PSD of Random Signals

Leonel Ricardo Araque 2204224
Leandro José Garzón 2194232
Universidad Industrial de Santander

Github: https://github.com/leo09p/COMMII_A1_G8/tree/Practica_2

Abstract
This report presents a practical and comprehensive analysis of different types of signals in both time and frequency domains, with special emphasis on Power Spectral Density (PSD) as a key measure of power distribution across the frequency spectrum. The study involves the use of simulation software (GNU Radio) for signal processing and includes the evaluation of random, binary, audio, and image signals. These signals are examined in terms of their spectral content, bandwidth, and intrinsic properties, as well as their interaction with various modulation schemes. The main objective is to deepen the understanding of the behavior and correspondence between signals, their spectral representation, and modulation techniques.

I. Introduction

Signal analysis plays a fundamental role in the field of communications. This report explores the characteristics and behavior of various types of signals, such as bipolar random signals, binary signals, audio, and images. Through the use of the GNURadio simulation software, these signals are examined in both the time and frequency domains, allowing for a deeper understanding of their properties. Furthermore, key concepts in communications—such as interpolation and modulation—are addressed, providing valuable insight into the processes involved in analyzing and manipulating signals across different applications.

II. THEORETICAL FRAMEWORK

Understanding the Power Spectral Density (PSD) of random signals is fundamental in communications, as it enables the analysis of noise, signal processing, and the characterization of signals in telecommunications. PSD is a mathematical function that describes how the power of a signal is distributed across frequencies, offering a detailed view of its energy in the frequency domain.

Several key concepts support this analysis: the Bipolar Random Binary Signal, which represents digital data through alternating positive, negative, and zero voltage levels; the Bit Rate, defined as the number of bits transmitted per second; and the Sampling Frequency, which converts a continuous signal into a discrete one by taking samples over time. Additionally, Bandwidth determines the transmission capacity of a communication link, while White Noise, characterized by the absence of statistical correlation, exhibits a constant PSD.

III. METHODOLOGY

The laboratory was carried out based on the provided guide and supported by research on the topic. Tools such as GitHub, Linux, and Moodle were used for development and process management. Each stage of the work was documented to ensure reliable results.

A. Step 1 – Environment Preparation and Signal Characterization

The initial stage consisted of verifying the proposed flowchart through the analysis of a bipolar random binary signal with a rectangular shape. Using the repository provided in the practice file, the pre-constructed flowchart for random binary signals was configured and characterized. Key parameters, such as Samples Per Symbol (SPS), were varied to observe their effect, which are shown in Table 1.

h	SPS	rate bit	samp rate	bandwidth
1, 1, 1, 1	4	32k	128k	64kHz
1, 1, 1, 1, 1, 1, 1, 1	8	32k	256k	64kHz
1,1,1,1,1, 1,1,1,1,1, 1,1,1,1,1, 1	16	32k	512k	64kHz
1	1	32k	32k	64kHz

Table 1. Signal Parameters According to h

B. Step 2 – Analysis of White Noise and PSD

In this stage, GNURadio was configured to generate and observe white Gaussian noise in both the time domain and its Power Spectral Density (PSD). By modifying the sources in the flowchart and adjusting the necessary parameters, the PSD of the noise was obtained and analyzed.

C. Step 3 – Analysis of Real Signals (Image and Audio)

In this step, a real-world signal, specifically an image and an audio file, was analyzed in both the time and frequency domains. Using GNURadio, the source blocks (File Source and Unpack K Bits) were configured to process the input and obtain its spectral representation. Tests with images of different shades and textures, as well as with an audio file, allowed the observation of how the characteristics of each signal influence its behavior after processing.

IV. *ANALYSIS OF RESULTS*

The analysis of results was carried out following the structure defined in the methodology. Based on the data obtained during the experiment, a detailed interpretation of the signal's behavior in different scenarios was made. The main findings and their relevance within the context of the study are presented in this section.

a. Part 1 - Environment Preparation:

To begin, several tests were carried out, and the data obtained were organized in Table 1. This table shows the main characteristics of the signal when varying the number of Samples per Symbol (Sps). The results confirm that modifying Sps does not affect the bit rate or the bandwidth, since these parameters are determined by the number of transmitted symbols and the modulation type, not by the sampling process. However, an increase in Sps directly increases the sampling frequency, as more samples of the signal are taken. This growth follows a pattern of multiples of two, except for the case when Sps = 1, where it is multiplied by four.

b. Part 2 – Analysis of White Noise and PSD

In this stage, white Gaussian noise was analyzed both in the time domain and in its Power Spectral Density (PSD). The signal observed in the time domain presented the expected random bipolar behavior, which is shown in Fig. 1. Regarding the PSD of the tested images, different behaviors were noted.

