

# Implementación y ensayo de un demodulador FSK

## Implementation and testing of a FSK demodulator

Leandro J. Ferrari<sup>1</sup>, Sergio D. Leoni<sup>2</sup>, Christian L. Galasso<sup>3</sup>, Martín E. Paz<sup>4</sup>

<sup>1,2,3,4</sup>Grupo SITIC – Dpto. de Ing. Electrónica - Facultad Regional Bahía Blanca - Universidad Tecnológica Nacional - Bahía Blanca - Argentina

<sup>3,4</sup>Servicio de Análisis Operativos, Armas, y Guerra Electrónica - Armada Argentina - Puerto Belgrano –Argentina.

<sup>3</sup>Escuela de Oficiales de la Armada – Universidad de la Defensa Nacional - Puerto Belgrano -Argentina.

<sup>3</sup>clgalasso@frbb.utn.edu.ar

Recibido: 16/10/20; Aceptado: 18/12/20

**Resumen—** En el presente trabajo se describe la implementación y ensayo de un demodulador FSK, en el marco de Re-Ingeniería de un modem para comunicaciones navales. El mismo de desarrolló sobre una FPGA Spartan 6 XCSLX25 con su código en VHDL. Para las pruebas se realizaron las modulaciones en un microcontrolador STM32F407 donde se agregaron distintos niveles de ruido para poner a prueba el sistema.

Palabras clave: FSK; Demodulación; Cuadri-correlacionador balanceado; Ensayo; SNR; AWGN; BER; FPGA; VHDL.

**Abstract—** This work describes the implementation and testing of a FSK demodulator, within the framework of a modem re-engineering for navy communications. It was developed over a Spartan 6 XCSLX25 FPGA with its code in VHDL. For testing, a modulator was implemented in a STM32F407 microcontroller where different noise levels were added to the signal to try out the system.

Keywords: FSK; Demodulation; Balanced Cuadri-correlator; Test; SNR; AWGN; BER; FPGA; VHDL.

### I. INTRODUCCIÓN

Dado el requerimiento de desarrollar un módem que trabaje con señales de voz en banda base y que sea completamente compatible con el existente (manteniendo protocolos y modulaciones), pero cuyo hardware sea lo suficientemente flexible para implementar a posteriori nuevas modulaciones y protocolos [1] para comunicaciones en canales de HF, VHF y UHF, se aborda la re-ingeniería del modem actual.

La necesidad de mantener compatibilidad hacia atrás radica en el hecho de que los prototipos que se fabriquen deberán convivir durante un tiempo considerable con el módem actualmente en uso, permitiéndose de esta manera una modernización paulatina de las unidades y la maduración gradual del desarrollo.

De todas las modulaciones que coexisten en el equipo actual, se comenzó por estudiar Binary Frequency Shift Keying (BFSK). Desde el punto de vista matemático, existen

muchos tipos de sistemas que permiten demodular una señal BFSK [2]. En este trabajo se abordó un demodulador que pudo ser sintetizado en un núcleo de FPGA, donde se buscó tener un buen performance para señales en las que se tienen pocos ciclos por símbolo y las frecuencias son cercanas entre sí.

En un intento previo, donde se implementó el modulador y demodulador íntegramente en el microcontrolador STM32F407vg [3], el espacio en memoria ocupado por los moduladores y demoduladores más el espacio ocupado por los protocolos de comunicación de modem a modem (AX 25) [4] y de modem a computadora (KISS) [5], dejó muy poco lugar para futuras implementaciones.

Por otra parte, resulta de interés para el grupo de trabajo incursionar en el procesamiento digital de señales sobre FPGA, y es por ello que se optó por este camino de diseño de un hardware mixto, compuesto por un microcontrolador, que realice las conversiones de las señales (AD y DA) y maneje los protocolos de comunicación, junto con una FPGA que se encargue de implementar los distintos moduladores y demoduladores.

El kit de desarrollo 3PX1 (Fig. 1), posee una FPGA Spartan 6 XC6SLX25 [6] de Xilinx, el mismo posee un cristal de 50MHZ, 2 memorias FLASH de 128Mbit, una memoria EEPROM de 2kbits y la interfaz necesaria para la configuración por USB.

