Implementation of an Electrical Signal Compression System Using Sparse Representation

Mateus Mostaro de Oliveira
Leandro Rodrigues Manso Silva
Carlos Augusto Duque
Luciano Manhes de Andrade Filho
Federal University of Juiz de Fora
Minas Gerais - Brazil

Paulo Fernando Ribeiro
Federal University of Itajuba
Minas Gerais - Brazil

Abstract—The storage of voltage and current signals over a period of time generates a large memory expense. Therefore, signal compression techniques became important in this context. This paper presents an implementation in Field Gate Programmable Array (FPGA) of an algorithm of sparse representation using redundant dictionaries, applied to the compression of electrical signals, from power systems. The representation will be based on a dictionary constituted by elements of the Fourier and Wavelet basis, that are capable to represent the stationary and transient components of the electrical signals. The results will be analyzed due to two parameters: the quality of the compressed signal, in terms of its correlation coefficient related to the original signal; and the number of elements in the representation, that is related with the compression ratio. The feasibility of the implementation in real time will be evaluated in terms of the consumed FPGA resources and the necessary frequency of operation.

Index Terms—Sparse Signal Representation, Redundant Dictionary, Power System Signal Compression, Matching Pursuit, FPGA.

I. INTRODUCTION

The analysis power system signal became increasingly important, mainly in the smart grids scenario, where there is a large insertion of distributed generation and non-linear loads, that causes some deformities in the voltage and current signals. Some of these deformities are already known and there are specific equipments designed to deal with them. On the other side, some new disturbances not know yet, may occur and need to be stored to a posterior analysis [1].

The continuous storage of raw data from the electrical power system is not a simple task due to the large amount of data to be stored and then transferred to a processing center. In addition, few commercial equipment are currently available for recording the oscillography data over a large period of time and at a relatively high sampling rate. In general the equipment available for this purpose is oriented to specific applications and is used only for the acquisition of a short period of the signal during a fault or a specific disturbance [2].

The power system system signals, as well speech and video signals, have a significant amount of redundancy and useless information. Then, if a proper signal processing technique is used, it is possible to obtain a sparse approximation of the signal, reducing the amount of data to be stored. The sparse representation must accurately reproduce all the important information in the signal with a smaller dimension than the original one [3].

Sparse representation over redundant dictionaries methods are used in various areas, such as: dictionary learning, image processing, image classification, visual tracking, etc. [4]. Methods based on this technique are proposed in [5], [6] where in the former the signal is represented using sinusoidal waves together with impulses; and, in the later, the elements are damped sinusoids based on gabor elements [7].

The paper is organized as follows. Section II presents the a review about the Sparse Representation problem. Section III presents the Matching Pursuit algorithm. Section IV presents the implementation of Matching Pursuit algorithm in FPGA. Section V presents the obtained results of the proposed system. And, finally, Section VI presents the conclusions.

II. SPARCE REPRESENTATION OF SIGNALS

Before introducing the sparse representation, it is important to understand the concept of atomic decomposition of signals, which consists of using predefined waveforms to express signals. These waveforms are called atoms and are elements of a set called dictionary. This decomposition is considered adaptive, since the elements of the dictionary are chosen according to the signal that will be represented. Mathematically, \( A \in \mathbb{R}^{N \times M} \) is the matrix called dictionary, \( x \in \mathbb{R}^{M} \) is the vector of coefficients, and \( b \in \mathbb{R}^{N} \) is the signal that will be represented by a linear combination of \( M \) columns (atoms) of the Matrix \( A \), as described in:

\[
Ax = b
\]  

(1)

Atomic signal decomposition techniques are used in various areas, such as: denoising [8]; harmonic analysis [9], [10]; parameter extraction [11]; time-frequency decomposition [12], [13];

In problems of sparse representation, the matrix \( A \) is considered a redundant dictionary, since it has more elements, or functions, than those necessary to establish a base.

978-1-5386-0517-2/18/$31.00 c 2018 IEEE
Thus, matrix $A$ has more columns than rows ($M > N$), and therefore the system shown in (1) has several solutions. The solution using fewest elements of the dictionary is considered the sparsest representation of the signal. Therefore, techniques of atomic decomposition can be used in signal compression, in which the problem is precisely to find a solution for equation (1) that is the sparsest one.

