Behavioral Model Comparison of two GFSK Demodulator Topologies for BLE

Arthur Cruz Morbach
Polytechnic School
Unisinos University
Sao Leopoldo, Brazil
arthurmorbach@edu.unisinos.br

Jonas Dandanel de Castro
Polytechnic School
Unisinos University
Sao Leopoldo, Brazil
jdcastro@edu.unisinos.br

Sandro Binsfeld Ferreira
Polytechnic School / itt-Chip
Unisinos University
Sao Leopoldo, Brazil
sbinsfeld@unisinos.br

***Abstract*—In this work, a low intermediary frequency (low-IF) Gaussian Shift Keying (GFSK) modulator and two different architectures of demodulators are implemented in Python. The paper discusses the main differences between the discrete-time quadricorrelator and the FM-discriminator. An algorithm based on the derivative of a Gaussian filter impulse response is used to detect the symbols. The simulations show that, the FM-Discriminator has a simpler implementation and presents a better perform than the quadricorralator, achieving a bit error rate (BER) of 0.1 % at the signal-to-noise ratio (SNR) of 15 dB, while the quadricorrelator achieves the same BER at 18.75 dB.**

***Index Terms*—GFSK, Demodulator, BLE, Python, FM-Discriminator, Quadricorrelator.**

I. Introduction

Bluetooth is a well-known technology when it comes to wireless communication. The extension of its standard version is called Bluetooth Low Energy (BLE) and it allows ultra-low power implementations aiming at IoT.

The bit error rate (BER) used as a reference in BLE applications is 0.1 % [1], and to achieve this rate with lower Signal-to-Noise ratio (SNR) is one of the main challenges in demodulator architecture research nowadays. Recent publications show that the required GFSK demodulator SNR typically varies from 14.4 dB to 18.7 dB [2], [3]. The SNR level can be a metric to show how robust the system is and how it will work under adverse conditions.

This work presents the comparison of two demodulator topologies: the quadricorrelator and the FM-Discriminator. Python language was chosen to model the system to analyse design parameters such as number of samples per symbol through the demodulator chain, intermediate frequency, number of quantization bits, filters order, etc. With the wide availability of libraries for digital communication and good documentation, Python becomes a suitable tool for the architecture design. Scikit Commpy [4] and the Fast Fourrier Transform (FFT) from Scipy [5] were the main libraries used in the implementation, besides the Toolbox [6] previously developed by the group.

The paper is structured as follows. Section II presents the main characteristics of the Gaussian Shift Keying modulation adopted in the BLE implementation. Section III introduces the architecture models and the simulation methodology for a modulator and two architectures for the demodulation process. The results in section IV present the implementation of the system in Python and the analysis of both tested architectures for the demodulator.

II. GFSK Modulation Characteristics

For a GFSK modulation, the symbols are shaped as a Gaussian distribution. The upsampled symbols pass through a Gaussian filter with an impulse response defined as (1).

$$h_{gauss}(t) = K\frac{e^{-\frac{t^2}{2\sigma^2T^2}}}{\sqrt{2\pi}\sigma T}, \tag{1}$$

where K is the filter gain, T is the symbol duration, and the standard deviation σ is defined by (2), also depending on the bandwidth B.

$$\sigma = \frac{\sqrt{\ln 2}}{2\pi BT} \tag{2}$$

Fig. 1 shows the impulse response of a Gaussian filter in the time domain. It's possible to verify that the shape width decreases when the product BT is increased. As specified for BLE [1], BT should be equal to 0.5.

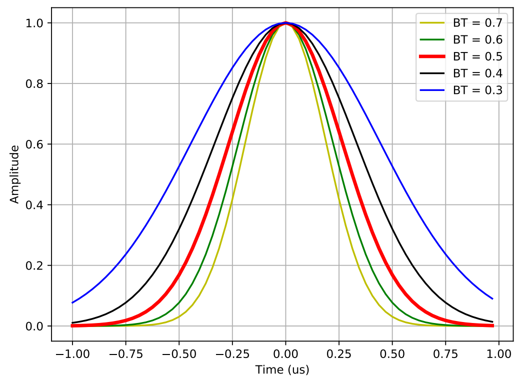

Fig. 1. Analysis of the parameter BT in the Gaussian Filter.