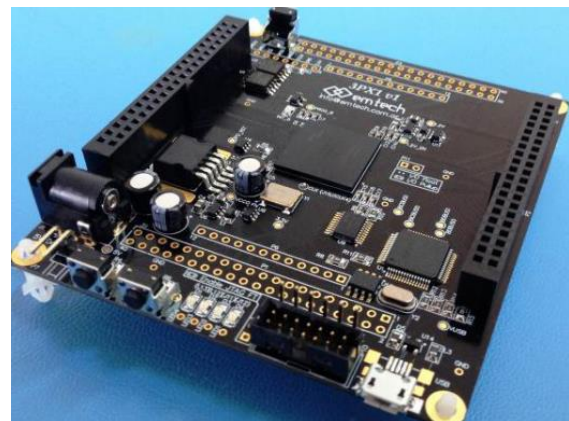


Fig. 1. Kit de desarrollo 3PX1.

Para el microcontrolador se eligió un kit de desarrollo (Fig. 2) que integra un microcontrolador STM32F407vg de 168 MHz, un núcleo ARM Cortex-M4F, 192 KB de SRAM y 1 MB de Flash.

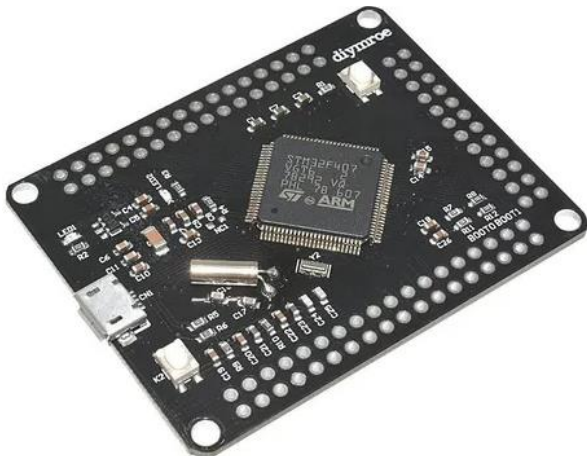


Fig. 2. Kit de desarrollo STM32F407vg.

Para la interconexión entre ambas placas de desarrollo, se diseñó una placa de expansión o “poncho” (Fig. 3), donde además de realizar las conexiones, provee al sistema con la capacidad de expandir el diseño agregando módulos que completen al prototipo del módem.

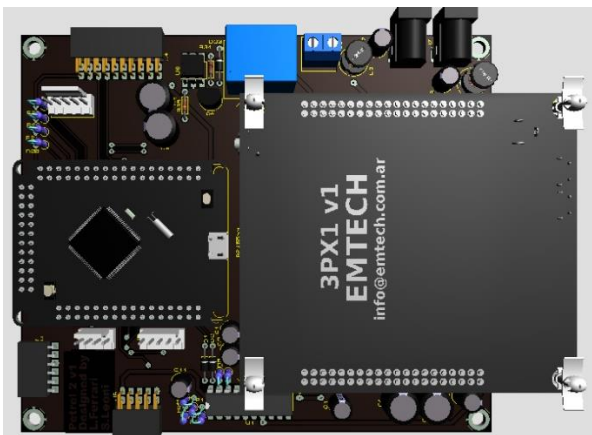


Fig. 3. Placa de desarrollo prototipo v1 modem.

## II. FUNDAMENTOS

### A. Estado del arte

Los dispositivos lógicos programables en campo (FPGA), desde su invención en la década del 80' han ido sumando prestaciones y encontrando nuevos lugares para su implementación. Entre las capacidades que han adquirido, están los módulos de multiplicación y acumulación (MAC) para poder realizar operaciones matemáticas aplicadas al procesamiento digital de señales. Estas capacidades de procesamiento han ido incrementándose de la mano de un avance en el desarrollo de la matemática discreta. Es por ello que hoy en día se pueden hacer una gran variedad de operaciones sobre señales, como: modulaciones, CRC's, codificación para la detección/corrección de errores e implementación de distintos protocolos.

Por su parte los Microcontroladores modernos incorporan una gran variedad de periféricos, entre ellos algunos que aplican muy bien para el procesamiento de señales como

son los conversores analógicos digitales, los conversores digitales analógicos y los canales de DMA (Del inglés: Direct Memory Access) que permiten pasar información de periféricos a memoria y de memoria a puertos GPIO (Entradas/Salidas de propósito general) rápidamente y sin intervención del núcleo.

Con el presente proyecto se buscó desarrollar un hardware, que permitiera implementar y ensayar, primeramente, en laboratorio y luego en campo, distintos protocolos y modulaciones. Se comenzó por implementar protocolos y modulaciones actualmente en uso, para comprobar experimentalmente la efectividad de demoduladores que puedan implementarse íntegramente de manera digital.