In this way, the techniques that are used for sparse representation faces two main problems: (i) given a dictionary matrix, how to find the solution of (1) with the least number of elements; and (ii) how to construct the dictionary matrix to obtain a sparse representation of the signal.

In the case of the first problem, the system in (1) is an indeterminate system of equations and a specific solution of this system is needed, therefore a criterion must be established that can express the desired characteristics of that solution. A way to introduce this criterion is through a cost function $J(.)$. Then a generalized optimization problem can be written as:

$$ \begin{align*}
(P_f) : \min_{x} & \ J(x) \quad \text{subject to} \quad b = Ax
\end{align*} $$

It is known that $l_p$ norm, for $p < 1$, generates sparse solutions. But these are not said to be formal norms, since some properties may not be satisfied. Among the norms $l_p$, for $p < 1$, the norm $l_0$ is the one that best describes sparsity, since its value is the number of nonzero elements contained in the vector. The problem $(P_f)$ now turns into $(P_0)$ and is shown in the following equation:

$$ \begin{align*}
(P_0) : \min_{x} & \ ||x||_0 \quad \text{subject to} \quad b = Ax
\end{align*} $$

The solution to the problem shown in (3) presents some challenges due to the discrete and discontinuous nature of the $l_0$ norm. The solution of $(P_0)$ is a classical problem of combinatorial search, in which one must generate all possible sparse subsystems $b = A_Sx_S$, where $A_S$ is a matrix containing only $|S|$ columns of matrix $A$ with indexes contained in $S$ and test whether each of these subsystems can be solved. The complexity of this solution increases exponentially in $m$ (number of columns of matrix $A$), so $(P_0)$ is classified as an NP-Hard problem [14].

Since the direct solution of the problem $(P_0)$ is impracticable in computational terms, it is necessary to seek for other possible reliable solutions. It is observed that the task of finding the variable $x$ can be divided in two steps: find its support, that is, the indexes of the nonzero elements and determine the value of these elements. The category of greedy algorithms are based on this strategy and will be described in the next section.

The choice of the dictionary has fundamental importance, since it will impact directly on the sparsity of the solution. The use of predefined dictionaries usually leads to faster transformations, but they are limited in representing, in a sparse way, only the class of signals for which it was designed. One way to overcome this limitation is to use adaptive dictionaries. For this, a set of training data is needed that contains signals similar to the signals that one wishes to represent, and then, the dictionary elements will be constructed according to this set.

Choosing a learning dictionary entails a high computational cost. There are several methods in the literature that perform this dictionary training. These methods use a set of training data $Y = (y_i)_{i=1}^N$ consisting of examples of the signals to be represented, and assume that there is a dictionary $A$ matrix that is capable of providing a sparse representation for each element $y_i$ of this set. The task is then to find out which is this matrix $A$ [15].

Since the power systems signal are formed by stationary (harmonics) and transient components, in this work, a predefined dictionary will be used. The chosen dictionary is constituted by elements of Fourier base, together with elements of the Wavelet base, that are suitable for representing sparsely the stationary and the transient components, respectively.

### III. THE MATCHING PURSUIT ALGORITHM

A way to find the solution of the $(P_0)$ problem described in (3) is through exhaustive searching. However, this solution proves impractical in almost all situations. For example, assuming matrix $A$ has $n$ rows and $m$ columns, and that vector $b$ is a linear combination of at most $k_0$ elements of this matrix, it is necessary to enumerate all possible combinations of $k_0$ elements of $A$ and to test them to obtain the sparsest one.

Use greedy algorithms is a way to obtain the sparsest solution in a manner that is computationally feasible. They generally focus in solving the problem the $(P_{0,\epsilon})$ problem, described in (4), that admits solutions that approximates the signal.

$$ \begin{align*}
(P_{0,\epsilon}) : \min_{x} & \ ||x||_0 \quad \text{subject to} \quad ||b - Ax||_2 < \epsilon
\end{align*} $$

The Matching Pursuit (MP), proposed in [16], is a greedy algorithm in which each iteration selects the dictionary element that has the greatest orthogonal projection in the residual signal (assuming the dictionary elements are normalized). The selected element is added to the solution support and its coefficient is its own projection value.

