# IEEE Signal Processing Cup 2016
# Technical Report

Alexis Pomares Pastor, *Student,* Marina Cañadas Ortega, *Student,* Jorge Pereira Delgado, *Student,*
Ricardo Jiménez-Marín, *Student,* David Garrido Leal, *Student,* Manuel Montoya-Catalá, *MsC,*
José M. Leiva-Murillo, *PhD.*

***Summary*-In this report we describe the technical aspects of our solution for the 2016 Signal Processing Cup competition in which we are required to both build a system that records the power grid signal of our location and also to develop a system that obtains the ENF signal of a provided set of power grid recordings and trains a classifying system that identifies the location of other power grid recordings.**

## I. INTRODUCTION

The 2016 Signal Processing Cup is an international undergraduate competition that explores using a time-varying signature embedded in media recordings to determine the location-of-recording. It is divided into two main parts; in the first part, teams are asked to determine the location-of-recording of different .wav files containing measurements of either power or audio recordings of the power grid from 9 different locations. The algorithm developed is required to also identify measurements that do not belong to any of the training grids. The second part of the competition is aimed at recording the power grid of our location. For doing so, teams must implement a hardware and software system that is able to preprocess the grid signal, read it, and finally store it into the computer.

The competition involves time/frequency signal processing, elementary statistics and machine learning, and both analogic and digital electronics for acquisition. In the next Section, we describe our approach to the extraction of the signal that describes the evolution of the electric network frequency (ENF), together with the statistics we obtain from the signal for the identification. In Section III, we describe the statistics we have extracted from the ENF and the classification method applied on the features extracted from the ENFs. The design of the hardware for the measurement of the ENF, its analog-digital conversion (ADC) and the procedure followed for sending and storing the data on a computer is described in Section IV. We finish the report with some conclusions in Section V.

## II. EXTRACTION OF ENF SIGNALS

The input files provided in the competition are .wav files containing Audio or Power recording of 9 different Power Grids. These wav files were obtained at a sample frequency of 1000 Hz and they do not only contain power (information) in their main frequency, but rather in a wide range of frequencies due to effects such as noise, sampling, finiteness of the signal and harmonics generated.

In order to obtain the ENF of the recordings we follow a very simple, yet effective approach: we obtain the spectrogram of the recorded signals using a sliding rectangular window. The frequency with the highest spectral power value in the range between 35 Hz and 70 Hz is set as the main frequency of the window. This way we avoid extracting one of the harmonics as the fundamental frequency. This kind of problem could arise due to a deficient acquisition (aliasing, clipped signals, etc). Some examples of the signals in the time domain are shown in Fig. 2.1.

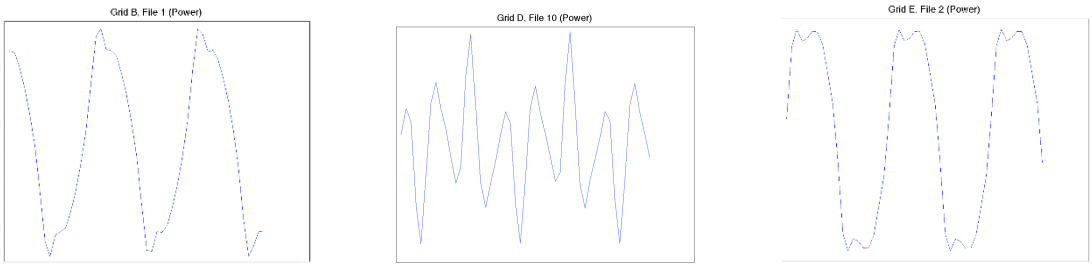



Figure 2.1: Some *pathological* signals found in the training database. 2.5 cycles of the signal are shown in each case.

We have decided to partition each file into segments of 3 minutes and 20 seconds. This way, the number of samples available to train our system is increased, assuming that the relevant information to identify the grid is contained in a segment of that length. On the other hand, a probabilistic per-segment classifier can easily deliver probabilistic estimates for per-file classification.