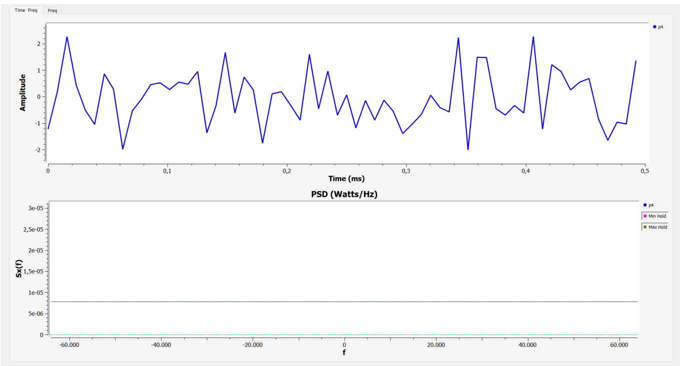

Fig. 1. White Noise and Its PSD

Test 1:

Fig. 2. Frog Image

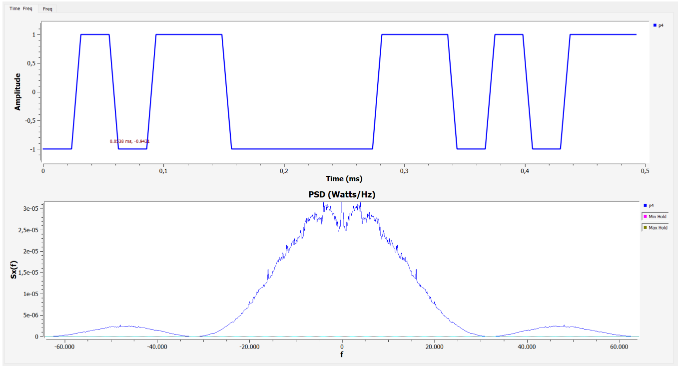

Fig. 3. Frog Signal in the Time Domain

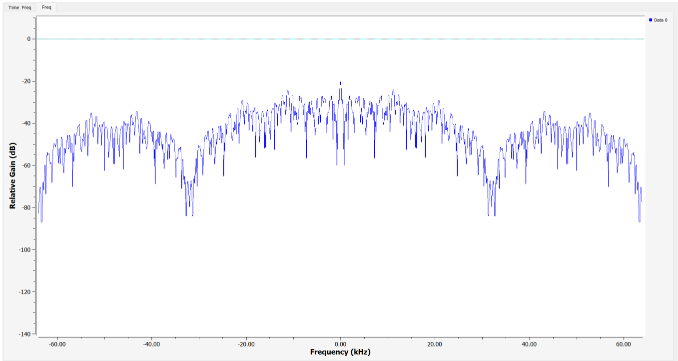

Fig. 4. Frog Signal in the Frequency Domain

Test 2:

Fig. 5. Red Background

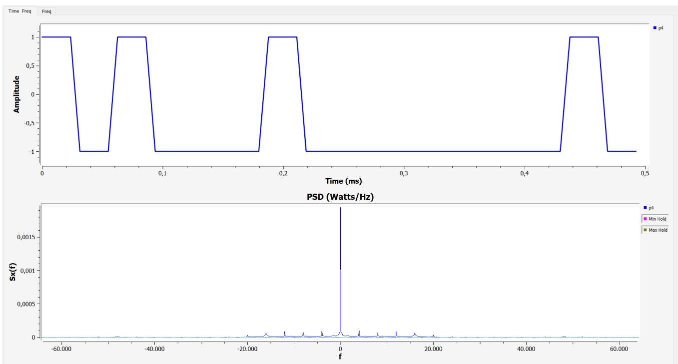

Fig. 6. Red Background Signal in the Time Domain

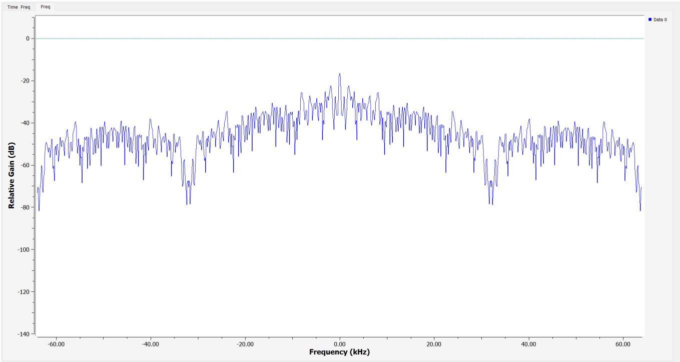

Fig. 7. Red Background Signal in the Frequency Domain

- The frog image, shown in Fig. 1, presented a relatively wide PSD, as illustrated in Fig. 3, due to the diversity of tones and textures. The dominant spectral components, highlighted in Fig. 4, corresponded to rapid changes in the image, while smoother areas contributed mainly to low-frequency components.
- The red background image, shown in Fig. 5, and its time-domain signal, shown in Fig. 6, exhibited no abrupt transitions since the image consisted of a uniform color. The PSD was concentrated at low frequencies with a dominant DC component, reflecting the absence of tonal variations.