The main reason for the adoption of a Gaussian pulse shape is the spectral efficiency. It can be illustrated when we look at the frequency spectrum at an intermediate frequency (IF) offset. In Fig. 2a, the frequency spectrum of an IF signal with square-shaped symbols is presented. Fig. 2b shows that a much higher spectral efficiency is achieved using the Gaussian-shaped symbols. Hence, the influence on the adjacent channels is reduced.

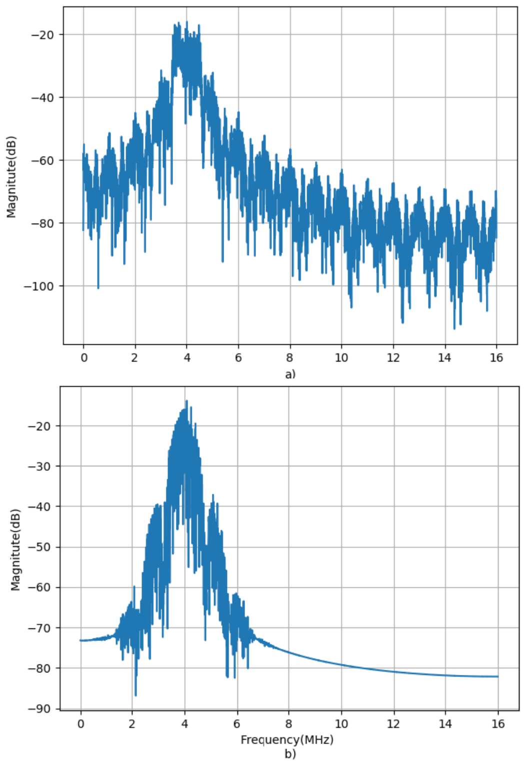

Fig. 2. a) IF modulated signal using a square filter. b) IF modulated signal using a Gaussian filter.

III. GFSK Modulator and Demodulator Models

GFSK modulator and demodulator models were developed in Python for intermediate frequency. As any binary frequency shift keying modulation process, each bit value corresponds to a frequency deviation. Deviation is positive when the bit is '1', and negative, when the bit is '0' [1].

A. Modulator

Fig. 3 presents the block diagram of the modulator architecture. The input data is mapped into the symbols: bit '1' $\rightarrow$ symbol 1, and bit '0' $\rightarrow$ symbol -1. The mapped data array goes to an up-sampler block that insert zeros between each symbol, preparing the signal to be convoluted with the Gaussian Filter impulse response (1).

In BLE, the modulation index must be between 0.45 and 0.55 [1]. This index defines the increment ($incr$) that corresponds to the value of the frequency deviation for a corresponding carrier frequency. The instantaneous phase is equal to the integral of the instantaneous frequency in the time domain according to (3), and the symbols are converted to an array of instantaneous phases.

$$\theta_i(t) = \int_{-\infty}^{t} \omega_i(t)dt \tag{3}$$

The phase output of the integrator is applied in parallel to a $cosine$, in the I arm, and to a $sine$, in the Q arm of the modulator. Both signals are mixed to phase and quadrature clocks at the carrier frequency f_c, respectively (4) and (5).

$$I(t) = cos(\theta_{i(t)})cos(2\pi f_c t) \tag{4}$$

$$Q(t) = sin(\theta_{i(t)})sin(2\pi f_c t) \tag{5}$$

By trigonometric relations, the difference between (5) and (4) is the modulated signal, $s(t)$, (6).

$$s(t) = cos(2\pi f_c t + \theta_{i(t)}) \tag{6}$$

B. Demodulator - Quadricorrelator

The first demodulator implemented topology is based on a discrete-time quadricorrelator [7], Fig. 4. The input signal (6) is parallel mixed by a cosine and a sine generated by the local oscillator. The result of the mixing process is filtered by a low pass filter (LPF), resulting in the corresponding I and Q signals (7) and (8).

$$I_{demod_{filt}}(t) = \frac{1}{2}cos(\theta_{i(t)}) \tag{7}$$

$$Q_{demod_{filt}}(t) = \frac{1}{2}sin(\theta_{i(t)}) \tag{8}$$

Based on (3), the derivative of the instantaneous phase can be understood as the instantaneous frequency, i.e., the transmitted symbols.

