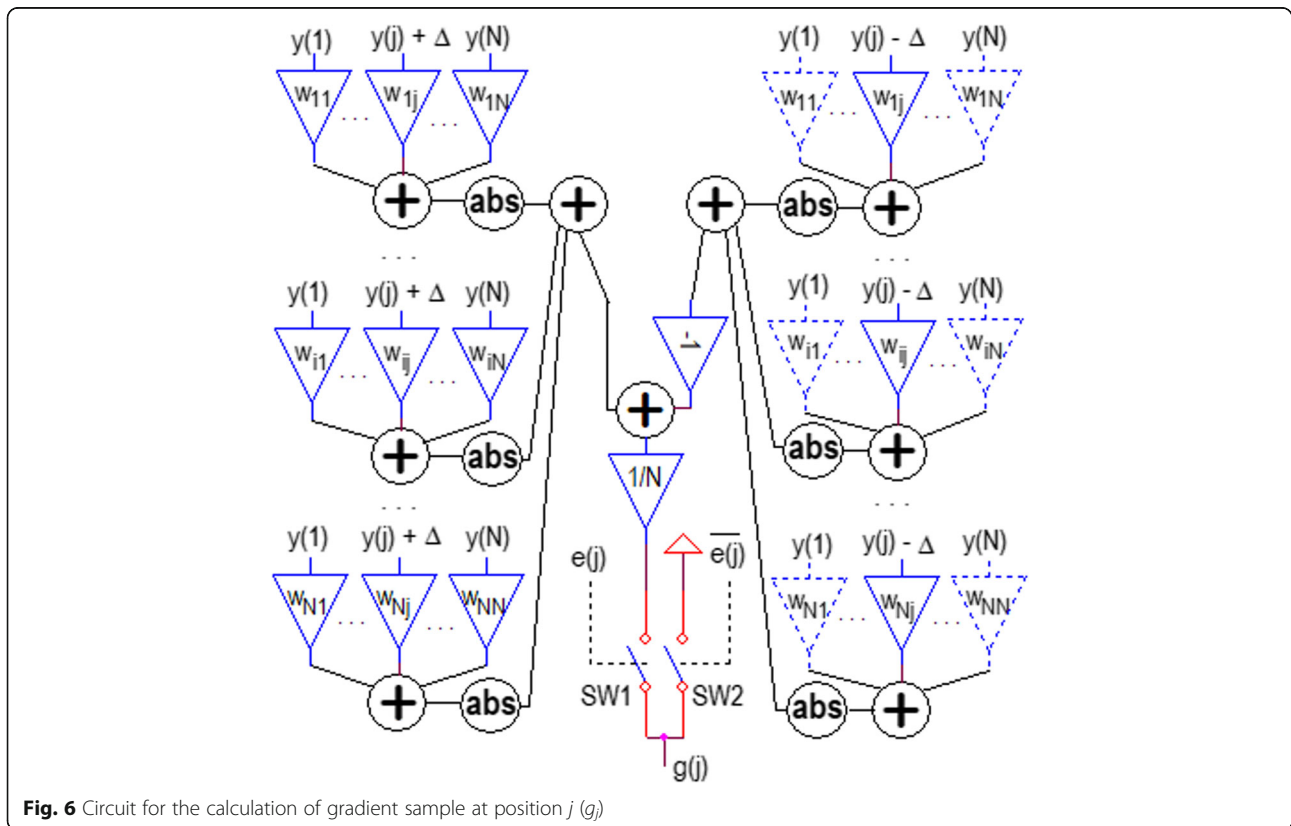
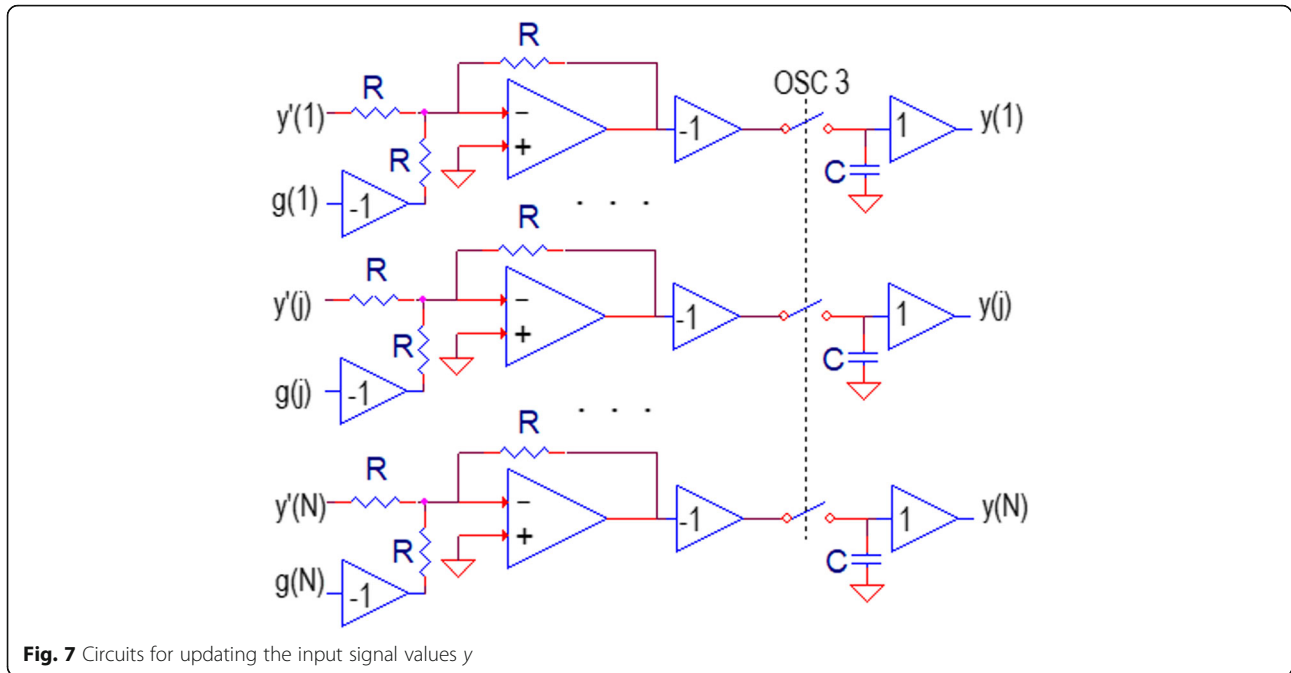