The rectangular window size and the number of samples of the fast Fourier Transform (FFT) are chosen according to the sampling frequency and the resolution desired in frequency. We have chosen a value of a 100.000 point NFFT which allows for a frequency precision of 0.01 Hz in the power spectral density (PSD). An example of spectrogram and the subsequent ENF are shown in Fig. 2.2.

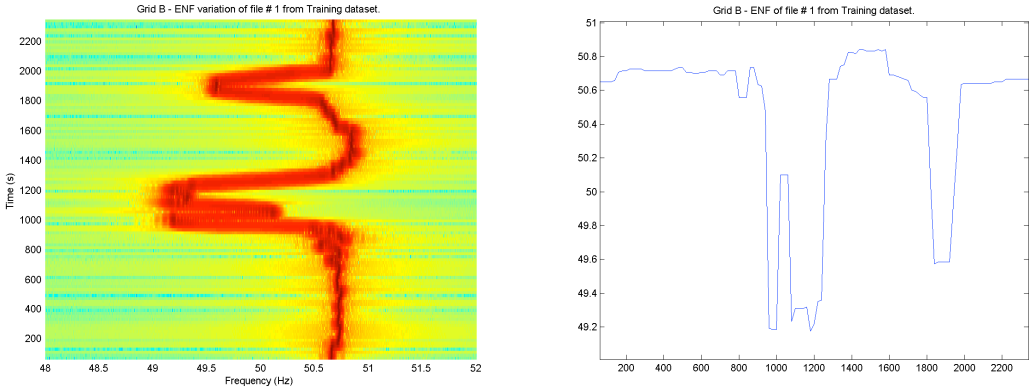



Figure 2.2: Example of spectrogram and the corresponding ENF.

## III. LOCATION CLASSIFICATION/IDENTIFICATION SYSTEM

In the following, we describe the identification system implemented. Following our previous experience in machine learning and data processing competitions, we have based our design decision on the idea of "the simpler the better" [4]. This philosophy also ensures the proper understanding of the underlying theory towards the undergraduate students of the team that were not previously familiarized with machine learning algorithms. We first describe the features we have obtained and then explain the classifying system used.

### *A. Feature Extraction*

We extracted an initial set of 16 features composed of basic statistics from the ENF signal and its derivative as well as information about the relative strength of the peak in the PSD. We performed feature selection using cross-validation, which led us to a final set of 6 features that we employed in our classification system, they are described in Table I. As we can see, there is a slight difference between the set of features employed for Power files and the ones used for Audio files. Due to the noisiness and lesser samples of the Audio signals, features 5 and 6 are not used in for classifying Audio files.

Table I: Features extracted from the ENF

| Index | Features | Used on Audio Files | Used on Power Files |
|---|---|---|---|
| 1 | Mean of the ENF | Yes | Yes |
| 2 | Mean of log(1+variance of ENF) | Yes | Yes |
| 3 | Variance of ENF | Yes | Yes |
| 4 | Variance of log(1+ENF) | Yes | Yes |
| 5 | Mean(Peak value of *PSD* /Power) | Yes | No |
| 6 | Mean(log(Peak value of *PSD*/Power)) | Yes | No |

The purpose of features 5 and 6 is to evaluate the *neatness* of the sinusoids, as it provides a measurement of the concentration of the signal over the fundamental frequency. This ability to quantify the noise present in the sample is exploited in order to differentiate Audio files from Power ones. The differentiation is accomplished by means of the 5$^{th}$ feature. The training set reveals that Audio and Power segments can be totally differentiated from each other by means of using a threshold over this feature, the threshold is automatically computed as the middle point between the minimum value for the training Power segments and the maximum value for the training Audio segments. If this feature has a high value, it means it has "clean" spectra so we consider it is Power-type; if it is below the threshold, it is "noisy" and thus Audio-type.

