Parallel Matching Pursuit Algorithm to Decomposition of Heart Sound Using Gabor Dictionaries

Carlos I. Nieblas Afirme Bank, System Area, Monterrey, México Email: 0carlos.nieblas@afirme.com

Abstract-In order to improve the performance of matching pursuit algorithm, we propose a Parallel Matching Pursuit algorithm to decompose phonocardiogram sounds of long duration. The main goal is to demonstrate the performance of the Parallel Matching Pursuit algorithm (PMP) compared with traditional iterative matching pursuit algorithm to decompose normal and pathological heart sounds of Phonocardiogram (PCG) using a Gabor dictionary. This architecture is implemented in open source Java SE 8 using a concurrency library, which is able to reduce computational cost using multi-threading until 83 % compared with traditional Matching Pursuit. Java language is widely used in the creation of web components and enterprise solutions so based on this point the main idea of this research is to set the base to implement Parallel Matching Pursuit algorithm (PMP) on web platforms focused on the monitoring of heart to sounds. This implementation allows exploring and applying iterative algorithms or sparse approximation which require processing long audio signals with low processing time.

Index Terms—Matching pursuit, gabor dictionary, parallel processing, heart sounds, sparse approximation

I. INTRODUCTION

Noncommunicable diseases, principally cardiovascular diseases, are one of the main causes of 80% of all deaths in the Region of the Americans [1]. Having monitoring tools that allow detecting pathological sounds signals might reduce the risk of death. The cardiac auscultation (listening heart sounds using a stethoscope) through different devices offers the possibility to recover Phonocardiogram signals (PCG) to be processed using different processing methods. Motivated by processing the heart sounds and sparse decomposition of signals, this study aims to set the base for creating PGC monitoring tools using Parallel Matching Pursuit algorithm (PMP) algorithm on web components to ease access by different platforms on the cloud.

The Matching Pursuit (MP) is an iterative greedy algorithm aimed at finding a sparse combination of waveforms called atoms which extract high level signal features; these atoms belong to a redundant dictionary of functions [2]. The decomposition of signals over a family of functions localized both in time and frequency have been studied to find a sparse representation of signals in several applications [2].

The first challenge is to make the MP tractable to process signals of long duration. Exists different studies about MP algorithm and how to make them tractable and fast like Matching Pursuit Tool Kit (MPTK) [3], however its implementation on monitoring platforms is limited. The MP algorithm has shown its efficiency to represent an audio signal as a combination of waveform for several goals i.e., detection, segmentation and denoising [4], [5].

The iterative algorithm demands a high computational cost if we want to process long duration signals. Minimizing the performance time has been already studied proposing different variants to this algorithm [6]. Some techniques to improve the performance of iterative algorithms like MP focus on proposing concurrent solutions aimed at real-time processing and modeling signals [7], [8]. To improve the performance and decrease the response time the selection of the dictionary plays an important role to find the best approximation with minimal iteration number of MP.

The MP algorithm has a disadvantage. The accuracy of constructed approximations depends on a suitable dictionary of functions. Incrementing the number of functions into the dictionary allows improving the sparsity but at the same time increases the time and computing complexity. The choosing of time-frequency atoms might get different properties in the decomposition of the signal.

The decomposition of phonocardiogram signals using MP and dictionary Gabor has been studied achieving good results in segmentation and denoising of this kind of signals considering the minus number of functions into the selected dictionary [9], [10].

The implementation of MP due its computational costs has been considered too slow to be applied to real-life. Motivated for this limit we propose a Parallel Matching Pursuit algorithm (PMP) implemented in Java to minimize the processing time of heart sounds signals.

The main goal of this article is to explain the architecture implemented in detail to achieve a better performance of MP algorithm using multi-thread in Java SE 8 in order to decompose heart sounds like a combination of Gabor waveforms.

Manuscript received October 21, 2021; revised December 11, 2021.

In this study we present a Parallel Matching Pursuit algorithm (PMP) implemented in Java SE 8. This paper is organized as follows: Section II describes the traditional Matching Pursuit algorithm. Section III shows the Gabor dictionary used to extract characteristics of heart sounds signals. Section IV describes our main contributions i.e. Parallel Matching Pursuit algorithm (PMP) to accelerate the response time to find the best approximation of normal and pathology heart sounds. Section V shows the experimental results comparing the sequential MP algorithm with Parallel MP. Finally, section VI presents the main conclusions and future work for this paper.

