Spectral Analysis in Mobile Communications

Alexandru CELMARE, Angela DIGULESCU, Cristi LECA, Cristina DESPINA-STOIAN,
Denis STĂNESCU and Dragoș NASTASIU

***Abstract*—Everything that is transmitted wirelessly depends on the frequency spectrum management. When radiocommunication equipment transmits simultaneously using the same frequency band, interference usually happens and the received signals are modified. One solution to monitor this problem and to assure that the message is received minimally altered can be spectral analysis. That can help estimate the noise level, frequency used, power level or even identify the type of communication being transmitted. The current experiment is aimed is at capturing the GSM (Global System for Mobile Communications) band using a Software Defined Radio and importing it in Python to process the data. Thus, the designed software accurately offers all the information mentioned above, depending on the collected data. After getting information from the spectrum, the GSM broadcast and control channels can be decoded in Wireshark for further information.**

***Index Terms*—Communication type identifier, GNU Radio, GSM, power level, SDR capture, traffic channel, unencrypted control channel decoding.**

I. Introduction

The spectral analysis is of utmost importance in radiocommunications. It does not only provide information unavailable in the time-domain, but it also defines how the frequency spectrum is managed.

This is one of the first steps that should be verified when problems on the receiver side occur. This usually happens because other devices emit unwanted signals in the same band.

Once the used radiocommunications standard is identified, more information about what is transmitted can be obtained. To work properly, every network needs to be ruled by protocols. In GSM, this is managed by the use of control channels. Some of them, such as BCCH (Broadcast Control Channel) and CCCH (Common Control Channel), are not encrypted. This means that everyone who sniffs the Air Interface (Um) can oversee a part of the control information broadcasted by the mobile station or base station transceiver, such as the cell identity of the mobile station, the power level that the base station transceiver emits with, the country, the mobile operator and so on.

The article under consideration aims to offer an insight not only on the information that can be decoded in Wireshark or in GNU Radio's QT GUI Sink, but through Python processing, the program can automatically detect the type of communication, whether the GSM channel is beacon or traffic and the power level. That could help newcomers in the domain to understand easier what is happening in GSM spectrum and to learn on their own, with the help of the program. Also, some security issues related to Air Interface and IMSI are mentioned in this article.

II. GNU Radio Processing

The GNU Radio scheme in Fig. 1 is used to capture GSM downlink band [1]. Its variables are related to Python processing. The program can only work with certain bandwidths, so first of all, the user has to introduce in Python the start and stop frequencies and FFT size to calculate how bandwidth should be modified. After getting values accepted by the program, these should be next introduced in the variables section in GNU Radio.

Signal capturing blocks must be related to the SDR used. For the processing part, the 'Stream to VecDecim' is used instead of 'Stream to Vec' to reduce the size of the captured data and therefore be able to process it faster.

That did not affect the quality, considering the TDMA structure with 8 logical channels. The block 'Complex to Mag^2' is used to express the power level and 'Log10' to express it in dB.

The purpose of this scheme is to export the data processed in Python, which means File Sink was needed. QT GUI Sink and QT GUI Vector Sink have the same output, but different types of input.

The scheme in Fig. 2 is used to analyse BCCH and CCCH, as these control channels are not encrypted [2]. No professional device is required for that. Even the cheapest SDR can capture a single channel in GSM [3].

As in every digital communication system, synchronization is essential. With the help of the synchronization block shown in Fig. 2, the receiver is aligned to TDMA structure and frequency corrections are done.

After processing, these are sent to the local host 127.0.0.1, where they will be subsequently analysed in Wireshark.

A. CELMARE is with Military Technical Academy "Ferdinand I", Communications Department, Bucharest, Romania (e-mail: celmare_george@yahoo.com)

A. DIGULESCU is with Military Technical Academy "Ferdinand I", Communications Department, Bucharest, Romania. (e-mail: angela.digulescu@mta.ro)

C. LECA is with Military Technical Academy "Ferdinand I", Communications Department, Bucharest, Romania. (e-mail: cristian.leca@mta.ro)

C. DESPINA-STOIAN is with Military Technical Academy "Ferdinand I", Communications Department, Bucharest, Romania. (e-mail: cristina.despina@mta.ro)

D. STĂNESCU is with Military Technical Academy "Ferdinand I", Communications Department, Bucharest, Romania. (e-mail: denis.stanescu@mta.ro)

D. NASTASIU is with Military Technical Academy "Ferdinand I", Communications Department, Bucharest, Romania. (e-mail: dragos.nastasiu@mta.ro)