*B. Classification*

The classification system built has two main particularities as we already stated. First, it contains two different classifiers at its core, one for Audio and another for Power signals, the discrimination between them is performed using an automatic threshold on the the 5$^{th}$ feature. Second, it divides the input .wav files into blocks of 200 seconds (200.000 samples), being each block an independent sample for the classification system. The cross-validation is done in such a way that ensures that all segments of a file are either in the training fold or in the validation one.

The main reason why Audio and Power classifiers are trained separately is because we observed significant statistical differences between the Audio and Power files of the same location and since our basic classifier algorithm does not have the enough expressivity to describe them both jointly, we decided to make this distinction. We also tried training the classifier with both Power and Audio but the results confirmed the previous observation, better accuracy values are found when classifying audio signals using only audio segments for training, and the same applies for power signals.

Although we performed some experiments with k-nearest neighbors (KNN) classification, we finally decided to use a generative classifier in which a Gaussian distribution is fitted with the training data from each grid; then, each test sample is classified by means of a maximum likelihood criterion. We have considered three different methods [3]:

*i)* A classifier in which there are no constraints on the covariance matrix of each Gaussian.

*ii)* A model in which the covariance matrix of all grids is forced to be diagonal, and

*iii)* A shared covariance matrix across classes (homoscedastic model).

These methods have been enumerated in decreasing order of complexity (number of parameters), in order to avoid the over-parameterization and the subsequent overfitting. We have found that the optimal configuration involves a full model for the Power files and a homoscedastic model for the Audio signals. This selection was performed using cross-validation accuracy,

The main advantage of this generative approach is that it easily allows for the combination of the *soft* (probabilistic) predictions at a segment level into per-file labeling. Basically, the segments are assumed to be independent and identically distributed data, so that the likelihood of a file given a class is given by the product of the probability of all the segments from the file (for a typical length of 10 minutes, three segments of 3:20 are needed).

Finally, we obtain the estimated likelihood of a segment belonging to the whole set of grids as a linear mixture of the nine Gaussians evaluated on that point. This way, we label as *N* ("none of the grids") all files that have a joint probability below $10^{-22 \cdot Ns/3}$, being *Ns* the number of segments in which the file has been divided. This threshold has been found reasonable after exploring the cross-validation probability estimation together with the estimations on the provided validation dataset.

### *C. Description of the Software*

A Matlab library has been provided with the following modules:

- **extract_features.m** is the Matlab function that obtains the ENF and then the discriminative features of the signal given as input.
- **obtain_X_Y.m** and **obtain_Xtest.m** are the functions that load the training or the practice/testing files, respectively, and call **extract_features.m** to build the data matrices and label vectors.
- **predict_test.m** is the function that builds a classifier from a training dataset and performs classification on a test set.
- **get_file_prediction.m** is the function that combines per-segment predictions to obtain per-file estimations.
- **crossValidation.m** partitions the data into disjoint training and validation sets to obtain an estimation of the classification error. It is used for model and feature selection.
- **main.m** is the executable script that calls the previous function in order to read data from the files, get cross-validation estimations, and apply the classification models to the practice and testing datasets. The execution of the different parts can be enabled and desabled thourgh flags located at the beginning of the file.

The sequence obtained for the practice dataset is:

AHCFF,BGINH,AFBDI,INCAE,HNBAD,CGHGB,HDCHG,EACHI,BHECF,FCGEI

Through the feedback system provided by the competition, we noticed this prediction has an accuracy value of 82%. Since our cross-validation accuracy value was 85% for power files and 47% for audio files, and considering that there are more Power files than Audio files and also the class "N" identification we concluded that our validation accuracy was representative of the testing error and so our system does not suffer from overfitting.