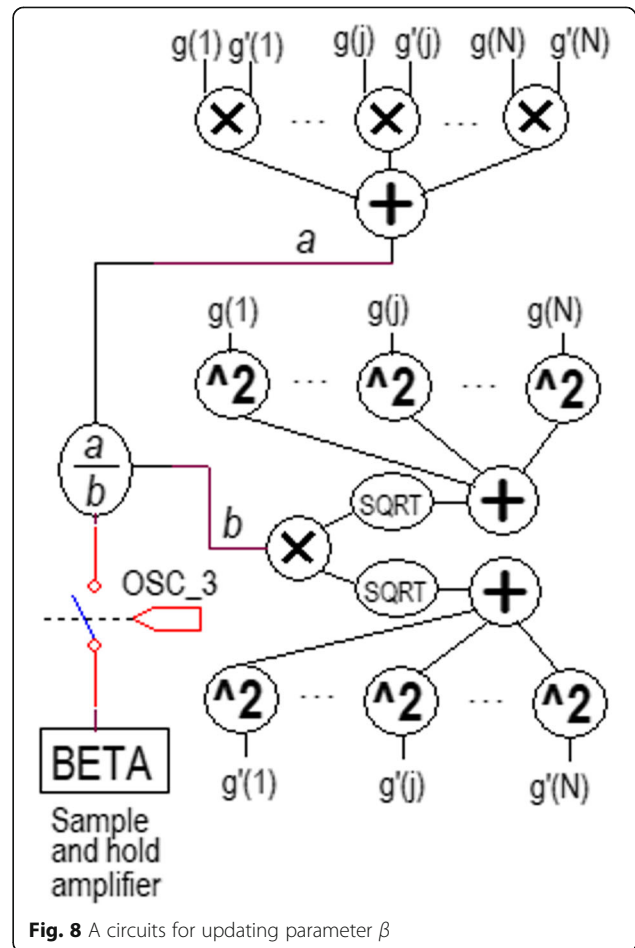