II. MATCHING PURSUIT ALGORITHM

Matching Pursuit (MP) is an iterative greedy algorithm that decomposes a signal x(t) into a sparse combination of waveforms that extract main characteristics of the signal x(t). The decomposition is achieved by projecting the signal x(t) over a redundant dictionary of functions $D=\{g(t)\}$ called atoms. Finally the signal x(t) can be reconstructed through the sum of I optimal atoms and residual term $R_I(t)as$

$$x(t) = \sum_{i=1}^{l} \alpha_i g_i(t) + R_I(t)$$
 (1)

Whit energy conservation equation

$$||x||^{2} = \sum_{i=1}^{I} \alpha_{i}^{2} + ||R_{I}(t)||^{2}$$
(2)

where g(t) and α are the *i*-th optimal atoms and coefficient respectively. The MP algorithm decomposes the residual $R_I(t)$ by projecting over *D*, before the first iteration the residual correspond to the signal x(t), *i.e.*, $R_0(t) = x(t)$. The optional match is selected. See Algorithm 1 (step 5). This procedure is repeated each time until the number of iterations *I* or the desired threshold has been reached. The best approximation yfit(t) is obtained from the sum of functions $g_i(t)$ that are chosen to be the best match its residues each iteration *i*.

Star	Algorithm 1. Traditional Matching Pursuit		
Step			
1	Input: $x(t)$; $D = \{g(t)\}$		
2	output: <i>yfit(t)</i>		
3	r(t) = x(t)		
4	repeat		
5	$g_i(t) = a r g m a x < r(t), g(t) > $		
6	$\alpha_i = < r(t), g_i(t) > $		
7	$r(t) = r(t) - \alpha_i^* g_i(t)$		
8	$yfit(t) = yfit(t) + \alpha_i^* g_i(t)$		
9	Until the desired threshold or <i>I</i> iteration has been reached.		

The MP depends of chosen the optimal atoms among all functions into D = g(t).

We suppose the atom $g_{\gamma 0} \epsilon D$. The vector x can be decomposed into

$$x = \langle x, g_{\gamma 0} \rangle = g_{\gamma 0} + Rx$$
 (3)

where Rx is the residual vector after approximating x in direction of $g_{\gamma 0}$. The vector $g_{\gamma 0}$ is orthogonal to Rx, hence

$$||x||^2 = |\langle x, g_{\gamma 0} \rangle|^2 + ||R_x||^2$$

In order to minimize || Rx ||, we must find the $g_{\gamma 0} \in D$ such as $| < x, g_{\gamma 0} > |$ is maximum. See Algorithm 1 (step 5).

The suit selection of dictionaries to find a representation sparse prove that the norm of the residual R_x decays exponentially reaches the desired threshold with the minimum number of iterations.

III. TIME-FREQUENCY GABOR DICTIONARY

The decomposition over family of functions belongs to the dictionary, well localized in time and frequency has been studied for different searchers to find particular properties in different kinds of signals. Window Fourier transform, wavelet transform are examples of timefrequency signal decomposition.

A good selection of dictionaries is important to achieve the most sparsity using MP. The nature and features of signals to process is the first step to select the adequate dictionary i.e., if the waveform of x(t) is similar to the waveforms in the dictionary, incrementing the absorption of energy by each optimal atom. The size of the dictionary is another factor we need to consider to control process time in each iteration of the MP algorithm.

Gabor waveforms are obtained by dilating, translating, and modulating a mother window w(t), which is generally real-valued, positive and of unit norm $\int |w(t)|^2 dt = 1$

$$g_{\gamma}(t) = \frac{1}{\sqrt{(s)}} w(\frac{t-u}{s}) e^{i2\Pi\xi(t-u)}$$
(4)

where s is used to control the width of waveform envelope, the time displacement u is used to specify the temporal location and the ξ is the frequency, see Fig. 1.



Figure 1. Gabor atom with s = 128, $\xi = 3$ and u = 128.