### B. Generalidades de las modulaciones FSK a implementar

El módem a reemplazar tiene dos configuraciones distintas de modulación BFSK; estas son BFSK1200 y BFSK300. Estos tipos de modulaciones se caracterizan por tener una señal portadora donde se varía la frecuencia de la misma en  $\pm\Delta f$  para cada símbolo (poseen solo dos símbolos o frecuencias), la frecuencia inferior es el espacio ('0') y la superior la marca ('1'). Para FSK300 se tiene una portadora de  $2200\text{Hz} \pm 100\text{Hz}$  y para FSK1200 la portadora es  $1700\text{Hz} \pm 500\text{Hz}$ .

El esquema de FSK1200, presentan un desafío a la hora de la demodulación, ya que la tasa de bits (1200 bps) es igual a la frecuencia del espacio ('0'), por lo que, para su detección, es necesario poder demodular en un solo periodo.

Para el caso de FSK300 (300 bps), se encuentra el problema de que posee un ancho de banda extremadamente angosto.

### C. Elección del demodulador a ensayar

Al elegir un demodulador, se busca que este posea un elevado rechazo al ruido, que no se necesite un bloque para la sincronización de cada símbolo, y a su vez que tenga una tasa baja de error. Las técnicas de demodulación más utilizadas son “PLL” [7], “Cuadri-correlacionador balanceado” [8], “Algoritmo de Goertzel” [9].

Basados en la necesidad de implementar un demodulador que cumpla con los requerimientos expuestos, y que sea sintetizable para su aplicación en FPGAs, se optó por una variante del “Cuadri-correlacionador balanceado”, cuyo diagrama se muestra en la Fig. 4.

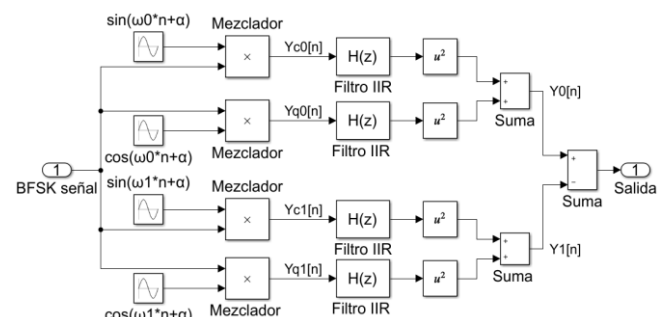


Fig. 4. Detector a implementar.

### D. Principio de funcionamiento

Los demoduladores cuadri-correlacionados poseen osciladores locales en cuadratura, sintonizados a las

frecuencias esperadas de entrada, es decir, del espacio y de la marca. Dicha cuadratura es necesaria para poder recuperar la energía de la señal, que se encuentra desfasada respecto a los osciladores.

Para poder observar el comportamiento de los correlacionadores, se generó una señal BFSK que se observa en la Fig. 5.a, donde desde el tiempo de muestra 0 hasta 100 corresponde a la frecuencia de espacio, y del tiempo de muestra 100 en adelante, a la frecuencia de marca.

El producto de dos sinusoides, si están correlacionadas, genera otra señal sinusoidal montada sobre una continua, caso contrario, dicho valor de continua desaparece, este efecto es visible en la Fig. 5.b. Para recuperarlo, se utiliza un filtro IIR que elimina las altas frecuencias, fenómeno que puede observarse en la Fig. 5.c.

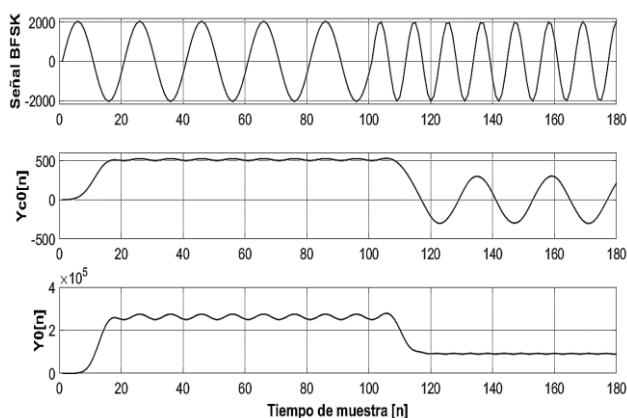


Fig. 5. Salida de correlacionador.