The algorithm is presented in Fig. 1 and it works as follows: in the sweep stage, the internal products between the residue $r^{k-1}$ and each element (column) $a_j$ in matrix $A$ are calculated; in the update support stage, the index $j_0$ of the element $a_j$ that presented the highest internal product value is added to the support; the updating of the provisional solution, the coefficients that were already part of the $S^{k-1}$ support are kept unchanged and the new coefficient, referring to $j_0$ is chosen as being $z$, which is the value of the internal product between element $j_0$ and matrix $A$ and the residue $r^{k-1}$; finally the new residue $r^k$ is calculated and the stopping criterion is evaluated.

Fig. 1. MP - Matching Pursuit [16].
IV. IMPLEMENTATION OF MP ALGORITHM IN FPGA

The implementation of the MP algorithm in Field Programmable Gate Array (FPGA), was done dividing the algorithm in three main blocks of processing as shown in Fig. 2. Each block contains an embedded processor. Two blocks named FFT and Wavelet, respectively, runs in parallel and are responsible for performing the inner products needed by the sweep stage: one is responsible to calculate the inner products related to the Fourier base and the other the inner products related to the Wavelet base. The third block named Compression, takes the results of all calculated inner products and calculates the provisional solution and updates the residual, that retrofits the FFT and WAVELET blocks until the stop criterion is reached.

Fig. 2. Block diagram of the proposed compression system.

The implementation of the three parts of the algorithm were done using the C language and the embedded processors that were built using Verilog language. Verilog was also used to implement the interface and control blocks such as FIFO (First In First Out) memories, Multiplexers (MUX) and Demultiplexers (Demux).

The processors implemented in the FPGA, are based on the Reduced Instruction Set Computer Architecture (RISC) and have separate memories for data and instructions (Harvard Architecture). Only internal FPGA resources are used, such as memories and Digital Signal Processor (DSP) blocks. In addition, the Arithmetic Logic Unit (ALU) of each processor contains only the necessary resources for the algorithm implemented in it. The ALU uses floating-point arithmetic, while allowing fast software development and accurate results [17].

The structure of the FFT and Wavelet blocks are shown in Fig. 3. It is important to mention that main function of these blocks are inner product among the residue and the dictionary elements calculation. The FFT block performs the FFT of the input signal since it is computationally more efficient than the calculation of the inner products.

Fig. 3. Logic circuit used for each stage of system processing.

The function of this block is take the results of the inner products, updates the representation and the residue, as well calculate the residue energy to evaluate the stop criterion. In this way, this block has three inputs that are the outputs of the FFT and Wavelet blocks and the input signal, and generates two outputs, the residue and the Control signal, that indicates if the stop criterion was reached or not.

In this structure the FIFO1 stores the input signal, and the FIFO2 and FIFO3 stores the output of FFT and Wavelet block respectively. The MUX is used to determine which FIFO the processor will read and the process works as follows. At first iteration the processor must read the input signal and stores it in its internal memory, the read of FIFO1 is done only in the first iteration since the signal is the same for all the others. At each iteration it is necessary to read the FIFO2 and FIFO3 that contains the inner products results. With these values the solution is updated and the new residue is generated.

V. RESULTS

The tests were performed using a predefined dictionary constituted by 100 elements of the Fourier base at harmonic frequencies and 128 Wavelet elements based on Daubechies 5 mother wavelet. The sampling frequency used was 7280 Hz, and the signal was segmented in 128 points segments, that contains one cycle of the fundamental component of 60 Hz.

The tests were performed in two stages, first the quality of the representation and the compression ratio were evaluated in order to test if the algorithm is performing well in the compression of power system signal. After the parameters of the processing time and occupied logic were evaluated in order to determine the feasibility of using this technique in real time on the FPGA platform.