We consider $\gamma = (s, u, \xi)$ as element of the set Γ . The factor $\frac{1}{\sqrt{(s)}}$ normalize to 1 the norm $g_{\gamma}(t)$ centered in u. The energy of $g_{\gamma}(t)$ is concentrated in the neighborhood of u and the size is proportional to s, and its Fourier transform $g_{\gamma}(\omega)$ is centered at frequency ξ with a dispersion in frequency of the order of $\frac{1}{s}$, [11]. These three components are enough to creating the dictionary or family of functions to decompose the signal x(t) which align with its structure and properties.

The Gabor dictionary is the set $D = \{g_{\gamma} \in \Gamma = R_+ X R^2\}$ of Gabor atoms with scale s > 0, time location $u \in R$ and frequency $\xi \in R$.

We have elaborated a test using Gabor dictionaries to decompose phonocardiogram sounds. We found that Gabor functions can model PCG heard sounds very well at detection and segmentation [9]. The good selection of $\gamma = (s, u, \xi)$ might decompose the normal heart sounds (S1, S2) using a minimum number of atoms of Gabor.

The Fig. 2 shows a Gabor dictionary with 16 atoms considering s = 128, $\xi = 3$ for different temporal locations u.



Figure 2. Gabor atom with s = 128, $\xi = 3$ and u = [124 - 140].

IV. PARALLEL MATCHING PURSUIT ALGORITHM

In this section we describe our main contribution Parallel Matching Pursuit algorithm (PMP). The group and block concepts were added to traditional MP algorithms to divide concurrently the performance of classical MP.

The first step consists in segmenting the signal x(t) using a size window wlen = 256 samples *i.e.*, we create a different block b_j of x(t) with the same size (256 samples), see Algorithm 2, step 5-7.

In the creation of our dictionary $D = (g_{\gamma}(t))_{\gamma \in \Gamma}$, we define Gabor atoms with scale parameter s = 256 i.e., $D = (g_{\gamma}(t))_{\gamma = (256, u, \xi) \in \Gamma}$. Our studies using Gabor atoms with s = 256 have reported a good sparsity with the MP algorithm for PCG signals 11025 Hz [9]. For computational convenience we only use atoms with length of powers of two, additional we use FFT (Fast Fourier transform) to find the best frequency each block,

for this reason is convenient keep power of two in the definition of scale parameter s for the all atoms into D, see Algorithm 2, step 9.

The second step consists in creating groups θ to apply the traditional MP algorithm to each group z simultaneously. The number of groups z depends on the number of the available threads ρ in the system. Each thread executes a specific task MP task over a particular group, see Algorithm 2, step 15.

	Algorithm 2. Parallel Matching Pursuit (PMP)			
Step				
1	Input: $x[n]$, wlen = 256, ρ = 6			
2	output: <i>yfit[n]</i>			
3	r[n] = x[n]			
4	$\tau = \text{length}(x[n]) / \text{wlen}$			
5	repeat j			
4	ip = 1 + j * wlen			
5	fp= wlen + j*wlen			
6	b_j [n] = r(ip : fp)			
7	end until j= τ			
8	repeat i			
9	$\xi_j = FFT(b_j[n])$			
10	repeat z			
11	ip2=z*sec			
12	fp2 = sec + z*sec -1			
13	$\theta_z = \{b_{ip2} \ [n], \ \xi_{ip2}; \ b_{ip2+1} \ [n], \ \xi_{ip2+1}; \dots$			
13	$b_{ip2+2}[n], \xi_{ip2+2}; \dots; b_{fp2}[n], \xi_{fp2}\}$			
14	end until $z = \rho$			
15	yAprox[n]= MP_task (θ_z)			
16	r[n] = r[n]-yAprox[n]			
17	yfit[n] = yfit[n] + yAprox[n]			
18	end until <i>i =I</i>			

Each block b_j is treated as a signal to process using the traditional MP algorithm. MP decomposes a signal in a sum of waveforms belongs to dictionary, in each block b_j the family of functions in D is built using the frequency ξ that correspond with a specific block, see Algorithm 3. The group θ_z is constituted of a set of blocks b_j where a group is executed as a task by thread. This approach allows minimizing the number of atoms in the dictionary to be considered in each block.