To acquire the symbols, the I and the Q components are mixed by the first derivative of Q and I respectively, resulting in (9) and (10).

$$I_{qc_{out}}(t) = \frac{\omega_i(t)}{8}(1 + cos(2\theta_{i(t)})) \tag{9}$$

$$Q_{qc_{out}}(t) = -\frac{\omega_i(t)}{8}(1 - cos(2\theta_{i(t)})) \tag{10}$$

The transmitted symbols, given by (11), are directly obtained by subtracting (10) from (9). For a discrete-time quadricorrelator implementation, the differentiators are replaced by delay elements [7].

$$I_{qc_{out}}(t) - Q_{qc_{out}}(t) = \frac{\omega_i(t)}{4} \tag{11}$$

Based on the concept of a matched filter, the derivative form of the Gaussian filter impulse response used at the modulator, Fig. 5, is implemented to determine when is the best time to sample the symbols.

Convolving the signal with the derivative form of the Gaussian filter impulse response results in a signal that crosses zero each time the symbol's amplitude has the maximum value, defining the best time for sampling it. This information feeds the downsampler and the demapper block generates the bits relative to the sampled symbols.

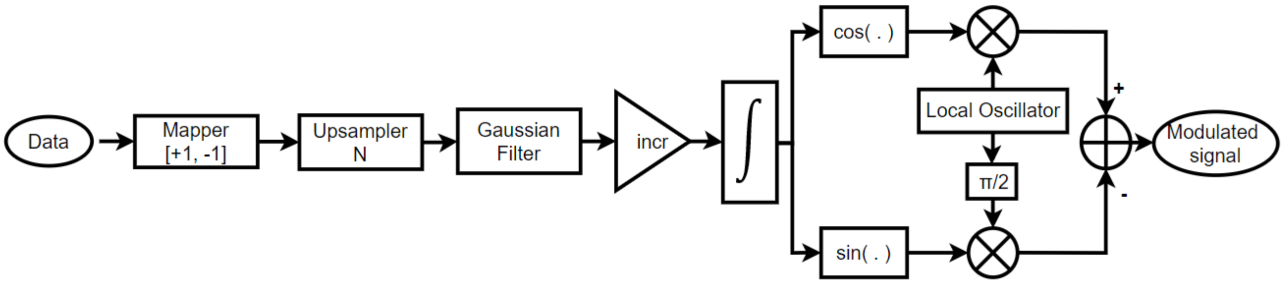

Fig. 3. GFSK modulator - block diagram.

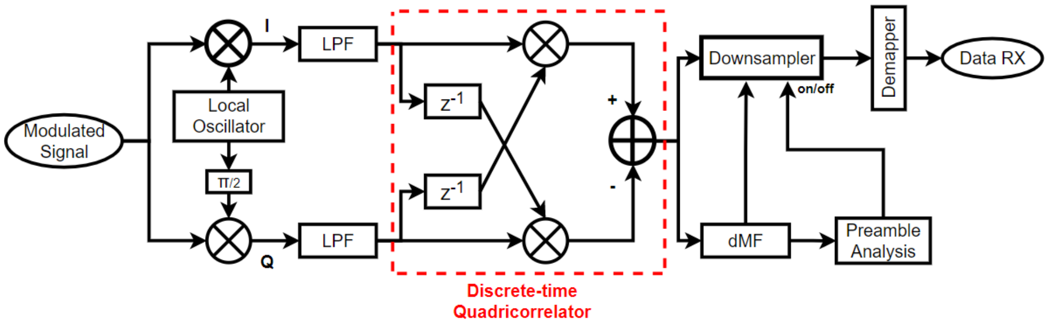

Fig. 4. GFSK demodulator based on a discrete-time quadricorrelator.