A. Quality and compression tests

Some metrics were chosen to evaluate the quality of the reconstructed signal after decompression. They are: Normalized Mean Squared Error (NMSE), defined in (5); the Cross-Correlation (COR) between the reconstructed signal and the original signal, defined in (6); and the Percentage of Energy Retained (RTE), defined in (7):

\[
\text{NMSE} = \frac{||x - \hat{x}||^2}{||x||^2} \quad (5)
\]

\[
\text{COR} = \frac{x^T \hat{x}}{x^T x} \quad (6)
\]

\[
\text{RTE} = \frac{\sum_{n=0}^{N} |\hat{x}[n]|^2}{\sum_{n=0}^{N} |x[n]|^2} \quad (7)
\]

where \(x\) represents the vector notation for the original signal, \(\hat{x}\), is the vector notation of the reconstructed signal and \(N\) is the signal length.

In the case of a signal compression application, the objective is to find an approximation that is close to the original signal and at the same time compact, that is, that uses few elements of the dictionary. Therefore in conjunction with the quality
of approximations, the number of elements used must also be evaluated.

The number of dictionary components and the metric results for each case are described in Table I and Table II respectively.

1) Signal with harmonics: For the first test, a signal constituted only by harmonic components was used. The input signal is described by Equation (8), where \( f \) represents the fundamental frequency of the signal. The relation between the reconstructed signal and the input signal is shown in Figure 5.

\[
x(t) = \sin(2\pi ft + \frac{\pi}{3}) + 0.5\sin(6\pi ft + \frac{\pi}{6}) + 0.2\sin(10\pi ft + \frac{\pi}{4})
\]

(8)

Fig. 5. Comparison of Results for case 1.

It can be seen that, for this case, the MNSE is low and that the values of COR and RTE are close to 100%, implying a good fidelity in the reconstruction of the input signal by the method. The absolute error is also very small (in the range of 10\(^{-3}\)), this is due to the fact that the input signal is stationary and is well represented by few Fourier components.

2) Interharmonic signal: The second test consists of two harmonic sinusoidal components added to an interharmonic sinusoidal component, as described by Equation (9). The relation between the reconstructed signal and the input signal is shown in Figure 6.

\[
x(t) = \sin(2\pi ft) + 0.1\sin(6\pi ft) + 0.05\sin(2.5, 3.\pi ft)
\]

(9)

Fig. 6. Comparison of Results for case 2.

For this second case the MNSE increased in relation to the previous case. This increase is due to the presence of a component outside the Fourier dictionary, the interharmonic. In contrast, the values of COR and RTE remain close to 100%, showing a good efficiency of the developed system. It is also noticed a greater absolute error for this case (in the range of 0.03), this difference occurs due to the limitation that the reconstructed signal can use only 8 (eight) dictionary coefficients of the dictionary. One difference from the previous case is the presence of Wavelet components of the dictionary to represent the chosen signal.

3) Signal with ramp sinking in magnitude: For this third case the generated test signal is constituted by a sinusoidal component that presents a variation of amplitude. The amplitude of the signal varies in step, with a negative change of 40% of its original value, after reaching this value it remains constant for a third of the period of the signal, then returning in step to its original value. This signal is described by Equation (10). The relation between the reconstructed signal and the input signal is shown in Figure 7.

\[
x(t) = A(t).\sin(2\pi ft)
\]

(10)

Fig. 7. Comparison of Results for case 3.

For the third case presented, the MNSE decreased in relation to the previous case, showing effective compression in this type of disturbance in the input signal. Another fact to emphasize is the values of COR and RTE that keeps close to the ideal value. It is visually evident that in observing the signal amplitude transition, small differences occur in relation to the original signal, just as in the previous case these differences occur due to the limitation that the reconstructed signal can use only 8 (eight) dictionary coefficients. In this case the compression system also needed to use Wavelet components to represent the signal, due to the non-stationary nature present in the signal.