C.	Algorithm 3. MP_task function		
Step			
1	Input: $\boldsymbol{\theta}_{z}$		
2	output: <i>yAprox</i>		
3	$b_{j}\xi_{j}$ =getblocks(θ_{z})		
4	repeat j		
5	$D=$ createDictionary $(\xi_j);$		
6	$g = argMax \langle b_j, D \rangle $		
7	$lpha \ = < b_j$, $g >$		
8	$\mathbf{r} = \mathbf{r} + \boldsymbol{\alpha} * \mathbf{g}$		
90	yfit = yfit + r		
10	until all blocks b_{j,}has been reache d		

The number of Gabor atoms in the dictionary in each block depends of the length of the block, in this particular case we define the length block *wlen* =256 that correspond with the length of Gabor atoms, therefore we can apply the inner product among a specific block and Gabor atoms belong to a dictionary. To make sure optimal atoms is selected we create a redundant dictionary with the parameter displacement u = [1,2,3,...256]. Every block performs 256 inner products for the one iteration of the traditional MP algorithm. Consequently, the number of atoms in the dictionary implemented in each block corresponds to 256.

The PMP algorithm basically encapsulate the traditional, matching pursuit dividing the signal to process x(t) in blocks of the same size and generate groups to implement a concurrent process and execute the MP to each block simultaneously.

The last step focuses on concatenating the best approximation in each block for all groups in the system for each iteration of MP. The detail of this method can be appreciated in Fig. 2.



Figure 3. Detail of PMP architecture for 1 thread applying the block and group concepts.

Typically each group contains a number of blocks which depends on the number of the size of the signal x(t). The whole PMP algorithm is described in the form of pseudo code in Algorithm 2. In addition, Fig. 4 shows a complete perspective of the normal heart sound signal using the PMP algorithm.



Figure 4. Parallel Matching Pursuit algorithm implementing multithreading architecture.

A stopping criteria was added to the PMP algorithm and was fixed in I = 10 iterations. Fig. 5 shows the absorption of energy AE in percentage in function of number of iteration. The 99 % of absorption of energy was achieving with 10 iterations of PMP as in

$$AE = \frac{|YFIT|^2 * 100\%}{|X|^2}$$
(5)

The architecture of PMP allows to keep the number of iterations independently of the size of the signal to process due that the PMP algorithm encapsulate the traditional MP i.e., apply inner product between each block b_i and its dictionary.



Figure 5. Absorption of energy in function of number of iterations of PMP algorithm for heart sound signal.

V. EXPERIMENTAL RESULTS

In order to illustrate the results, we have applied the PMP algorithm to different heart sounds signal which

include normal and pathological sounds. The signals were samples 11,025 Hz available in WAV format. We applied the same signals to evaluate the performance time using traditional MP and PMP implemented in Java.

1) Parameters of the PMP algorithm

In the present work we applied the PMP algorithm to decompose the heart sounds signals into time- frequency Gabor atoms. The dictionaries created for each block consist of 1 block with the following [window length, window shit, FFT size] parameter to build a Gabor dictionary (in number of samples): [256,0,256]. The target of the PMP algorithm was fixed to I = 10 iterations.

2) Evaluation

The implementation of our algorithm was performed using Java SE 8 and utility classes commonly useful in concurrent programming (package java.util.concurrent). The computation was performed on intel 5. The test was carried out using the PMP algorithm and traditional MP to test the performance of booth solutions.

We selected normal heart sounds, sampled 11025 Hz. with duration of 4.33 seconds. We fixed the number of iterations in 10 getting 98% absorption energy from the reconstructed signal. Table I illustrates the results using $\rho = 6$ threats in the PMP algorithm to decompose Normal heart sounds signal (S1).

 TABLE I.
 Results of decomposition of normal heart sound signal using MP and PMP algorithm

	Iteration	Algorithm	Processing time (Seg)	Absorption of energy
1	10	PMP	1.95	99%
2	10	MP	13.1	99%

From Table I, we can appreciate that the processing time in the PMP algorithm is seven times faster than the traditional MP algorithm. With 10 iterations in both algorithms we get 99% of absorption energy. Fig. 6 shows the main cardiac cycle event of normal heart sound signal (S1).



Figure 6. Main cardiac cycle event of Normal Split S1.

We apply the same approach to decompose a heart sound signal that corresponds with a diastolic rumble. This sound was sampled 11025 Hz. with duration of 5.33 seconds. Both traditional MP and PMP proposed algorithms were evaluated in terms of the processing time achieving 99% of absorption energy with I = 10 iterations. See Table II, The number of threats in PMP was fixed to $\varrho = 6$. The Fig. 7 shows the main cycle cardiac of diastolic rumble signal.