The sequence predicted for the test dataset is:

NDHND,HHDAF,ACGBG,DFCEH,GNHAG,HFDAI,HNFHI,IBCBD,ENIBE,EGCAG,CINIG,
HAEFC,NHFHG,CEIGI,EICEC,BEBHA,DGHAG,AABIH,CCDBA,GBFEB

## IV. CIRCUIT DESIGN AND DATA ANALYSIS FOR ENF ACQUISITION

In this section we describe the electronic system built to read the voltage signal of our power grid and store it into a computer for further processing. First it is needed a system to acquire the signal from the distribution network that is going to be developed in subsection A. Then it is required an electronic pre-processing of the signal in order to make it possible for an analogic-digital converter to sample it. We are going to first describe the theoretical model of the circuit and we are going to use simulation software (subsection B) to back up the model. Once the simulation has successful results, the circuit is built and tested in a laboratory in order to see if it has the expected performance, explained in the subsection C. After checking with an oscilloscope that the signal meets the requirements of the analogic-digital converter, it is sampled with Arduino and stored into our computer.

### *A. Acquisition of the signal*

To obtain the power grid signal, we have developed a simple homemade system, whose fundamental part consist in a transformer of 230/15 V, given by the department of electronic technology in our university. The nominal input voltage (230V) has been chosen due to its use in domestic installations, and the output one (15V) is given by the transformer itself, we didn't have access to other types of transformers.

The socket-transformer adaption was built using an old laptop charger cable; the male ending of the cable was cut and welded to the transformer in order to connect it to the power grid. The output of the transformer is carried out by two rigid cables with a section of 0.5 $mm^2$, which are commonly used in electronic circuits. The transformer is also protected by a little plastic box to electrically isolate it. Figure 4.1 shows the used device.


Figure 4.1: Transformer.

### *B. Simulations of the signal's pre-processing system*

Once the signal from the grid is acquired at a voltage level suitable for standard circuits, it's necessary to pre-process it in order to make it possible for the analogic-digital converter (further ADC in this text) to sample it. Since we used an Arduino UNO device, we have to meet the signal requirements of its ADC. Arduino UNO only accepts positive voltage values with a maximum of 5V so we developed an adder circuit to convert the 15 $V_{rms}$ signal in a 5 $V_{pk}$ one with an offset of 2.5 V. The design is shown in Figure 4.2.

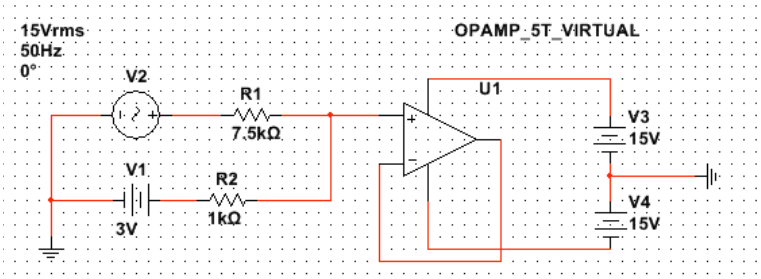



Figure 4.2. Circuit for the adaptation of the signal.

In this circuit, the alternate voltage V2 represents the output of the transformer and the output of the circuit is the output of the operational amplifier. Using National Instruments Multisim to simulate the circuit, the output signal obtained is shown in Figure 4.3.

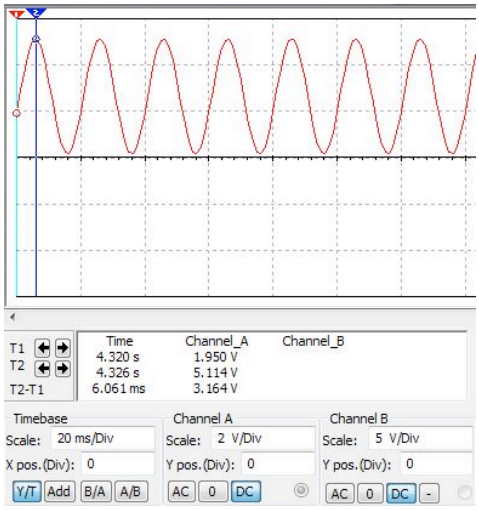



Figure 4.3. Results of the simulation.