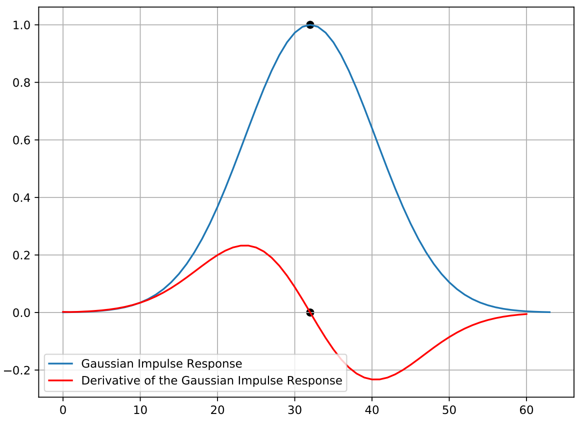

Fig. 5. Gaussian filter impulse response in blue and its derivative form in red.

C. Demodulator - FM Discriminator

The FM Discriminator method [8], also known as FM-to-AM conversion, allows the architecture to be very simple and direct. Instead of mixers in quadrature like the previous method, the input signal is mixed by its own delayed form. The block diagram is presented in Fig. 6.

The output of the mixer can be described as (12). After expansion, trigonometric simplification,and filtering by a LPF, the signal (13) is obtained [8].

$$V_{disc_{out}}(t) = cos(2\pi f_c t + \theta_{(t)}) cos(2\pi f_c t + \theta_{(t)} + \phi(\tau)) \quad (12)$$

$$LPF(V_{disc_{out}}(t)) = \frac{1}{2} cos(\phi(\tau)) \quad (13)$$

where $\phi(\tau)$ is the phase shift caused by the time delay. The delay is chosen in a way that it will cause a phase shift of $\pi/2$ at f_c. If there is a positive deviation from the carrier, the phase shift will be higher than than $\pi/2$, and the output value will be negative. If a negative deviation from the carrier is detected, the phase shift will be smaller than $\pi/2$, making the output value positive [9].

The derivative of the Gaussian filter impulse response is also used in this topology, feeding the preamble analysis block that searches for the BLE preamble sequence and turns on the downsampler.

IV. Results

The communication system was implemented in Python to evaluate and compare the two GFSK demodulator architectures.

At the modulator, a carrier frequency of 4 MHz was used, the symbol period of $1\mu s$ and a 500 kHz bandwidth produce a BT of 0.5. The frequency deviation is 250 kHz, therefore the modulation index is 0.5 [9]. The maximum packet size allowed for BLE was used for the simulations, 2120 bits [1].

The algorithm responsible for the preamble analysis had a satisfactory result, recognizing the 10101010 or 01010101

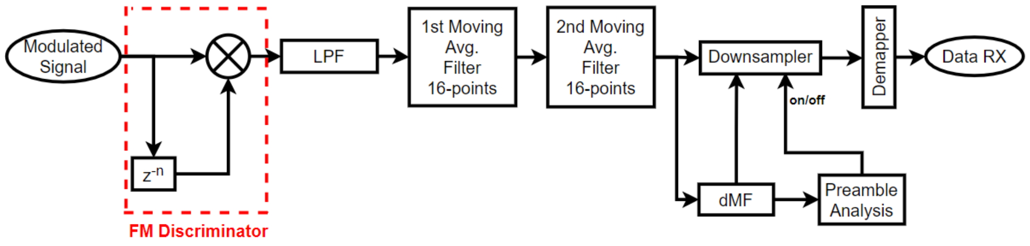

Fig. 6. GFSK demodulator based on an FM discriminator.

BLE sequence [1]. Fig. 7 shows that the signal filtered by the derivative of the Gaussian filter can detect the peak of each symbol from the preamble, feeding the preamble analysis and downsampler in both architectures.

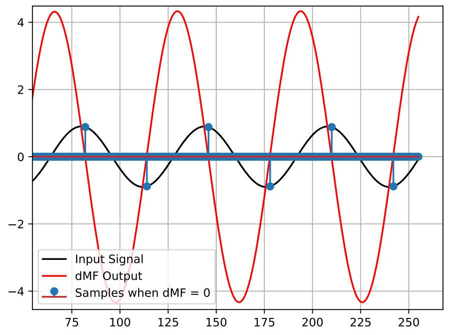

Fig. 7. Input signal in black, output of the derivative of the Gaussian filter in red and timing detection in blue.