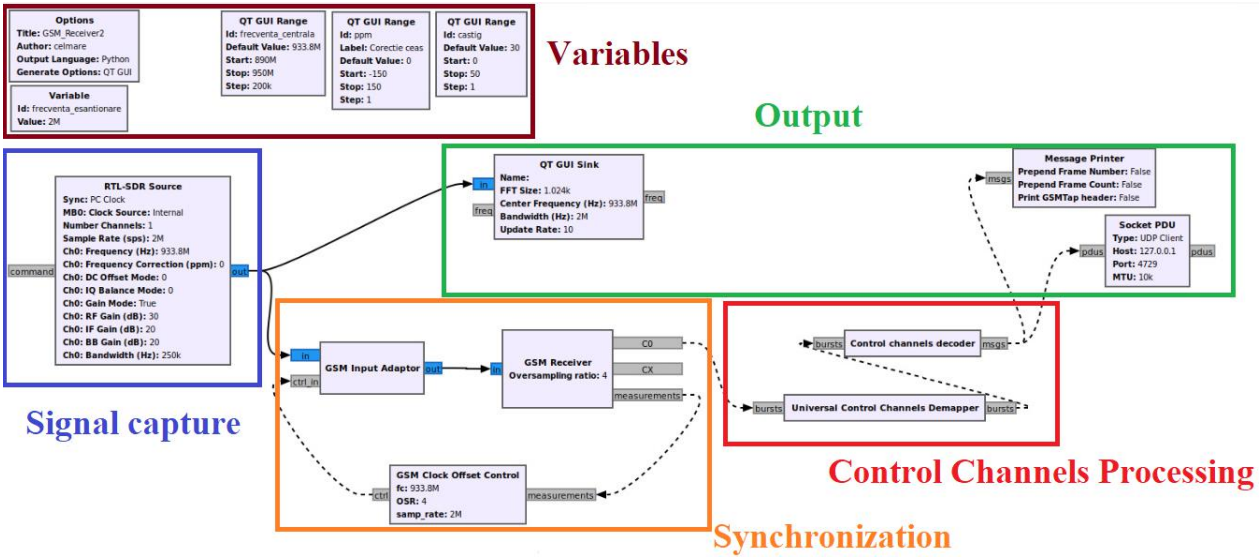

Figure 1. GNU Radio processing for Python

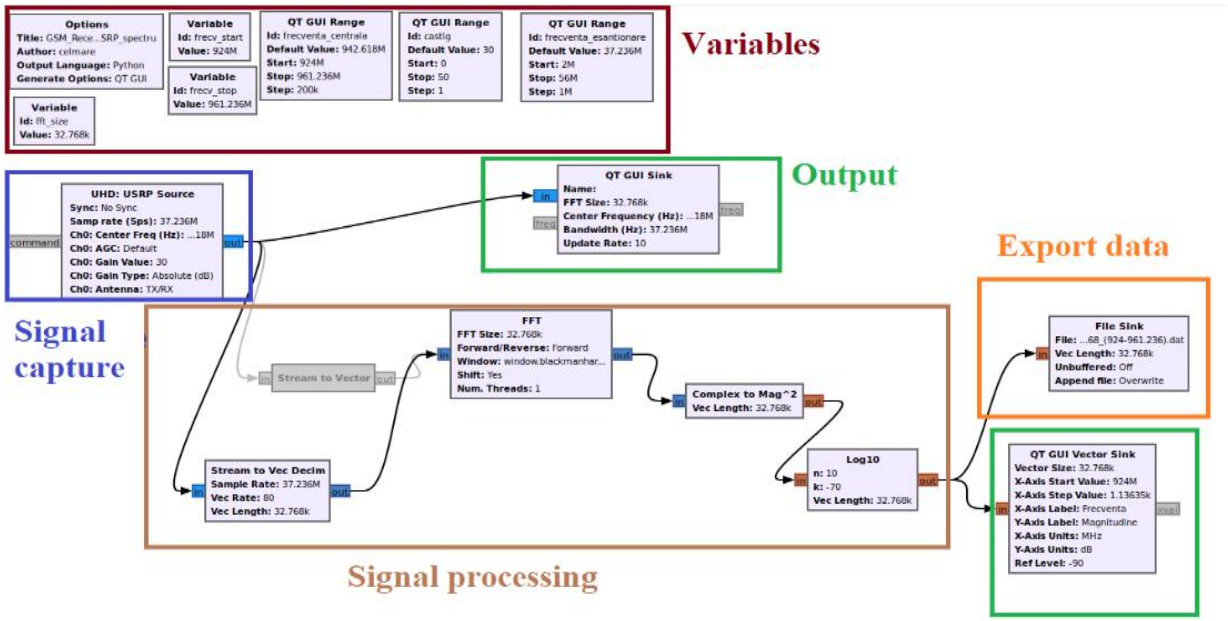

Figure 2. GNU Radio processing for Wireshark

III. PYTHON PROCESSING

The Python script is able to analyse the GSM downlink band and to detect signal and noise power, central frequencies of 2G (even with frequency hopping) and 4G channels, whether or not a 2G channel is for traffic or beacon.