La rama superior del detector de la Fig. 4, se utiliza para poder recuperar todos los símbolos de espacio o '0', y la rama inferior es la encargada de detectar todos los símbolos del tipo marca o '1'. Ambas ramas son necesarias para poder detectar los unos y los ceros, así como también la ausencia de señal en la entrada del demodulador.

### E. Fundamentación matemática

La salida de cualquier correlacionador es la multiplicación en el dominio del tiempo discreto, de la señal de entrada con la señal de referencia en la entrada del mismo. Cabe destacar que existe un desfase entre la señal de entrada y los osciladores, los cuales se representan en la ec. (1) el cual es la salida del oscilador a en cuadratura.

$$Y_{qx}[n] = A \cos(\omega_i n + \phi_i) \cos(\omega_L n + \phi_L) \quad (1)$$

Los desfases presentados en la ec. (1), son producto de la falta de sincronización entre la señal de entrada y los osciladores locales. Aplicando identidades trigonométricas a la expresión anterior, tenemos el resultado de la ec. (2).

$$y_{qx}[n] = \frac{A}{2} \{ \cos[n(\omega_i - \omega_L) + \phi_i - \phi_L] + \cos[n(\omega_i + \omega_L) + \phi_i + \phi_L] \} \quad (2)$$

Para el caso donde la frecuencia recibida sea la menor (frecuencia de espacio)  $(\omega_0, \phi_0)$  y considerando que  $\omega_L = \omega_0$  se obtiene la ec. (3).

$$y_{q0}[n] = \frac{A}{2} [ \cos(\phi_i - \phi_L) + \cos(2n\omega_0 + \phi_i + \phi_L) ] \quad (3)$$

En la ec. (3) se puede observar que el primer término es una constante que depende exclusivamente del desfase entre la señal y los osciladores. El segundo término es una componente del doble de frecuencia. En el caso de que la frecuencia de entrada sea distinta que la del oscilador, la salida serán dos componentes frecuenciales, la suma y la resta.

Finalmente, se pasa por un filtro pasa bajos IIR con una frecuencia de corte de la mitad de la velocidad de transferencia. El resultado a la salida del filtro es una constante, como se puede ver en la Fig. 5.c.

El problema de no recuperar la portadora es que la energía del símbolo se dispersa a causa de la diferencia de fase, esto se soluciona haciendo una correlación con dos osciladores en cuadratura. Aplicando la identidad fundamental de la ec. (4), se puede recuperar la energía de cada símbolo.

$$y_{0[n]} = \frac{A}{2} [ \cos^2(\phi_i - \phi_L) + \sin^2(\phi_i - \phi_L) ] = \frac{A}{2} \quad (4)$$

## III. IMPLEMENTACIÓN

### A. Consideraciones previas de diseño en VHDL

Se impuso como una de las condiciones de diseño, la optimización del uso de recursos en la FPGA, de forma de tener espacio para implementar, a futuro, otras modulaciones.

Cada modulación a implementar, necesitará utilizar uno de los recursos más escasos en estos dispositivos, como son los bloques DSP (multiplicadores y acumuladores). Se pudo optimizar el empleo de dichos bloques, ya que las frecuencias de las señales a demodular son ampliamente inferiores al reloj utilizado. Es por ello que para la descripción de los filtros se optó por evaluarlos en varios ciclos de máquina, utilizando un solo bloque DSP, a evaluarlos en un solo ciclo, utilizando varios bloques.

Otro punto que se tuvo en cuenta fue el tratamiento matemático de los números. Mientras que los reales (puntos flotantes) implican una gran implementación de hardware, el uso de enteros no lo requiere. Por esta razón, a los coeficientes del filtro, los cuales son números de punto flotante, se los escala con un número múltiplo de 2 (ya que dividir por dos es desplazar a la derecha un bit), para transformarlos en enteros.

### B. Diseño bloques en VHDL

Para el diseño del demodulador en VHDL, se describieron diferentes bloques como los que se muestran la Fig. 4.

Como en el sistema coexisten dos relojes, uno propio de la implementó un circuito sincronizador [10] a fin de evitar problemas de meta estabilidad. La nueva señal generada por este se usa para tomar los datos de las muestras que van llegando.

1) *Osciladores locales*: conformados por una tabla de valores que van desde -255 a +255, los cuales representan a los senos y cosenos de las distintas frecuencias (1200Hz, 2200Hz, 2100Hz y 2300Hz). Estos valores son seleccionados mediante un contador que se incrementa a la velocidad de

muestreo del AD (convertor analógico digital). El sistema es similar a un DDS (Síntesis Directa Digital) [11], mecanismo que se muestra en Fig. 6.