Fig. 6 Circuit for the calculation of gradient sample at position j (g_j)



The architecture *Block 2* is active when *OSC_2* (T2 impulse) is high and it is used to update the values of the gradients g (for each missing sample position), as shown in Fig. 5. Current values of y are placed in y' (for the i th iteration $y' = y^{(i)}$ in the algorithm). The values of signal samples y are loaded into the sample and hold amplifiers y' . Further, the architecture *Block 2* allows parallel computation of all gradient samples in g . A circuit used to calculate a single gradient sample is shown in Fig. 6. It consists of amplifiers representing the transformation coefficients w_{ij} belonging to the transformation matrix W (defined in (5)), adders and absolute value circuits. The calculation of a gradient sample g_i requires $N(N + 1) + 1$ amplifiers, $2(N + 1)$ adders with N inputs, $2N$ absolute value circuits, 1 adder with two inputs, 1 inverter circuit, and 2 switches. Note that the amplifiers drawn by the dashed line (right side of Fig. 6) are not supposed to be implemented in hardware. Namely, these amplifiers are shown in Fig. 6 only to provide a clearer circuit illustration, but the necessary values are already available at the output of the appropriate amplifiers on the left side of Fig. 6. The switches in Fig. 6 are controlled by the signal e containing the indicators of available and missing samples. The switch *SW1* needs to be closed on the positions of missing samples and opened on the positions of available samples (the gradients are calculated only for the missing samples positions), while the situation with *SW2* is opposite. Hence, at the position of available sample, the signal $e(i)$ keeps the switch *SW1* open, while the signal $\overline{e(i)}$ closes the *SW2* switch. At the position of missing sample, the signal $e(i)$ keeps the switch *SW1* closed, while the signal $\overline{e(i)}$ opens the switch *SW2*.



The architecture *Block 3* is active when *OSC_3* signal (*T3* impulse) is high and it is used to obtain updated signal values y using gradient g and current signal values y' from *Block 2* ($y^{(i+1)} = y^{(i)} - g^{(i)}$ in the algorithm). This part of *Block 3* is shown in Fig. 7. Another part of the architecture *Block 3* is responsible for updating value of β (Fig. 8). The value of β is fed to the input of the circuit that validates the condition for changing the step Δ .

The circuit in Fig. 8 consists of analog voltage adders, multipliers, squaring and rooting circuits, division (multiplication) circuits, and sample and hold amplifiers. It can be seen that the calculation of β requires 3 adders with N inputs, $N + 1$ multiplier, $2N$ squaring circuits, two square-rooting circuit, and division circuit. Note that the array denoted by g' holds the gradient values from the previous iteration. Update of values in g' array is done in *Block 4*.

The architecture *Block 4* is active when *OSC_4* signal (*T4* impulse) is high and it is used to load current values from g into g' . The circuits in *Block 4* are further used to check the parameter β and activate *Correc* signal (Fig. 9a), update the value of T_r (Fig. 9b), and check whether the stopping criterion is met (Fig. 9c). The circuit in Fig. 9b consists of the adders ($N + 2$), multipliers, N inverters, $2N$ squaring analog voltage

circuits, and 1 division circuit. The circuit in Fig. 9 a compares β and β_{REF} , and if $\beta < \beta_{REF}$, the signal *Correc* is activated as a trigger to update the value of Δ . The active state of signal *Correc* allows us to check the stopping criterion (Fig. 9c). If the stopping criterion is met, the *STOP* signal is activated. This causes deactivation of all oscillator outputs and consequently all switches in the hardware become open. In the case where the stopping criterion is not met, *RcCc* time constant is used to keep the high voltage of *Correc* signal for the next *T1* period. It is used to enable update of sample and hold amplifiers denoted by y_p .

The required components per block are shown in Table 1 (only the main and most represented components are listed, such as amplifiers, adders, and multipliers).

4 Results and discussion

The simulation of the proposed hardware architecture is done using the PSpice (OrCAD 16.6). For the realization of multipliers, division circuit, squaring, and square-rooting circuit the analog device AD734 is used. The circuit represents a precise four-quadrant high-speed analog multiplier that is able to perform all the abovementioned functions. For the

