Synthetic software tool for Compressive Sensing reconstruction

Sanja Zuković, Milica Medenica, Irena Orović, Srdjan Stanković
Faculty of Electrical Engineering
University of Montenegro
Podgorica, Montenegro

Abstract— **A synthetic software tool for the reconstruction of Compressive Sensed signals is proposed. Compressive Sensing is a new signal sensing approach aiming to decrease the requirements for resources in real digital systems. Using very complex mathematical algorithms, it is possible to reconstruct the Compressive Sensed signals using just a small number of randomly chosen samples. Accordingly, the proposed software comprises and implements different signal reconstruction algorithms, providing different reconstruction performances. There is also an open possibility to include other methods within the software. Here, we will present just some of the most important algorithms and functionalities provided by the proposed tool. The software options and efficiency will be demonstrated on synthetic and real-world signals.**

Keywords-compressive sensing, sparsity, synthetic software

I. INTRODUCTION

Standard signal acquisition methods are based on the Shannon-Nyquist theorem, which requires signal sampling with frequency at least two times higher than maximal signal frequency. In this way, the sampling process results in a large number of samples which will further require large storage capacities, especially for signals with high frequency range. Compressive Sensing/Sampling (CS) [1]-[5] receives a lot of interest in the last decade, as a new emerging method for signal sampling and recovery. The CS provides efficient signal analysis and reconstruction using a very small set of samples [6], [7]. The signal should be sparse in a certain transform domain, i.e., it can be represented using small number of nonzero coefficients. The reconstruction procedure is usually based on different norm minimization: l_0, l_1, l_2 etc. The commonly used is optimization based on l_1 norm minimization, which is solved using convex optimization algorithms [5], [6]. Much faster reconstruction algorithms, called greedy algorithms, are based on iterative procedures: Orthogonal Matching Pursuit (OMP), Gradient Pursuit (GP), CoSaMP, etc. [6], [8], [9]. These algorithms have shorter execution time compared to the convex optimization methods, but may assume that the number of signal components is a priori known.

In this paper we propose a software tool for CS signal reconstruction that enables a comparison between various available methods for CS reconstruction. It allows users to choose the most appropriate approach for a particular signal. Furthermore, it provides an efficient analysis for a large class of sinusoid-like signals, analytical and real ones. The proposed tool includes the following optimization algorithms: OMP, minimization using primal-dual interior point method and the non-iterative method proposed in [10].

The paper is organized as follows. In Section II, the theoretical background on CS is given including the reconstruction algorithms. The overview of the synthetic software properties, facilities and performance are given in the Section III, while experimental results are presented in the Section IV. Section V contains concluding remarks.

II. THEORETICAL BACKGROUND

A. Compressive Sensing

Consider a real-valued, one-dimensional signal x, which can be viewed as an $N\times1$ column vector in R^N with elements $x[n]$, n = 1, 2,..., N. The signal x can be represented in terms of basis of $N\times1$ vectors $\{\psi_i\}^N_{i=1}$ that defines a certain transform domain basis. Using the $N\times N$ basis matrix ψ with the vectors $\{\psi_i\}$ as columns, the signal x can be represented as:

$$x=\sum_{i=1}^{N} s_i\,\psi_i=\psi s \tag{1}$$

where s is the $N\times1$ column vector of transform domain coefficients. Furthermore, we assume that only M measurements (arranged in $M\times1$ vector y) from x are available. The measurement matrix Φ that selects M out of N samples is of size $M\times N$. According to the CS theory, a set of measurement can be acquired as [1]-[3], [11]:

$$y=\Phi x. \tag{2}$$

It is said that $\boldsymbol{x}$ is sparse if it can be well approximated using $M<<N$ coefficients in $\boldsymbol{s}$ (only few coefficients have large magnitudes and the others are zero). Then, from (1) and (2), y can be written as:

$$y=\Phi x=\Phi\psi s=\theta s \tag{3}$$

where $\theta=\Phi\psi$ is an $M\times N$ matrix. System (3) is undetermined system of equation and therefore, it is solved by using optimization algorithms [12] - [14].

B. Signal Reconstruction Algorithms

In the sequel, we will describe several approaches to reconstruct the original signal from the subsampled random

This work has been supported by the project CS-ICT (New ICT Compressive sensing based trends applied to: multimedia, biomedicine and communications), funded by Montenegrin Ministry of Science.

measurements. All optimization algorithms are based on finding the sparsest solution of the undetermined problem.