To do so, we considered that a step of 200 kHz would be enough to differentiate two different channels, because this is the distance between two adjacent channels. Considering that the data is stored in samples and Ox axis shows the number of samples (not Hz), which is the same as FFT size, by analogy, we calculated the step as number of samples.

The problem consists in the fact that this step must be an integer. To meet this condition, it was not enough to round the number, because that error would propagate and could desynchronise. Therefore, if the first decimal is 0, the error can be neglected. This is the reason why this program can run only with certain data that respects the following conditions:

- starting frequency must be a multiple of 200 kHz (for instance 924.2e6, not 924.1e6);
- the bandwidth must be a number whose value divided by 200 kHz results in a step with the first decimal 0.

If the program requests the name of the file where the data is stored, it means that it can do the processing. After confirming that the parameters are convenient, these have to be introduced in GNU Radio to capture the proper bandwidth. The program will run only with a good set of data.

The next stage is supposed to determine the central frequencies and, thus, to be able to identify the channel types later. When the signal level is calculated, it takes into consideration more samples in that region. No more than 80 kHz to the central frequency are added because more bandwidth would mean entering in the transition band or in another channel. Considering that the rounding was made with floor, meaning a lack, for a longer set of data (tens of MHz) would represent a small shift before the real central frequency. This is one more reason for our choice of 80 kHz. Also, an average would show a value more related to reality.

 Digital Object Identifier 10.32754/JMT.2021.2.01

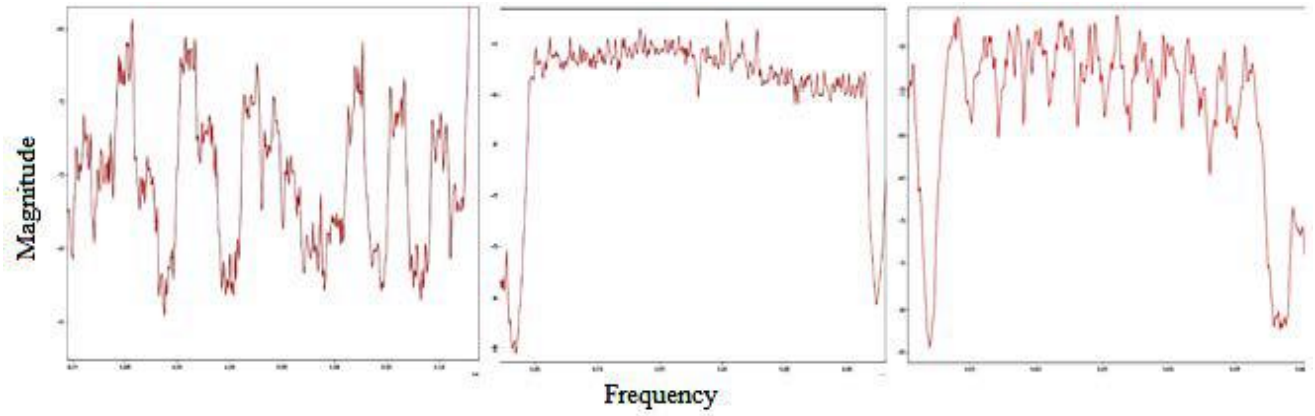

Figure 3: 2G, 2G frequency hopping, 4G spectrum

Fig. 3 shows the types of communication the program can recognise. These are obtained by keeping only maximum values for each frequency during the analysis. Instant values are not relevant in this part of identification because for traffic channels, information may not always be transmitted, as for beacons.

To be able to differentiate those three, one more vector is considered in order to store the values of the central frequencies shifted at 100 kHz, which is the end of the GSM channel. Based on it and imposing an empirical threshold, in our case 4 dB between two adjacent channels, the program makes the difference among the types of spectra.

Also, to differentiate whether a 2G channel is beacon or traffic, instant values were taken into consideration, a different threshold than the one mentioned above, and a possible error. For a channel to be a beacon [4], it means that it transmits continuously and every sample must be above the noise level. If the instant value is over the threshold (quite bigger than the noise power, because it can fluctuate), it is counted and compared to the total number of comparisons made on that frequency. In this analysis, we noticed that beacon channels never had all samples over threshold, so a small deviation is empirically set which recognizes the type in more than 85% of cases.