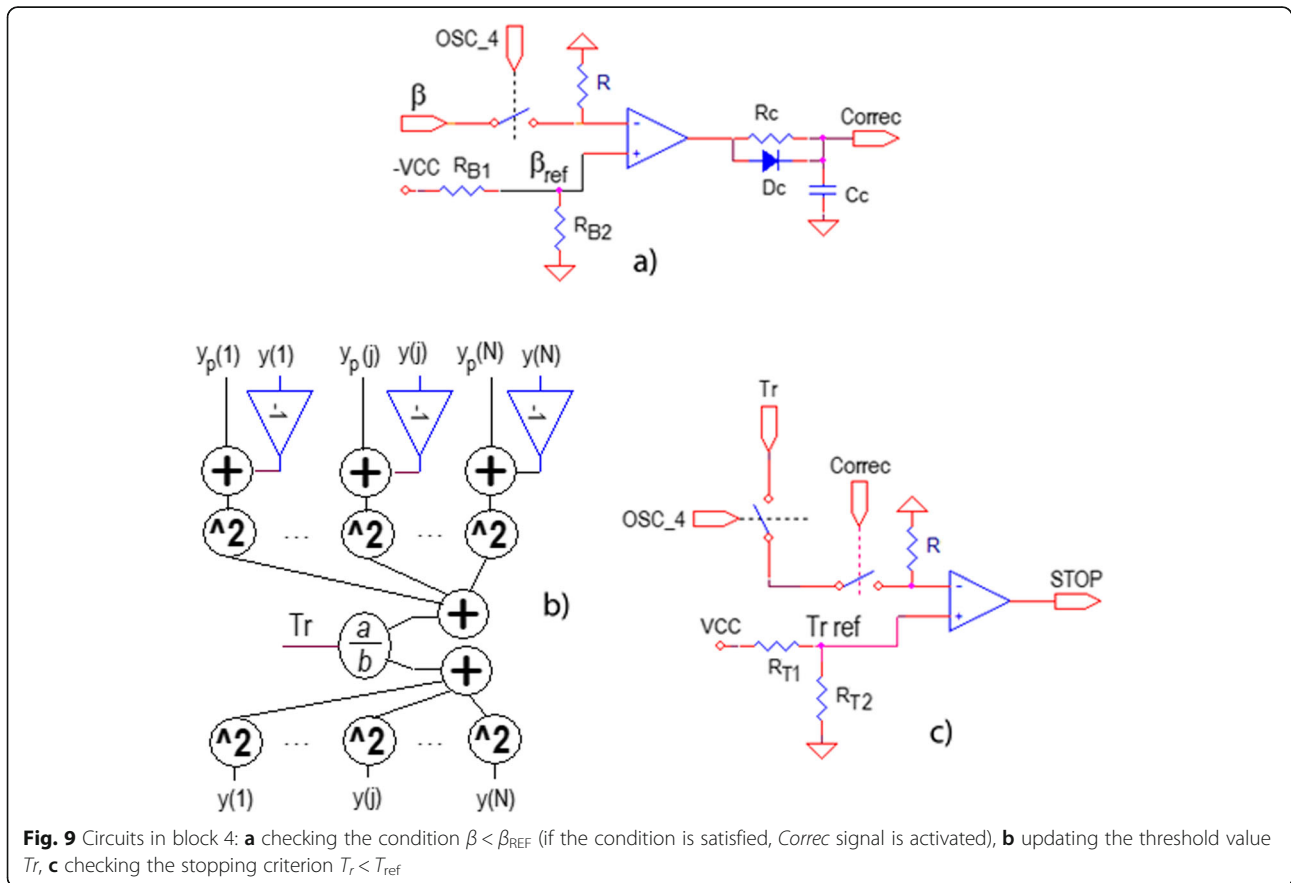


Fig. 9 Circuits in block 4: **a** checking the condition $\beta < \beta_{REF}$ (if the condition is satisfied, *Correc* signal is activated), **b** updating the threshold value T_r , **c** checking the stopping criterion $T_r < T_{ref}$

Table 1 The number of the most represented circuits required

	Input signal circuits	Block 1	Block 2	Block 3	Block 4	Total
Amplifiers			$N^3 + N^2 + 3N$	$2N$	N	$N^3 + N^2 + 6N$
Adders			$2N^2 + 4N$	$N + 3$	$N + 2$	$2N^2 + 6N + 5$
Absolute value circuits			$2N^2$			$2N^2$
Sample and hold amplifiers	N	N	$2N$	1	N	$5N + 1$
Switches		$N + 2$	$4N$	$N + 1$	$N + 3$	$7N + 6$
Multipliers/divisions				$3N + 4$	$2N + 1$	$5N + 5$

realization of amplifiers (inverters), adders and absolute value circuits, we employed the ultra-high-speed operational amplifier LT1191. For the analog voltage comparison, we employed the circuit LT1394 (comparator) [20] (with settling time 7 ns). The circuit provides the complementary outputs and allows the zero-crossing detection. The sample and hold circuits AD783 are used for storing the voltage values. Furthermore, in the simulation, we used the analog switches MAX4645 [21] and diodes 1N4448 [22].