1) L1 minimization implementation algorithms are usually based on the standard interior-point method. To recover the sparse signal x from its measurements $y=\Phi x$, one needs to find the solution to the highly non-convex problem. Generally, these methods use a linear optimization problem to recover the signal, which relies on Linear Programming and provides strong guarantees and stability. The original signal x is sparse in certain basis, which indicated that in this basis signal has only a small number of nonzero entries. It means that the recovery algorithm should be able to identify these nonzero entries. In other words:

$$s^*=argmin||s||_{l1} \qquad (4)$$

The key idea is to find s^* whose L1 - norm is minimal among all vectors s that satisfy (3). After minimization, we obtain the reconstructed vector s^*.

2) The second approach for sparse reconstruction is based on the greedy algorithm – OMP [7]. This method finds the support of the signal x iteratively, and reconstruct the signal using the pseudo inverse. Greedy methods are usually much faster but not always accurate as the convex optimization methods. In the OMP, the residual r is firstly set to the measurement vector y. Then, at each iteration we choose the column of Φ that is most strongly correlated with the remaining part of the signal. Then we subtract its contribution from the measurement vector and iterate on the residual.

3) Unlike the previous two methods, the third algorithm represents the non-iterative signal reconstruction solution [10]. The algorithm is based on calculating the variances of values that a random process takes (taking random K-samples positioned of original signal). After that, algorithm use only values which are smaller than average value of variance multiplied by already known ratio. Among already two presented algorithms, non-iterative has received much attention because of its features for fast reconstruction and good efficiency for sparse signals.

Variance-based non-iterative algorithm for sparse signal reconstruction could be summarized as follows [10]:

1. Choose M random positioned samples.

2. Calculate variance at each random position, according to: $V(k)=\mathrm{var}(x(q_M))e^{-j2\pi kq_M/N}$, $k\in(0,N-1)$ where x is signal of length N and $k=0,...,N$.

3. Determine the signal support as follows:

$$k_{0_i}=\arg\min\{V(k)<T\},\ \ for\ \ k=1,...,N\ .$$

Parameter T represents certain threshold, calculated as e.g. $\alpha\cdot\max\{V(k)\}$ and $0.85\le\alpha\le0.95$ [10].

4. Form CS matrix θ. Matrix rows correspond to the available measurements positions, while matrix columns correspond to the extracted frequencies k_{0_i}.

5. Solve optimization problem in the form: $X=(\theta^*\theta)^{-1}\theta^*y$.

III. SOFTWARE TOOL FOR CS

The software tool for CS signal reconstruction is presented in this Section. Software is implemented in Matlab 7 and it uses several Matlab features and its toolboxes [15]. The software interface is shown in Fig. 1. It allows user to create experiments that provide run-time access to design parameters. These variable design parameters are made accessible from the front panel in the form of controls, or inputs, in the interface. The complete system design and simulation can co-exist in a single software with graphs and other useful analysis indicators, or outputs, on the front panel as well. In this way, the user can change design parameters and see immediate feedback. The software interface consists of two main parts: ***Definition*** and ***Results***. The ***Definition*** part contains: the block for ***signal selection***, the ***percentage of measurements choice*** block, the ***algorithm choice*** block and ***domain choice*** block. The part ***Results*** consists of ***numerical results*** block and ***graphical results*** block. ***Numerical results*** block contains error between original and reconstructed signal (MAE - Mean Absolute Error, MSE - Mean Squared Error) as well as execution time and algorithm in use. The part denoted as ***Graphical results*** has original and reconstructed signal plots in time and frequency domain. The users initialize CS method in software by defining the input signal type (***Signal selection*** block). The input signal can be chosen from the list of defined signals where several illustrative examples are offered. Choosing already defined signal, the user can select signal which is consisted of two or more sinusoids with different frequency and amplitude or can upload real data audio signal. Specifying, for example, an audio signal, users have opportunity to hear sound of uploaded signal in order to compare the quality of the recovered signal. Otherwise, the users can create their own signals with different frequencies and amplitudes. Also, the signal length should be specified. When user specifies signal with these parameters, the feedback is given in the graphical form – signal in time and frequency domain. In software tool, random measurements are selected as percentage of original signal which is taken (***Percentage of measurements choice*** block). The user can choose the domain of signal sparsity: Discrete Fourier Transform or Discrete Cosine Transform domain (DCT or DFT). The choice of different reconstruction algorithms is provided within ***Choose method*** panel: L1-norm based algorithm, non-iterative algorithm, or OMP (***Algorithm choice*** block).