In the frequency domain, the red background image showed a stable spectrum due to the lack of transitions, while the frog image displayed patterns resembling "hills," associated with periodic color changes and distributed energy across different frequency ranges.

Audio test:

In the second part of the process, the procedure was repeated using an audio signal source, allowing the analysis of its behavior in both the time and frequency domains.

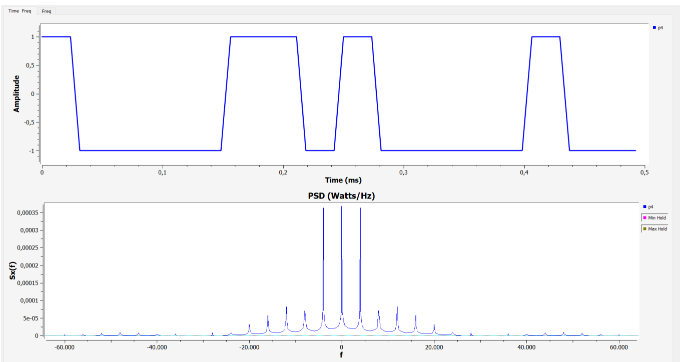

Fig. 8. Test Audio Signal in the Time Domain

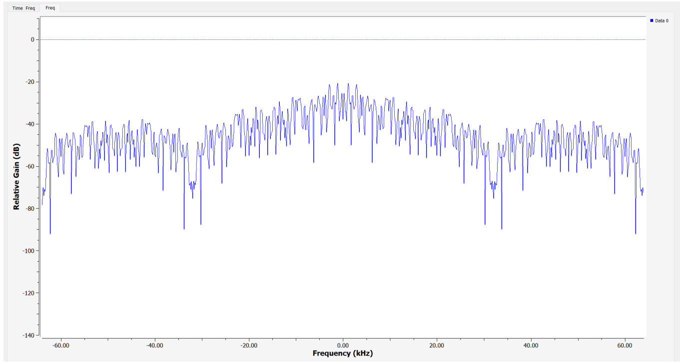

Fig. 9. Test Audio Signal in the Frequency Domain

For the audio signal, the time-domain analysis revealed pronounced peaks in the PSD, as shown in Fig. 8, caused by abrupt tonal changes in the recording. This behavior indicates the strong presence of specific frequency components that dominate the spectrum at certain moments, as illustrated in Fig. 9.

Control Question

A. This block chain recenters and rescales the incoming waveform. The constant source subtracts a fixed offset (shifting the signal baseline), and the subsequent multiply block stretches the amplitude. The combined effect is to produce a zero-centered signal with the desired amplitude range (for example mapping values into approximately −1,1).

B. The Interpolation FIR Filter upsamples the stream by inserting extra samples and filtering them to shape the spectrum. Practically, it increases the effective sample rate by the interpolation factor and applies a finite impulse response kernel (the h coefficients) to smooth and control aliasing caused by the added samples.

C. Because audio and images carry different kinds of structure: audio usually varies smoothly over time and concentrates energy in specific bands, while a binary image often contains abrupt spatial edges and textured regions that produce stronger high-frequency components. Thus, identical binary formatting yields different PSDs because the underlying data statistics differ.

D. Throttle enforces a target processing/sample rate in pure-software flows. Without a hardware clock it prevents the flowgraph from running at maximum CPU speed; it paces the stream so CPU use is bounded and real-time-like behavior is approximated for observation and debugging.

E. Keeping the signal unipolar introduces a nonzero mean, so the PSD will show a pronounced DC (zero-frequency) component and higher relative energy at low frequencies. Converting to bipolar removes that offset and shifts energy away from DC, yielding a flatter low-frequency response.

F. The ideal concept of white noise has uniform power at all frequencies (infinite bandwidth). In GNU Radio you observe a roughly flat PSD inside the simulator's frequency window;

beyond that window the simulator cannot show spectrum, so the white-noise behavior is seen within the system's finite frequency range and matches theory only within that limit.

G. Theoretically yes, because discontinuities introduce arbitrarily high-frequency components. Practically, the measured PSD in GNU Radio is band-limited by sampling rate, filters, and FFT resolution, so the spectrum appears finite — high-frequency content is attenuated or not represented due to system limits.

H. Use the symbol rate $f_b = f_s/Sps$. Lobes occur at multiples of f_b; the number visible up to Nyquist ($f_s/2$) is approximately $N \approx \frac{f_s/2}{f_b} + 1 \approx \frac{Sps}{2} + 1$, where the extra 1 counts the central main lobe.