As it can be observed, using this configuration we get a 5.1Vpk signal and all its values are positives, so it is suitable for the ADC to process it. But this signal will not be given as input to the ADC, since it is necessary a filtering process to remove high frequency harmonics that do not satisfy Nyquist criterion at sampling. The signal shown has a frequency of 50 Hz, and it has not any distortion. However, it is expected that the signal obtained from the transformer will have distortions and parasite frequencies that might make difficult the analysis of the signal. This is the reason why it is necessary a filtering process before inputting the signal to the Arduino device. The filtering is also necessary to avoid aliasing while the ADC is sampling the signal.

A Butterworth filter (Figure 4.4) was used in order to get a plain response on low frequencies, considering these low frequencies are all lower than 250 Hz (cut-off frequency, determined expecting a signal with a fundamental harmonic of 50 Hz). At these low frequencies, the signal is not attenuated neither distorted. In higher frequencies we get a significant attenuation of the signal, having a -25dB gain for 500 Hz. This consideration is important, since it is expected to do the sampling of the signal at a frequency of 1kHz and the Nyquist theorem imposes that the sampling frequency must be at least twice the maximum frequency of the signal, so we need to ensure we eliminate these high frequencies to prevent aliasing and therefore the distortion of the sampled signal. The frequency response of the filter is shown in Figure 4.4.

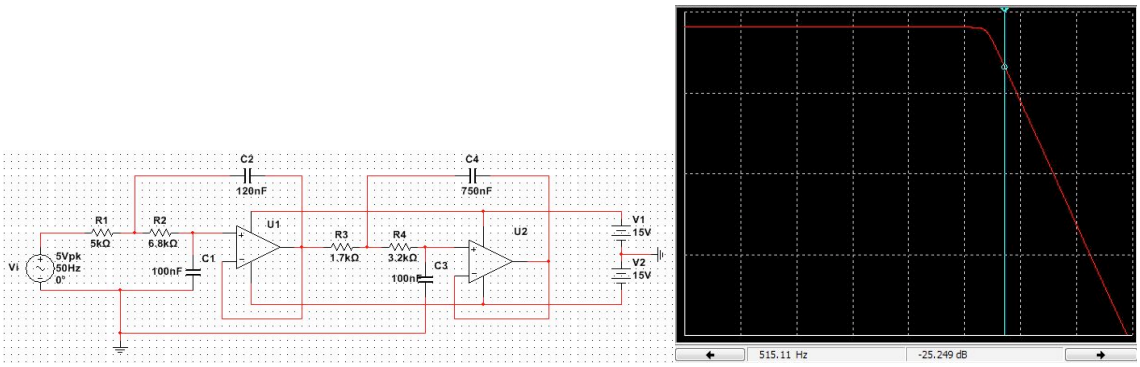



Figure 4.4: Antialiasing filter (left) and the simulated frequency response (right)

Combining the two systems independently simulated, we obtain the complete acquirement and pre-processing system, as presented in Figure 4.5.

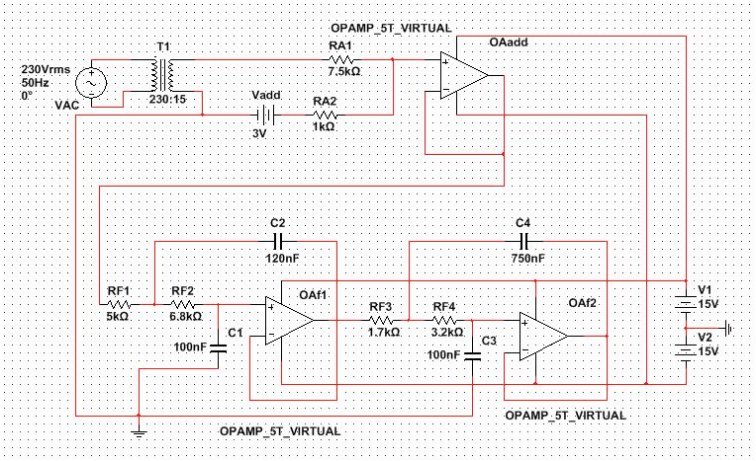



Figure 4.5: Complete system.