The simulation has shown that the required duration for the oscillator outputs are $T1 \approx 320$ ns, $T2 \approx 740$ ns, $T3 \approx 650$ ns, and $T4 \approx 610$ ns. The time between the impulses (Tp) is required to turn off the analog switches used in the proposed hardware. The turn off time for switches MAX4645 is up to $Tp = 40$ ns. Hence, the minimal duration for one algorithm iteration is $T1 + T2 + T3 + T4 + 4Tp = 2.48 \mu s \approx 2.5 \mu s$. The total calculation time for the reconstruction of an input signal depends on the number of iterations. The proposed solution allows up to 400 iterations within 1 ms, and it is interesting to mention

that most of the signals in real applications can be completely accurately reconstructed within a few hundreds of iterations [16].

Another advantage of the proposed hardware solution is the robustness on hardware imperfection. The used components generate a very small error. The main source of errors is the circuit AD734. However, it contributes in error with typical 0.1% of the full scale or 0.25 in full temperature range from -40 to $+85$ °C. Our simulations have shown that the maximal introduced error can cause only a few additional iterations to achieve a high precision reconstruction performance, which is negligible in terms of time and processing costs.

4.1 Example

In order to provide an illustration of signal reconstruction results, let us observe the experimental synthetic signal that is sparse in the Hermite transform domain. The original signal contains 64 samples, but only 26 samples are available on the random positions within the signal, while 38 samples are missing and should be

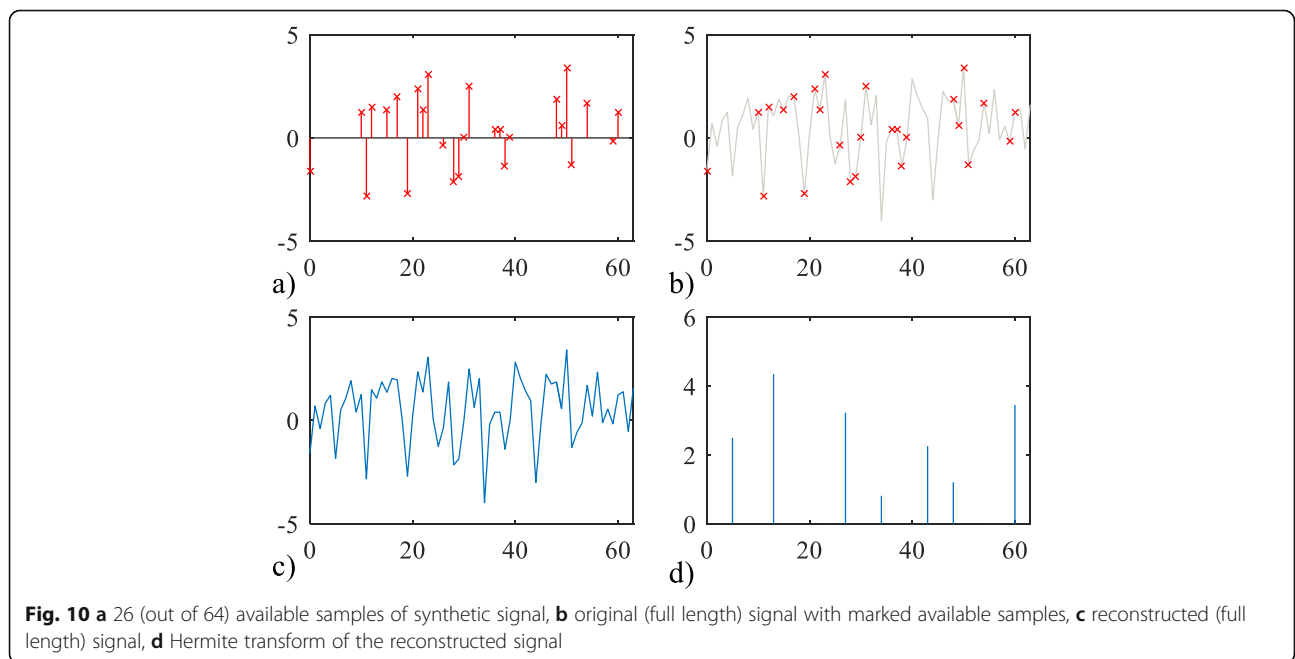


Fig. 10 **a** 26 (out of 64) available samples of synthetic signal, **b** original (full length) signal with marked available samples, **c** reconstructed (full length) signal, **d** Hermite transform of the reconstructed signal