Another way to differentiate the beacon from traffic is to analyse the FCCH (Frequency Control Channel) [7], as in Fig. 4. Only beacons have it and can be seen in the spectrum with a small spike close to the central frequency of each channel. This solution is not chosen, because it is based on resolution, meaning that if the FFT size is too small, this channel cannot be noticed properly and that could affect the decision.

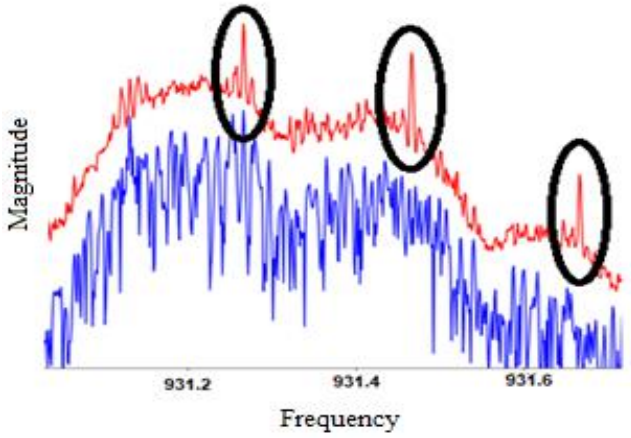

Figure 4. FCCH

To check whether the results shown by the program meet those from reality, we compared them with Fig. 5. The maximum values registered during the capture are shown in red, on the left side of the figure. That helps to identify more easily the type of communication: 2G, 2G frequency hopping [5] or 4G. Coloured in blue is an instant capture that differs every single moment of the analysis. Based on those coloured in blue, the beacons or traffic channels were determined and also the power level. In other words, for 2G channels, if the blue graph is close to the red graph all the time, that channel is a beacon. That can be verified in the spectrogram (which can be seen in GNU Radio) on the right side of Fig. 5. To check if the power level was calculated right, we should notice that it is a few dB lower than the red graph. The project was designed based on empirical formulas, which implies a continuous verification and comparison to see if the findings are accurate.

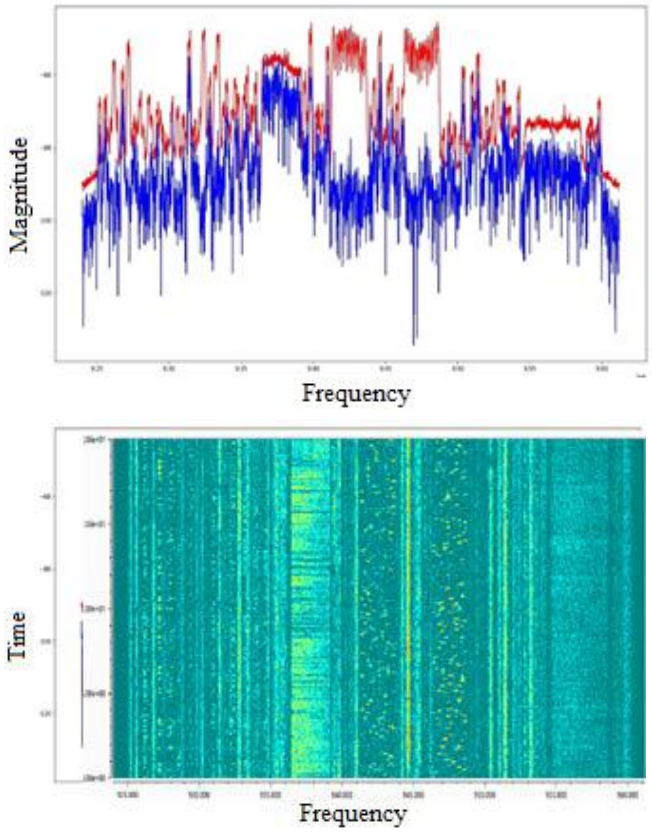

Figure 5. Frequency spectrum and spectrogram of GSM downlink band analysed

IV. WIRESHARK PROCESSING

Wireshark can be used as a packet sniffer, being passive and not interrupting the connection between two entities [3]. It analyses the protocols that rule in GSM and are sent over the air interface. This decoding was made for broadcast messages (which are unencrypted), not for the traffic, so the privacy of the subscribers is not infringed. Mentioning that, it is worth bringing up in discussion IMSI (International Mobile Subscriber Identity) and TMSI (Temporary Mobile Subscriber Number). IMSI is a unique and permanent number that identifies a SIM (Subscriber Identification Module). It means that if somebody knows the IMSI of a subscriber, it can find its location. To avoid that, on Air Interface, TMSI is widely used [6]. It is only known by the mobile station and the VLR (Visitor Location Register). Every time a mobile station changes its location area, it is given a different TMSI by its new VLR. Based on how GSM was designed, sometimes it is compulsory to send IMSI through the Air Interface (for instance, when the phone connects to the network) [3]. Unfortunately, this lack of security can be very easily exploited when malicious intention appears, not using any sophisticated equipment.