### *C. Experimental Analysis of the acquirement and pre-processing system*

The transformer and the circuit shown below has been built and tested. The transformer gave 16.15 $V_{rms}$, slightly higher than the 15V expected but it does not imply any problem. The circuit was at first tested independent, using an alternate voltage source to generate the signal, that was replaced later by the transformer, and two DC power suppliers to get the offset voltage ($V_{add}$) and the power source of the operational amplifiers, which will be used during all the analysis. Figure 4.6 shows the circuit built.

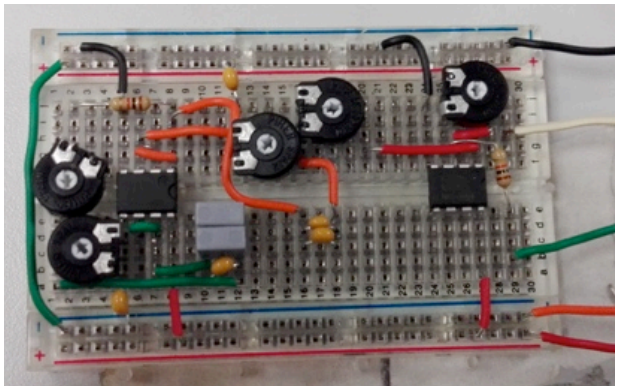


Figure 4.6. Pre-processing circuit on the protoboard.

As it can be observed, many of the resistances shown in the schematic of the circuit have been replaced by potentiometers in order to adjust the final response of the circuit. These manual adjustments were performed to compensate the differences between the theoretical model and the real model. In the circuit, the white cable is its input, the green one is for the offset voltage, red and black cables has been used to feed the operational amplifiers and ground is the orange cable.

Having obtained positive results with the first tests, obtaining a good response from the filter and an acceptable output signal, we proceeded to test the complete system. The output signal obtained is shown in Figure 4.7.

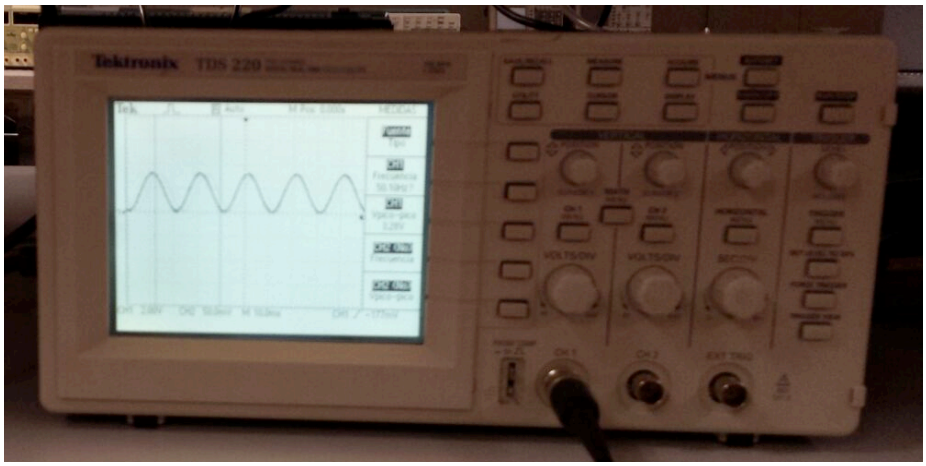



Figure 4.7: Measured signal.

The final pre-process signal is a sinusoid with a voltage value of 4.7 $V_{pk}$ being all voltage values higher than 0V, therefore it is suitable to be read by the Arduino device, since as we said, it is only able to sample voltage values from 0V to 5V. The Arduino device used to sample the signal is shown in Figure 4.8. The software codes developed for the purpose of configuring the Arduino and storing the signal into our computer are simple and efficient, they are also included in the .zip file of the submission. At a glance, the Arduino board reads the analog signal and transmits it via Serial communication to the computer, where we store all the values coming from the Serial in a CSV format file, which can be easily read from Matlab.