By pressing CS button, the chosen reconstruction algorithm will be run and recovered signal will appear in the plotting window (time and frequency domain). Output is at the same panel as input so it is easy to compare recovered and the original signals. The numerical result (MAE, MSE, execution time for running algorithm) are above the graphs so user can easily realize the performance of algorithm.

One of the benefits of working with presented software is that users can optionally see graphical result of reconstruction error. Using ***graphic error icon*** graphical results are shown in new panel which represent absolute error. Using numerical and graphical results various parameters can be monitored in order to make conclusions about the performances of the algorithm.

The proposed tool can provide analysis for a large class of signals in real-world applications. With this approach, applying new types of signals or new algorithms is as simple as replacing an icon that represents one algorithm by another. For

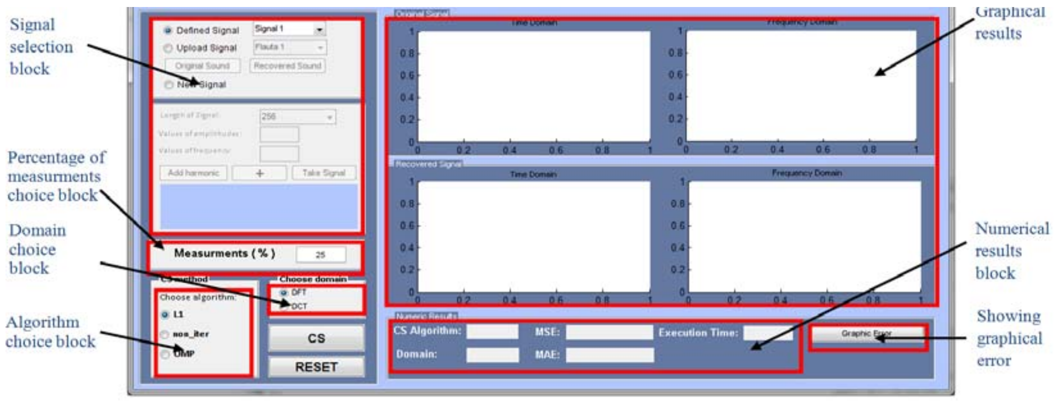


Fig.1. The outlook of the proposed software tool

example, one can easily test different methods for different types of signals, or for different transformations, or percentage of measurements.

IV. Experimental Results

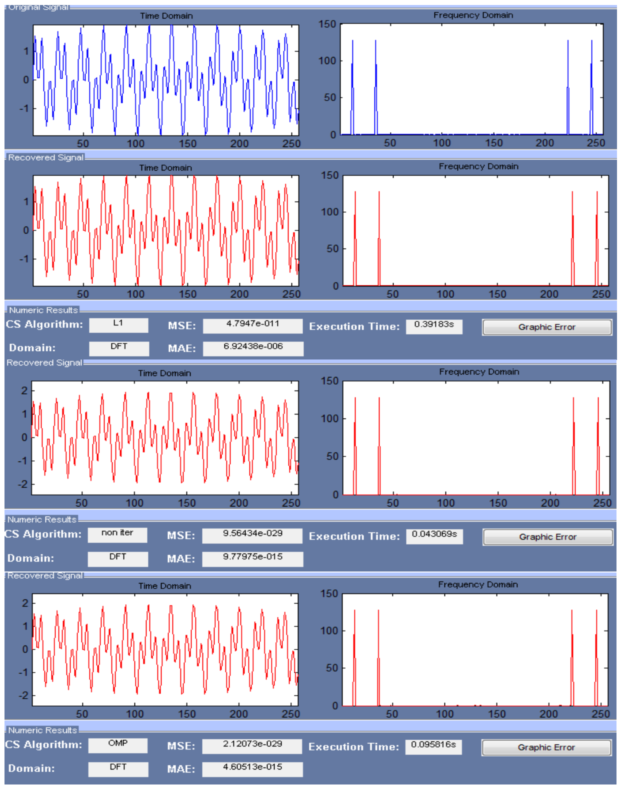
The simulation results are presented in this Section in order to examine the behavior of the proposed software tool for certain examples of signals. We have considered two simulation models to compare the performance of the different reconstruction algorithms. The chosen signals are sparse in frequency domain and random samples are taken in time domain. The properties, advantages and constraints of reconstruction algorithms are analyzed and compared.