Fig. 6 shows a scenario in which BTS warns the mobile station that is getting a call through the Paging Channel (PCH). If it accepts the request, it asks through RACH (Random Access Channel) a random dedicated channel, depending on the actual resources of the network. If there is a free dedicated channel, BTS informs through Access Granted Channel (AGCH). After that, the mobile station sends its control information through the SDCCH (Standalone Dedicated Control Channel) to be allocated a traffic channel. A part of this process can be seen in Wireshark: PCH and AGCH appear as CCCH.

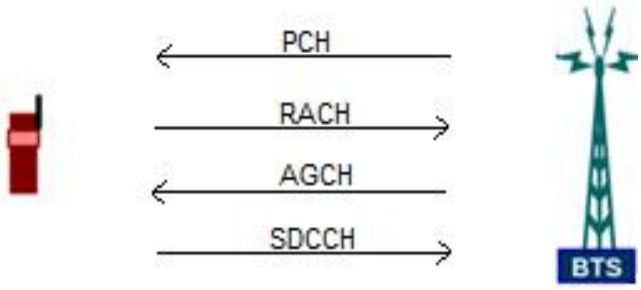

Figure 6. Channels used by BTS and mobile station to establish the connection to the network

Analysing CCCH and other packets with signalling information, can be attained information about country, mobile network, location area, cell identity, whether frequency hopping is used, emission power level, description of neighbour cells (to be prepared for handover) [3].

V. DISCUSSION

To have a better view over the entire band, a professional SDR should be used. We tried both cheap (RTL-SDR dongle) and expensive (USRP 2901) SDR and for analysing packets in GSM, it made no difference because a channel has 200 kHz bandwidth and even the weakest equipment on market this day has a minimum of 2 MHz instantaneous bandwidth. On the other side, to have a complete view over the GSM downlink band, equipment with a minimum 35 MHz instantaneous bandwidth was needed, since in Romania, it is between 925-960 MHz The USRP 2901 suited our needs, as it has 56 MHz instantaneous bandwidth.

As mentioned in chapter III, this program can work properly with only certain sets of data, whose parameters can be verified at the beginning, after introducing and running them.

VI. CONCLUSION

Besides the help offered to the newcomers in this domain, the analysis provided information about how mobile operators share the spectrum.

Even though there have been many years since the introduction of 2G, it is not obsolete and new phones are still compatible with this standard. One of the reasons why this technology is not abandoned is the fact that it has the largest coverage and there are still devices which are not compatible with newer standards. Therefore, the research efforts on this subject are still a topical issue and these types of solutions have real impact on research and testing activities.

REFERENCES

[1] "FFT GNU Radio," [Online]. Available: https://wiki.gnuradio.org/index.php/FFT. [Accessed 2021]

[2] "Intro to Software Defined Radio and GSM/LTE," [Online]. Available: https://www.blackhillsinfosec.com/intro-to-software-defined-radio-and-gsm-lte/. [Accessed 2021].

[3] K. Vachhhani, A. Dubey and D. Vohra, "Investigating GSM Control Channels with RTL-SDR and GNU Radio," in *IEEE WiSPNET 2016*, 2016.

[4] R. S. d. Castro, P. Godlewski and P. Martins, "Cognitive Beacon Channel via GSM and UMTS," *Proc. 21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Istanbul, Turky, 2010. doi:10.1109/PIMRC.2010.5671705

[5] J. Wigard, P. H. Michaelsen, P. Mogensen and T. T. Nielsen, "Slow frequency hopping solutions for GSM networks of small bandwidth," in *Vehicular Technology Conference*, 1998. doi:10.1109/vetec.1998.686454

[6] I. Marghescu, S. Nicolaescu and N. Cotanis, "Terrestrial Mobile Communications", Ed. Tehnică, 1998.

[7] L. M. Tomas, "Using the Spectrum Analyzer as an Educational," [Online].Available: http://ocw.upm.es/pluginfile.php/1140/mod_label/intro/articulot3.pdf. [Accessed 2021].

Link to code:
https://github.com/CelmareAlexandruGeorge/Spectral-Analysis-in-Mobile-Communications.git

 Digital Object Identifier 10.32754/JMT.2021.2.01