I. The sampling frequency is $f_s = R_b * Sps$.The spectrum observed extends up to the sample-rate limits (practically analyzed up to Nyquist at $f_s/2$), so the total simulated span is determined by that product.

J. Spectral resolution (the DFT bin width) equals $\Delta f = f_s/N$. A larger FFT size N produces finer frequency bins for a given sampling frequency.

K. Unpacking with K=16 expands each packed word into 16 separate bit samples. This increases the temporal granularity of the bit stream and typically reveals additional spectral lines or harmonics in the PSD because more bit-level transitions become explicit.

L. Estimate the symbol spacing f_b from lobes (lobes ≈ multiples of f_b): $f_b \approx \frac{BW}{N_{lobes}}$. Then the sampling frequency before unpacking is $f_s \approx f_b * Sps$. Alternatively, relate lobes and bit rate to derive $f_s \approx N_{lobes} * R_b$ when appropriate.

M. The output rate scales by the unpacking factor: $f_{out} = f_{in} * K$,
because each input sample is expanded into K output bit samples.

N. Data-type conversion does not change timing; the sample rate remains the same. The block converts bytes to floats but preserves the sample timing, so $f_{out} = f_{in}$.

O. When the symbol is heavily oversampled (large Sps), the discrete-time PSD becomes smoother and more uniform. In practice, Sps values on the order of several units (commonly ≳ 8) make the PSD appear noise-like across a broad band.

P. To obtain a sawtooth wave signal, the array in the variable h was replaced with a list created using the np.arrange function. Additionally, the value of the constant source block was set to 0, and the input was left with the random block.

Q. To achieve Unipolar RZ line coding, the size of the vector h was increased to take a completely square shape. Additionally, the constant source block was set to a value of 0 so that the signal appeared as expected.

R. For Manchester coding, the vector h was manipulated with the corresponding encoding so that the random binary signal passed through the filter and behaved as follows: each bit 1 corresponds to a transition from low to high, and each bit 0 corresponds to a transition from high to low.

S. To achieve an output signal that meets the expected characteristics of OOK modulation, the array in h was replaced with a sinusoidal signal concatenated with a constant value of 0. The np.concatenate and np.arrange functions were used for this purpose.

T. To obtain BPSK modulation at the output, the array in the variable h was replaced with two concatenated sinusoidal functions—one with negative amplitude and the other with positive amplitude—to achieve the phase shift characteristic of this modulation. Additionally, to efficiently analyze the result, the input was changed to an infinitely repeating vector alternating between 0 and 1.

U. The output signal took the form of ASK modulation by replacing h with a sinusoidal function. Additionally, to make the results easier to analyze, the random input was replaced with a predefined vector.

V. The output signal took the shape of a heartbeat by replacing h with a sinc(x) function. To make it resemble the expected shape, adjusting the step values was crucial. The random input was replaced with a vector alternating between 0 and 1 to obtain a more natural response.

W: To obtain the rippled pulses, the NumPy function np.concatenate was used, adding a negatively scaled decaying exponential, followed by an array of ones, and another positively scaled decaying exponential. The random input was replaced with a predefined vector, resulting in a graph very similar to the desired one.

X. A unipolar signal has a nonzero mean which appears as a strong DC component in the PSD; energy is concentrated at low frequencies. A bipolar signal (zero mean) lacks that DC peak and spreads power more across nonzero frequencies, producing a spectrum with less energy at DC and improved spectral utilization.

V. *CONCLUSIONS*

- The use of PSD turned out to be fundamental for the analysis, since it allowed observing how the energy of the signal was distributed throughout the frequency spectrum. Through this representation, it was possible to detect the influence of noise and evaluate the most relevant spectral components, facilitating a deeper understanding of the transmission process.
- The distinction between bipolar and unipolar signals proved relevant in the practice. Bipolar signals, by maintaining an average value close to zero, generated spectra with better energy distribution and without a strong DC component, which improved their efficiency. On the other hand, unipolar signals showed a constant offset, producing a DC level, which although less favorable for spectral efficiency, was useful in particular cases where phase stability was required.
- For the construction and analysis of the signals, it was essential to correctly configure parameters such as sampling rate, samples per symbol, and interpolation. These elements directly influenced the quality of the generated signals in GNU Radio, ensuring that the temporal and spectral representations were consistent with the expected theoretical behavior.

VI. *REFERENCIAS*

[1] H. Ortega y O. Reyes, «Comunicaciones Digitales basadas en radio definidas por sofware», Google Docs. Accedido: 29 de agosto de 2025. [En línea]. Disponible en: https://drive.google.com/file/d/1fd9M4_bIjwLOajQdkdN9ex2plYRoXomz/view?usp=drive_link&usp=embed_facebook