Model 1 (Defined Signal)

In this example we observe a signal consisted of two sinusoids with different frequencies and amplitudes. The number of samples is 256. After running the reconstruction algorithms, for the measurements representing only 25% of original signal, the results are shown in Fig 2. The user may choose a domain within the ***Domain choice block***. The DFT domain is chosen as domain where the signal is sparse. As we can see from the results, OMP algorithm has the best performance (MAE = 10^{-15}; MSE = 10^{-29}; exec. time = 0.0958s for OMP and MAE = 10^{-6}; MSE = 10^{-11}; exec. time = 0.3918 for L1). Non-iterative method has the smallest execution time, and the same precision as in the case of L1 magic (MAE = 10^{-15}; MSE = 10^{-29}; exec. time = 0.043s for non-iterative algorithm). Users optionally can see graphical result of error recovered signal by pressing "***graphic error***" icon. Graphical results will be shown in new panel (as shown on Fig. 3) which represent the absolute error. Error is shown within the regular as well as within the zoomed graph (for small values of absolute error). Consequently, users can see the difference between original and recovered signal.

Model 2 (Uploaded Signal)

Furthermore, the proposed synthetic software allows one to upload signals from a certain file. An illustration for a real musical signal is given in Fig. 4 (input option ***Upload a signal***).



CS Algorithm: L1 MSE: 4.7947e-011 Execution Time: 0.39183s
Domain: DFT MAE: 6.92438e-006

CS Algorithm: non iter MSE: 9.56434e-029 Execution Time: 0.043069s
Domain: DFT MAE: 9.77975e-015

CS Algorithm: OMP MSE: 2.12073e-029 Execution Time: 0.095816s
Domain: DFT MAE: 4.60513e-015

Fig. 2: Original (blue) and recovered (red) signal, reconstructed with L1, non-iterative and OMP using 25% samples of original signal: time and frequency domain

For real musical flute signal, we measured 30% of signal samples (3000 samples). After running the reconstruction algorithms, we get the result as in Fig 4. For real musical signals, we have used the DFT Domain.

It is important to emphasize that the magnitudes of DFT components of musical signals may differ between each other more than 150 times (when observing different frequency regions). Hence, it is not easy to reconstruct the smallest components in a single iteration (using non-iterative algorithm). Namely, in order to reconstruct small components it is necessary to remove the influence of the larger ones, as the iterative algorithms do. Therefore, for small number of samples, the MSE and MAE of non-iterative algorithm are larger compared to L1 and OMP, although it can provide very low execution time. The achieved errors are provided below, for L1 and OMP algorithm:

L1: MAE = 0.0521; MSE = $2.71*10^{-3}$; exec. time = 24.925s

OMP: MAE = 0.0368; MSE = $1.35*10^{-3}$; exec. time = 72.557s

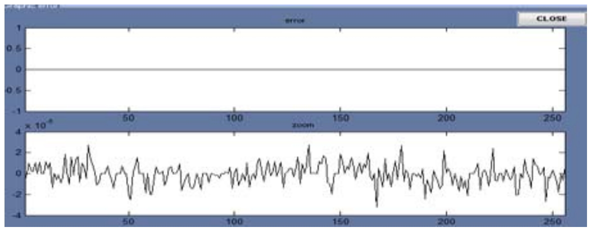
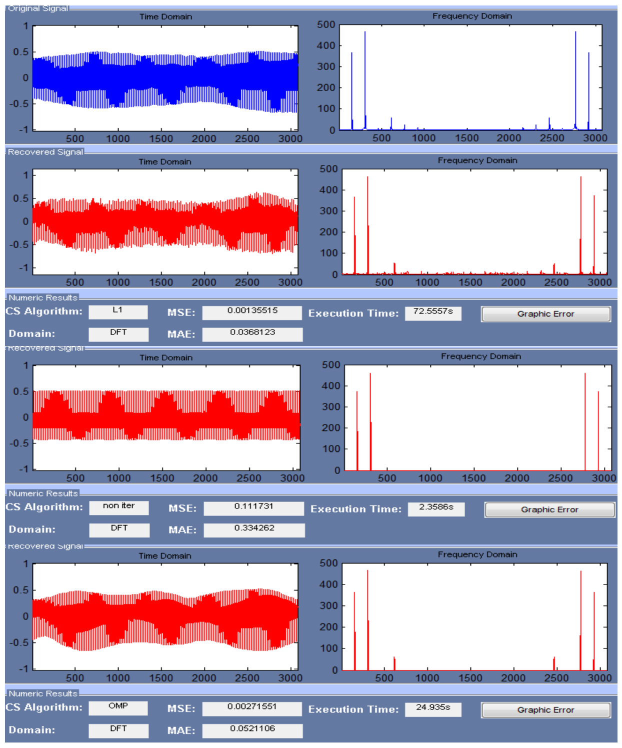