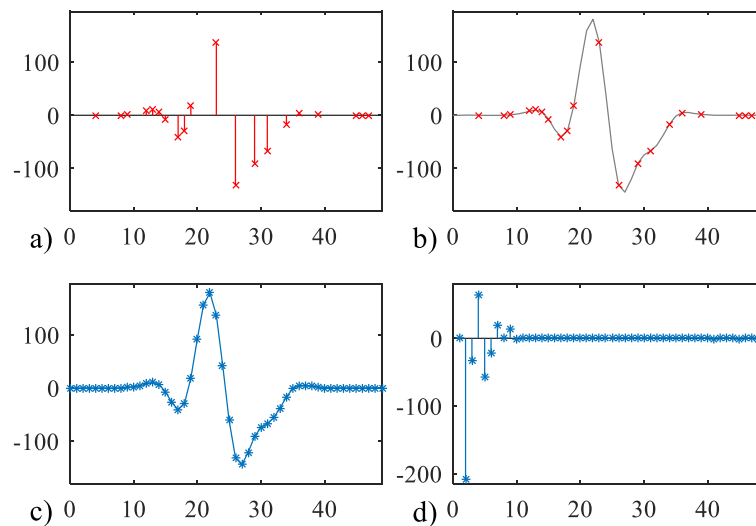


Fig. 11 **a** 20 available samples (out of 50) of QRS signal, **b** original QRS signal (full-length signal consisting of 50 samples) with available samples marked in red, **c** reconstructed (full length) QRS signal, **d** Hermite transform of the reconstructed signal

reconstructed. The available samples are shown in Fig. 10a. The original signal with missing samples marked in red color is shown in Fig. 10b. The reconstructed signal is shown in Fig. 10c. The sparse Hermite transform after the reconstruction is shown in Fig. 10d. The achieved mean square error is of order 10^{-3} which can be considered negligible in the case of observed signal. Thus, the reconstruction result is highly accurate.

In real-world applications, the same concept can be applied for the reconstruction of QRS complexes (as shown in Fig. 11) in ECG signals, or ultra-wide band (UWB) signals in communications, that are also sparse in the Hermite transform basis. In the case of other communications signals (e.g., FHDSS), the Fourier transform basis would be more appropriate for achieving sparse signal representation. Finally, without loss of generality, this approach can be applied in the same way by using any other transform basis as long as it allows sparse representation of the specific observed signal. Therefore, the proposed hardware implementation is very amenable to various practical applications, for instance in communications, radars and remote sensing, biomedicine, etc.

5 Conclusion

In this work, we presented an analog hardware architecture for gradient-based algorithm for sparse signal reconstruction. The algorithm is suitable for different types of signals and therefore is suitable for real-time implementation which opens a wide range of applications, including time-frequency signal analysis and instantaneous frequency estimation. The proposed

analog design allows fast processing and low-complexity realization, executing significant number of algorithm iterations in short-time intervals (at the order of 1 ms), which is highly satisfactory for real-time applications.

Abbreviations

CS: Compressive sensing; DFT: Discrete Fourier transform; DHT: Discrete Hermite transform; DWT: Discrete wavelet transform; DCT: Discrete cosine transform

Acknowledgements

Not applicable.

Authors' contributions

All authors made contributions in the discussions, analyses, and implementation of the proposed hardware solution. IO and NL were actively involved in writing the manuscript. All authors read and approved the final manuscript.

Funding

There are no financial sources of funding for the research to be reported.

Availability of data and materials

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Consent for publication

The manuscript does not contain any individual person's data in any form (including individual details, images, or videos) and therefore the consent to publish is not applicable to this article.

Competing interests

The authors declare that they have no competing interests.

Author details

¹Faculty of Electrical Engineering, University of Montenegro, Podgorica, Montenegro. ²COPELABS, Universidade Lusófona de Humanidades e Tecnologias, Lisbon, Portugal.