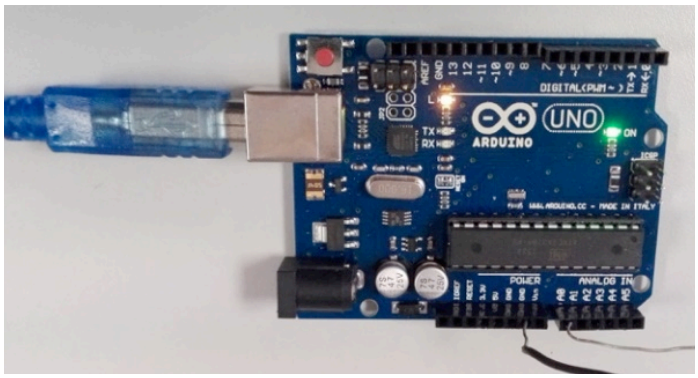



Figure 4.8. Arduino board and its connections.

To sum up, we conclude the system we have designed is valid for our purposes and we will use it to obtain the power grid signal.

### *D. Acquired signal*

In Figure 4.9 we show the spectrogram of a 10 minute segment signal obtained with our acquisition system, together with its corresponding ENF.

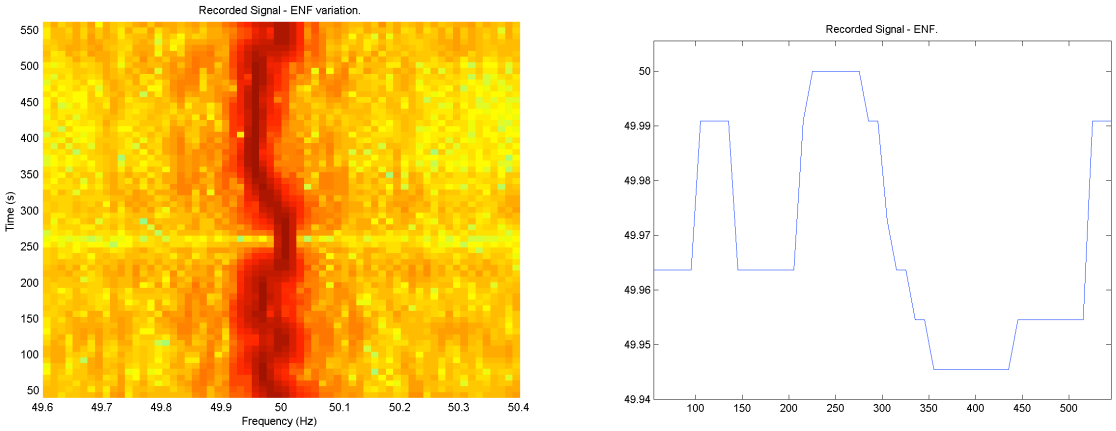



Figure 4.9: Spectrogram and ENF of our 10 min signal.

## V. CONCLUSIONS

We have concluded that the problem of grid identification from the ENF is feasible from direct voltage measurements but challenging from audio recordings. The characteristics of the Audio recordings were quite different from the Power recordings of the same grid location, even the mean frequency differs significantly. Due to the simplicity of our classifier, our system required to treat the Audio and Power signals separately. In addition, we have found anomalies in the provided data that make the problem even harder. Once a grid is identified, it is not easy to conclude whether it is due to the grid's properties or the acquisition particularities. The cross-validated accuracy of the training set is similar to that one obtained from the Practice set so we expect our system not to suffer from overfitting.

Concerning the acquisition system, we proved that is possible to build a platform for measuring the power grid signal in a simple, reliable, cheap and open source manner. The ENF signal obtained has enough information (changes) to identify the grid and does not suffer from high distorting effects that could damage the signal and make it unable to be characterized for further classification.

In any case, this competition has given us the opportunity of working together on an engineering challenge that involves signal processing, machine learning and both analog and digital electronics. It's been an enriching experience for all of us and therefore we would like to thank the Signal Processing Society for this initiative.