Fig. 3: Absolute error for L1 magic



CS Algorithm: L1 MSE: 0.00135515 Execution Time: 72.5557s
Domain: DFT MAE: 0.0368123

CS Algorithm: non iter MSE: 0.111731 Execution Time: 2.3586s
Domain: DFT MAE: 0.334262

CS Algorithm: OMP MSE: 0.00271551 Execution Time: 24.935s
Domain: DFT MAE: 0.0521106

Fig. 4: Original (blue) and recovered (red) flute signal, reconstructed with L1, non-iterative and OMP using 30% samples of original signal: time and frequency domain

V. CONCLUSION

In this paper we propose software tool that implements and compares different CS reconstruction algorithms. It represents a user-friendly tool that can serve to researchers and practitioners in the area of CS. The design of software allows us to: choose different input signals and return information within the implemented software; execute actions (recovery algorithms for CS method) and represent graphical and numerical results; help the user to adjust the parameter settings for these actions. Software can be used for different types of signals, different domains of signal sparsity and different number of available samples. The tool measures the performance of the algorithms using mean absolute error, mean square error and execution time.

REFERENCES

[1] E. J. Candès, M. B. Wakin, "An Introduction To Compressive Sampling", IEEE Signal Processing Magazine 2008.
[2] D. Donoho, "Compressed sensing," IEEE Transaction on Information Theory, vol. 52, no. 4, 2006, pp. 1289 – 1306.
[3] R. Baraniuk, "Compressive sensing," IEEE Signal Processing Magazine, vol. 24, no. 4, pp. 118-121, 2007.
[4] S. Stankovic, I. Orovic, M. Amin, "Compressed Sensing Based Robust Time-Frequency Representation for Signals in Heavy-Tailed Noise," Information Sciences, Signal Processing and their Applications, ISSPA 2012, Canada, 2012.
[5] L. Stankovic, S. Stankovic, I. Orovic, M. Amin, "Robust Time-Frequency Analysis based on the L-estimation and Compressive Sensing," IEEE Signal Processing Letters, vol. 20, no. 5, pp. 499-502, 2013.
[6] M. Elad, "Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing," Springer 2010.
[7] S. Stankovic, I. Orovic, E. Sejdic, "Multimedia Signals and Systems," Springer-Verlag, New York, 2012.
[8] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," IEEE Transactions on Information Theory, vol. 50, no. 10, pp. 2231–2242, 2004.
[9] T. Blumensath, M. E. Davies, "Gradient Pursuits," IEEE Transactions on Signal Processing, vol. 56, no. 6, pp. 2370-2382, June 2008.
[10] S. Stankovic, L. Stankovic, I. Orovic, "A Relationship between the Robust Statistics Theory and Sparse Compressive Sensed Signals Reconstruction," IET Signal Processing, special issue on Compressive sensing and robust transforms, in print, 2013.
[11] J. Romberg, "Imaging via Compressive Sampling," IEEE Signal Processing Magazine, March 2008.
[12] S. Stankovic, I. Orovic, M. Amin, "L-statistics based Modification of Reconstruction Algorithms for Compressive Sensing in the Presence of Impulse Noise," Signal Processing, vol. 93, no. 11, November 2013, pp. 2927-2931.
[13] P. Boyd, L. Vandenberghe, "Convex Optimization", Cambridge University Press, Mar 8, 2004 - Business & Economics - 716 pages.
[14] D. L. Donoho, Y. Tsaig, "Fast Solution of l1-norm Minimization Problems When the Solution May be Sparse," IEEE Transactions on Information Theory, Nov. 2008.
[15] I. Orovic, M. Orlandic, S. Stankovic, Z. Uskokovic, "A Virtual Instrument for Time-Frequency Analysis of Signals with Highly Non-Stationary Instantaneous Frequency," IEEE Transactions on Instrumentation and Measurements, vol. 60, no. 3, pp. 791 - 803, 2011.