TABLE II. RESULTS OF DECOMPOSITION OF DIASTOLIC RUMBLE SIGNAL USING MP AND PMP ALGORITHM

	Iteratio n	Algorithm	Processing time (Seg)	Absorption of energy
1	10	PMP	1.97	99%
2	10	MP	12.48	99%



Figure 7. Main cycle cardiac of diastolic rumble signal.

All the simulations carried out using the dataset of the different PCG, see Table III. The PCG signals were sampled at 11,025 Hz available in WAV format.

TABLE III. DATASET OF PCG SIGNALS USED TO DECOMPOSE SIGNALS USING TRADITIONAL MP AND PMP ALGORITHMS

	Heart sound signal	Duration time (seg)	Main cycle number
1	Normal Split S1	4.30	4
2	Normal Split S2	4.30	4
3	S3	4.30	4
4	S4	4.30	4
5	Early systolic murmur	4.30	4
6	Late systolic murmur	4.30	4
7	Opening Snap	4.30	4
8	Diastolic rumble	4.30	4
9	Ejection click	4.30	4

We present the results in processing time and absorption of energy using traditional matching pursuit and PMP proposed algorithm for different PCG signals of the Table III.

	Heart sound	Algorithm	Processing time (Seg)	Absorption of energy
1	S2	PMP	2.00	98%
2	S2	MP	13.05	98%
3	S1	PMP	1.95	99%
4	S1	MP	13.10	99%
5	Opening snap	PMP	2.01	99%
5	Opening snap	MP	14.67	99%
6	Late_Systolic_Murmur	PMP	2.14	99%
6	Late_Systolic_Murmur	MP	14.10	99%
7	Early_Systolic_Murmur	PMP	2.20	99%
7	Early_Systolic_Murmur	MP	14.34	99%
8	Diastolic_Rumble	PMP	1.97	99%
8	Diastolic_Rumble	MP	12.48	99%
9	Ejection_Click	PMP	2.01	99%
9	Ejection_Click	MP	14.56	99%
10	S3	PMP	2.12	99%
10	S3	MP	13.35	99%
11	S4	PMP	14.23	99%
11		MP	2.17	99%

TABLE IV. RESULTS OF DECOMPOSITION OF DIASTOLIC RUMBLE SIGNAL USING MP AND PMP ALGORITHM

We evaluate the PMP algorithm with heart sound of long duration which was acquired using a sampling frequency of 11025 Hz and 16 bit per sample and are stored in WAV format. We can appreciate from Table IV that the PMP algorithm finds a sparse representation for all heart sounds with processing time that correspond 15 % over the processing time of the classic MP algorithm.

Table V shows the results obtained in terms of processing time for different heart sounds using traditional MP and PMP algorithms considering long duration signals for several individuos. We can validate that even when the duration of the signal is different, the percentage of processing time is reduced by 83% using the PMP algorithm.

The dataset was obtained from Performance Evaluation of Heart Sounds Biometric Systems on an Open Dataset [12].

	Individuos	Algorithm	Processing time (Seg)	Duration of signal (Seg)
1	Female	PMP	25.87	70
1	Female	MP	169	70
2	Female	PMP	26.13	70
2	Female	MP	170	70
3	Female	PMP	21.95	60
3	Female	MP	146	60
4	Female	PMP	11.11	30
4	Female	MP	71.14	30
5	Female	PMP	10.81	30
5	Female	MP	71.10	30
6	Male	PMP	9.9	26.78
6	Male	MP	64	26.78
7	Male	PMP	26,19	70
7	Male	MP	167.62	70
8	Male	PMP	26.32	70
8	Male	MP	168.79	70
9	Male	PMP	26.22	70
9	Male	MP	168.22	70
10	Male	PMP	26.86	70
10	Male	MP	169.61	70