4) Signal with exponential variation in the magnitude of a signal component: For the fourth case the generated test signal consists of a fundamental sinusoidal component added to another sinusoidal component in which its amplitude presents an exponential variation. The amplitude of the twenty-first harmonic component varies exponentially by simulating a transient variation of an electric signal. After reaching 7.5% of the fundamental component amplitude, exponential growth is interrupted. For 0.002 seconds the constant remains, and after this time interval returns to zero in the form of a decreasing exponential. This signal is described by Equation (11). The relation between the reconstructed signal and the input signal is shown in Figure 8.

\[
x(t) = \sin(2\pi ft) + 0.075u(t).\sin(2\pi.21. ft)
\]

(11)

Fig. 8. Comparison of Results for case 4.

It is noticed that for this case a small quadratic Mean Error, slightly higher than the cases of ramp variation. The values of cross correlation and Percentage of maintained energy remained close to 100%, this implies a good efficiency of the proposed system. It is possible to note small differences with the input signal, just as in the previous cases these differences occur due to the limitation that the reconstructed signal can use only 8 (eight) dictionary coefficients.

Table I and Table II summarizes the results due to the type of disturbance in the input signal. Another fact to emphasize is the values of COR and RTE that keeps close to the ideal value. It is visually evident that in observing the signal amplitude transition, small differences occur in relation to the original signal, just as in the previous case these differences occur due to the limitation that the reconstructed signal can use only 8 (eight) dictionary coefficients.

<table>
<thead>
<tr>
<th>Case</th>
<th>Number of Fourier and Wavelet Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

Table I

Table II

For the third case presented, the MNSE decreased in relation to the previous case, showing effective compression in this type of disturbance in the input signal. Another fact to emphasize is the values of COR and RTE that keeps close to the ideal value. It is visually evident that in observing the signal amplitude transition, small differences occur in relation to the original signal, just as in the previous case these differences occur due to the limitation that the reconstructed signal can use only 8 (eight) dictionary coefficients.
B. Real-time tests

To the real time implementation some important issues have to be considered, such as computational cost, adequate hardware, processing time.

To evaluate the real-time implementation feasibility, a simulation of the implemented circuit was done, and the number of clocks needed by each processing block, for representing each cycle of the signal were computed and are summarized in Table III.

<table>
<thead>
<tr>
<th>Case</th>
<th>Normalized Mean Squared Error (NMSE)</th>
<th>Cross-correlation (COR)</th>
<th>Percentage of Energy Retained (RTE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>0.1321%</td>
<td>99.9857%</td>
<td>100%</td>
</tr>
<tr>
<td>Case 2</td>
<td>1.0148%</td>
<td>99.6321%</td>
<td>99.2519%</td>
</tr>
<tr>
<td>Case 3</td>
<td>0.2426%</td>
<td>99.7394%</td>
<td>99.4726%</td>
</tr>
<tr>
<td>Case 4</td>
<td>0.3165%</td>
<td>99.9723%</td>
<td>100%</td>
</tr>
</tbody>
</table>

It is important to note that the FFT and Wavelet blocks runs in parallel, so as the Wavelet block is slower, its number of clocks should be considered and added to the number of clocks of the Compression block to obtain the number of clocks of each iteration. In this way a minimum of 1045795 clock cycles are needed in to process one cycle window. Considering the sampling rate of 7680Hz a minimum clock frequency of 63 MHz is needed. As it is possible to operate the FPGA with frequencies of up to 200MHz, it is possible to conclude that the developed system is capable of being implemented in real time.

VI. CONCLUSION

This paper presented a study and implementation in real time of a sparse representation technique in redundant dictionaries, the Matching Pursuit (MP) algorithm. A dictionary formed by the union of the Fourier and Wavelet basis was used due to the power system signals characteristics. The techniques featured in the issue of DLP logic optimization, was the method using the embedded processor. It allowed the flexibility of implementing several algorithms using the same hardware. The proposed system, although with a high computational cost, is capable of being implemented in real time, with some limitations, but still, it is able to reconstruct the signals with high quality.

Several electrical signals were tested, and, based on the results it is possible to conclude which developed system achieved the objectives that were proposed to accomplish. It has been shown to be effective, providing good levels of compression to the tested signals, and mainly, maintaining a good level of reconstruction. It is a very useful tool to solve several problems of storage of signals coming from electrical power systems.

REFERENCES