The simulations show that the FM-discriminator has a better performance than the quadricorrelator and a less complex implementation.

Fig. 8 shows the BER for each model according to a certain SNR. The FM discriminator achieves 0.1 % BER at a 15 dB SNR level, while the quadricorrelator achieves that same BER at 18.75 dB.

V. Conclusions

A GFSK communication system was implemented in Python to analyze two architectures of demodulators. The simulations were performed using 4-MHz low-IF. The FM-discriminator showed a better performance despite its simpler architecture, achieving a BER of 0.1% at 15 dB SNR, while the quadricorrelator achieves the same BER at 18.75 dB. For future works, the group wants to focus on the VHDL implementation and make the full project documentation available.

VI. Acknowledgment

The Authors would like to thank FAPERGS and Unisinos University.

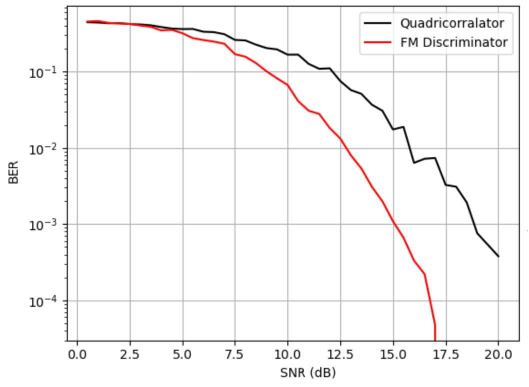

Fig. 8. BER vs SNR curve for the Quadricorrelator in black and for the FM-Discriminator in red.

References

[1] Specification of the Bluetooth System v4.2. [Online]. Available: https://www.bluetooth.com/specifications/.

[2] M. S. Pereira, J. C. Vaz, C. A. Leme, J. T. de Sousa and J. C. Freire, ”An ultra-low power low-IF GFSK demodulator for Bluetooth-LE applications,” 2015 IEEE International Symposium on Circuits and Systems (ISCAS), 2015, pp. 1226-1229, doi: 10.1109/ISCAS.2015.7168861.

[3] S. Hong, A. K. George, D. Im, M. Je and J. Lee, ”A 1.0 V, 5.4 pJ/bit GFSK Demodulator Based on an Injection Locked Ring Oscillator for Low-IF Receivers,” in IEEE Access, vol. 8, pp. 185209-185217, 2020, doi: 10.1109/ACCESS.2020.3029863.

[4] PyPi scikit-commpy , “Digital Communication Algorithms with Python,” https://pypi.org/project/scikit-commpy/ (accessed Jun. 20, 2020).

[5] PyPI Scipy, “SciPy: Scientific Library for Python,” https://pypi.org/project/scipy/ (accessed Jun. 20, 2020).

[6] A. C. Morbach, J. D. Castro, S. B. Ferreira, “16 QAM Communication Toolbox in Python,” https://16qam-system.readthedocs.io/en/latest/index.html (accessed Mar. 20, 2021).

[7] D. Chang and T. Shiu, ”Digital GFSK Carrier Synchronization,” APCCAS 2006 - 2006 IEEE Asia Pacific Conference on Circuits and Systems, 2006, pp. 1523-1526, doi: 10.1109/APCCAS.2006.342513.

[8] M. Silva Pereira, J. Caldinhas Vaz, C. Azeredo Leme, J. T. de Sousa and J. Costa Freire, ”A 170 μA All-Digital GFSK Demodulator With Rejection of Low SNR Packets for Bluetooth-LE,” in IEEE Microwave and Wireless Components Letters, vol. 26, no. 6, pp. 452-454, June 2016, doi: 10.1109/LMWC.2016.2562639.

[9] R. Schiphorst, F. Hoeksema and K. Slump, Bluetooth demodulation algorithms and their performance.