TABLE V. DATASET OF PCG SIGNALS USED TO DECOMPOSE SIGNALS USING TRADITIONAL MP AND PMP ALGORITHMS

VI. CONCLUSION

11.83

76.32

10.61

69.81

32.13

32.13

29.07

29.07

We have proposed an efficient parallel matching pursuit algorithm to decompose heart sound signals using Gabor dictionaries. The PMP algorithm showed high performance, achieving a reduction of processing time in 83% compared with traditional matching pursuit algorithm. We evaluated the performance of PMP for different normal and pathological PCG signals. Both traditional MP and PMP algorithms were developed on java platform. The implementation on java platform will allow creating monitoring solutions taking advantage of different open source technologies and at the same time implement sparse algorithms to detect, separating of sources or extraction of main components of heart sound signals on cloud platforms. We believe that this algorithm can be used as the base for other applications such as heart sound classification. In the future we plan to improve the performance of PMP using other techniques

11

11

12

12

Female

Female

Male

Male

PMP

MP

PMP

MP

of concurrency. The experimental results show that the PMP implementation can decrease the computation cost than the traditional MP algorithm. We consider that it is possible to minimize the number of atoms in each block and improve the processing time by PMP algorithm.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

ACKNOWLEDGMENT

The author wishes to thank CICESE and AFIRME bank for their support.

REFERENCES

- [1] Pan American Health Organization, *NCDs at a Glance: NCD Mortality and Risk Factor Prevalence in the Americas*, Washington, D.C.: PAHO, 2019.
- [2] S. G. Mallat and Z. Zhang, "Matching pursuits with timefrequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397-3415, Dec. 1993.
- [3] C. I. Nieblas. "Detection of trumpet signals using molecular Gabor dictionaries and matching pursuit algorithm," in *Proc. 8th International Conference on New Music Concepts*, Treviso, Italy, 2021.
- [4] R. F. Ibarra, M. A. Alonso, S. Villarreal, and C. I. Nieblas, "A parametric model for heart sounds," in *Proc. 49th Asilomar Conference on Signals, Systems and Computers*, 2015, pp. 765-769.
- [5] S. Krstulovic and R. Gribonval, "Mptk: Matching pursuit made tractable," in *Proc. IEEE International Conference on Acoustics* Speech and Signal Processing, 2006.
- [6] M. J. Wang, W. D. Yu, and Y. Wang, "Fast sparse signal approximation via matching pursuit," in *Proc. 2013 IEEE 10th International Conference on High Performance Computing and*

Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing, 2013, pp. 1944-1947.

- [7] L. Daudet, "Audio sparse decompositions in parallel," *IEEE Signal Processing Magazine*, vol. 27, no. 2, pp. 90-96, March 2010.
- [8] A. A. Kim, O. O. Lukovenkova, Y. V. Marapulets, and A. B. Tristanov, "Sparse approximation methods and their parallel implementation," in *Proc. 2018 3rd Russian-Pacific Conference on Computer Technology and Applications (RPC)*, 2018, pp. 1-5.
 [9] C. I. Nieblas, M. A. Alonso, R. Conte, and S. Villarreal, "High
- [9] C. I. Nieblas, M. A. Alonso, R. Conte, and S. Villarreal, "High performance heart sound segmentation algorithm based on Matching Pursuit," in *Proc. 2013 IEEE Digital Signal Processing* and Signal Processing Education Meeting (DSP/SPE), Napa, CA, 2013, pp. 96-100.
- [10] C. Nieblas, R. Ibarra, and M. Alonso, "A novel fourth heart sounds segmentation algorithm based on matching pursuit and Gabor dictionaries," *Research in Computing Science*, vol. 80, pp. 9-16, 2014.
- [11] R. Gribonval and E. Bacry, "Harmonic decomposition of audio signals with matching pursuit," *IEEE Transactions on Signal Processing*, vol. 51, no. 1, pp. 101-111, Jan. 2003.
- [12] A. Spadaccini and F. Beritelli, "Performance evaluation of heart sounds biometric systems on an open dataset," in *Proc. 2013 18th International Conference on Digital Signal Processing (DSP)*, 2013, pp. 1-5.

Copyright \bigcirc 2021 by the authors. This is an open access article distributed under the terms of Creative Commons attribution-noncommercial license (<u>CC BY-NC-ND 4.0</u>).



Carlos I. Nieblas was born in Oaxaca, Mexico. He graduated from Anahuac University in 2011. He received a MC degree in electronic and telecommunications from CICESE in 2013. His search interest includes signal processing and creating of enterprise software.