## REFERENCES


[1] A. Hajj-Ahmad, R. Garg and M. Wu (2015). ENF-Based Region-of-Recording Identification for Media Signals. *Information Forensics and Security, IEEE Transactions on*, *10*(6), 1125-1136.

[2] P. Top, M. Bell, D. Coyle and O. Wasynzuk (2012), "Observing the Power Grid", IEEE Signal Processing Magazine, 29(5):24-32.

[3] C. Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.

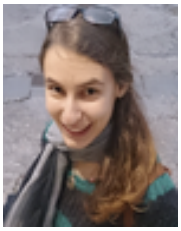
[4] J. Leiva and S. Martens (2010), MLSP Competition, 2010: Description of first place method, *IEEE Int. Workshop on Machine Learning for Signal Processing (MLSP),* 2010, Kittila, Finland.


## APPENDIX: TEAM COMPOSITION

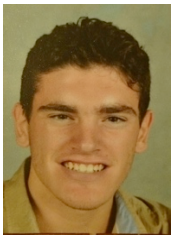
The team AC/DC is composed by three students in biomedical engineering, one student on electronic engineering, one student in enginering of multimedia systems, and a master student and a PhD on multimedia and communications.

Alexis Pomares Pastor is a student of Biomedical Engineering at Universidad Carlos III in Madrid. He comes from Tenerife, Canary Islands and he will spend the next year finishing his degree in Singapore. He participated in the national phase of Physics Olympics in 2013, and he likes traveling abroad as a volunteer and practicing sports, such as skydiving or martial arts. His academic interests include medical instrumentation, electronics, signal processing and machine learning.

Ricardo Jiménez was born in Madrid in 1995 and is a student of electronic engineering at Universidad Carlos III in Madrid. He has been awarded many times for his excellent academic career and participated in the 2013 edition of the Spain's National Physics Olympics. He is also a member of BEST organization (Board of European Students of Technology) and is determined to develop himself internationally, both academically and professionally.



Marina Cañadas Ortega is an undergraduate student in Biomedical Engineering at Universidad Carlos III in Madrid, born in Málaga, in 1995. She will expend next year studying at the University of Technology of Sydney, and her academic efforts were rewarded with an excellence scholarship for three consecutive years. She is volunteer in Red Cross and has a temporal job teaching a child. Her academic interests are mainly focused in robotics and prosthesis.



David Garrido Leal was born in Toledo, Spain, in 1995 and is a student of Biomedical Engineering at Universidad Carlos III in Madrid. His academic career has been acknowledged with a National Undergraduate Award in 2013 and participations in two national literary contests. He is also an Outdoor Activities Instructor, and is personally interested in travelling abroad and playing sports such as swinmming or handball. His academic interests include tissue engineering, medical instrumentation and signal processing, areas on which he desires to focus profesionally.

Jorge Pereira Delgado was born in Madrid, Spain, in 1992 and is a student of Engineering in Multimedia Systems at Universidad Carlos III in Madrid. In his academic career he has passed several subjects with honours, specially in the field of electronics. In 2016, he is awarded with a four months fellowship at University Carlos III, in which he will research the automatic detection of Alzheimer based on magnetic resonance imaging. In his free time, he enjoy playing the guitar, walking his dog and traveling.

Manuel Montoya is a predoctoral scholar at the Data Processing Group, Department of Signal Theory and Comunications of Universidad Carlos III in Madrid. He obtained the M. Sc. degree in telecommunication engineering from Universidad de Alcalá in 2014 and he is currently taking a M. Sc. on Machine Learning at Universidad Carlos III in Madrid. His research interests include signal processing, machine learning, neural networks and PGMs.

Jose M. Leiva-Murillo is a visiting professor at the Data Processing Group, Department of Signal Theory and Comunications of Universidad Carlos III in Madrid. He obtained the M. Sc. degree in telecommunication engineering from Universidad de Málaga in 2001 and the Ph. D. in computer science in 2007 from Universidad Carlos III de Madrid. His research interests include signal processing, machine learning, and clinical data processing.