Received: 20 August 2019 Accepted: 22 November 2019

Published online: 17 December 2019

References

1. D. Donoho, Compressed sensing. *IEEE Trans. Inf. Theory* **52**(4), 1289–1306 (2006)
2. R. Baraniuk, Compressive sensing. *IEEE Sig. Proc. Magazine*. **24**(4), 118–121 (2007)
3. M. Elad, Sparse and redundant representations: from theory to applications in signal and image processing. Springer (2010)
4. S. Foucart, H. Rauhut, *A mathematical introduction to compressive sensing* (Springer, New York, 2013)
5. S. Stankovic, I. Orovic, E. Sejdic, *Multimedia signals and systems: basic and advance algorithms for signal processing*, 2nd edn. (Springer-Verlag, New York, 2015)
6. Y.C. Eldar, *Sampling theory: beyond bandlimited systems*, (Cambridge University Press) (2015)
7. E. Sejdić, I. Orović, S. Stanković, Compressive sensing meets time-frequency: an overview of recent advances in time-frequency processing of sparse signals. *Digital Signal Processing* **77**, 22–35 (2018)
8. I. Orović, S. Stanković, T. Thayaparan, Time-frequency based instantaneous frequency estimation of sparse signals from an incomplete set of samples. *IET Signal Processing* **8**(3), 239–245 (2014)
9. E. Candes, T. Tao, Decoding by linear programming. *IEEE Trans. Inf. Theory* **51**(12), 4203–4215 (2005)
10. J. Tropp, A. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans. Inf. Theory* **53**(12), 4655–4666 (2007)
11. A. Draganić, I. Orović, and S. Stanković, "On some common compressive sensing recovery algorithms and applications—review paper," *Facta Universitatis, Series: Electronics and Energetics*, Vol 30, No 4 (2017), pp. 477-510, DOI Number 10.2298/FUEE1704477D, December 2017
12. J. Music, I. Orovic, T. Marasovic, V. Papić, S. Stankovic, Gradient compressive sensing for image data reduction in UAV based search and rescue in the wild, *Mathematical Problems in Engineering*, 2016 (ID 6827414), 14 pages
13. N. Lekić, A. Draganić, I. Orović, S. Stanković, and V. Papić, "Iris print extracting from reduced and scrambled set of pixels," *Second International Balkan Conference on Communications and Networking BalkanCom 2018*, Podgorica, Montenegro, June 6-8, 2018
14. M. Brajovic, I. Orovic, M. Dakovic, S. Stankovic, Gradient-based signal reconstruction algorithm in the Hermite transform domain. *Electron. Lett* **52**(1), 41–43 (2016)
15. N. Nedjah, R. M. da Silva, L. de Macedo Mourelle, Analog hardware implementation of artificial neural networks, *J. of Circuit Syst. and Comp.*, **20**(3), 349-373 (2011).
16. L. Stankovic, M. Dakovic, On a gradient-based algorithm for sparse signal reconstruction in the signal/measurements domain, *Mathematical Problems in Engineering*, 2016 (ID 6212674), 11 pages.
17. S. Vujovic, M. Dakovic, I. Orovic, S. Stankovic, An architecture for hardware realization of compressive sensing gradient algorithm, in *Proc. 4th Mediterranean Conference on Embedded Computing, MECO 2015*, Budva, Montenegro, 2015.
18. I. Orović, A. Draganić, N. Lekić, and S. Stanković, "A system for compressive sensing signal reconstruction," *17th IEEE International Conference on Smart Technologies, IEEE EUROCON 2017*
19. N. Zaric, N. Lekic, S. Stankovic, "An implementation of the L-estimate distributions for analysis of signals in heavy-tailed noise," *IEEE Transactions on Circuits and Systems II*, Vol 58. No.7, pp. 427-432, 2011
20. LINEAR TECHNOLOGY CORPORATION, 7 ns, Low Power, Comparator – LT1394 (datasheet). 16 pages, [Online]. Available at: <http://cds.linear.com/docs/en/datasheet/1394f.pdf>. Accessed Apr 2019.
21. MAXIM, Fast, Low-Voltage, 2.5 Ω, SPST, CMOS Analog Switches – MAX4645 (datasheet). 11 pages, [Online]. Available at: <https://datasheets.maximintegrated.com/en/ds/MAX4645-MAX4646.pdf>. Accessed Apr 2019.
22. VISHAY SEMICONDUCTORS, Small Signal Fast Switching Diodes – 1 N4448 (datasheet). 4 pages, [Online]. Available at: <http://www.vishay.com/docs/81858/1n4448.pdf>. Accessed Apr 2019.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com