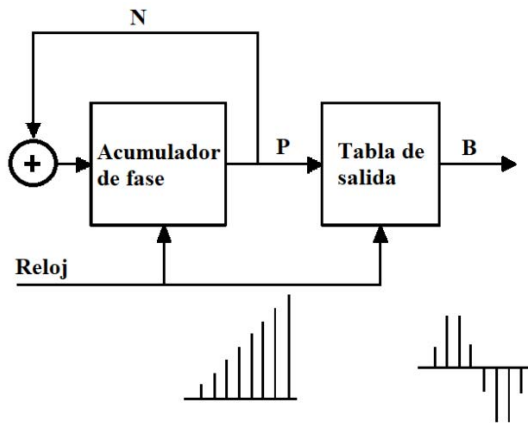


Fig. 6. Diagramas de bloques de DDS.

2) *Mezcladores*: consisten en una multiplicación punto a punto de la señal de entrada, representada por 11 bits más el bit de signo (-2047 a 2047), con la señal entregada por los osciladores, formada por 8 bits más el bit de signo. La multiplicación entre dos números de  $n$  y  $m$  bits, respectivamente, tiene como resultado un número de  $n+m$  bits. En este caso, el resultado se debe almacenar en una señal de 21 bits, eliminando los 8 bits menos significativos, equivaldría a dividir por 256, lo que quitaría la amplitud de los osciladores, quedando una señal de 13 bits. Finalmente, se elimina el bit más significativo, debido a que por la multiplicación el bit del signo se duplicó, y de esta manera se conserva un ancho de 12 bit.

Como la frecuencia de muestreo de la señal (24KHz) es mucho menor a la frecuencia de trabajo del dispositivo lógico (50MHz), se pueden realizar varios cálculos entre muestras. Para este caso, se realizó una máquina de estados que entrega las cuatro multiplicaciones en 6 ciclos de reloj, usando un solo multiplicador.

3) *Filtro IIR*: consiste en un filtro de tipo Butterworth, de orden 2, cuyos coeficientes se escalaron a enteros para una mejor síntesis. Este filtro utiliza un solo multiplicador y finaliza su cálculo en 10 ciclos de reloj, también, al igual que el mezclador se truncó la salida y se conservaron 12 bits más significativos, y de esta forma ahorrar en conexiones interna de la FPGA.

4) *Detector*: el módulo de detección que está formado por las operaciones de cuadrado de la señal BFSK, sumas, restas y recortador, se realizó en un solo bloque, que consiste en la descripción de una máquina de estados donde se resuelve el proceso completo, con un retardo de solo 6 ciclos, logrando ahorrar recursos de hardware.

El diseño se realizó para que, con un simple cambio de registro, se pueda conmutar el demodulador de FSK1200 a FSK300 y viceversa. De esta forma se pueden reutilizar bloques de VHDL con la consecuente reducción en el porcentaje de área utilizada en la FPGA, tal como se puede observar en la TABLA I, los recursos utilizados ocupan un 10%, lo que deja un margen bastante amplio para seguir trabajando.

TABLA I  
RECURSOS UTILIZADOS EN LA FPGA

Recurso	Utilizado	Total	Porcentaje
Registros	1531	30064	5%
LUTs	859	15032	5%
Flip-Flop	2130		
IO	25	186	13%
DSP	6	38	15%

C. Simulación

Para la simulación se generó, con ayuda de una herramienta matemática, una señal modulada en FSK1200 con diferentes niveles de SNR (Relación Señal Ruido), agregando ruido blanco Gaussiano [12]. Dicha señal se cargó en una tabla, dentro del hardware descrito, y se la recorrió con un contador mientras se alimentaba al demodulador, como se puede apreciar en la Fig. 7.

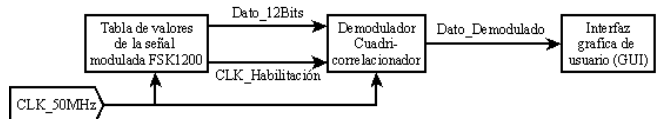


Fig. 7. Diagrama en bloque de simulación.

En la Fig. 8, se presenta un ensayo con una SNR de 40dB, donde podemos identificar a simple vista como responde el sistema ante las variaciones de frecuencias (espacio y marca).

Cabe resaltar que se observa un pequeño desfasaje entre la señal modulada y su respectiva demodulación, este comportamiento es normal por los retardos que infieren los distintos bloques.

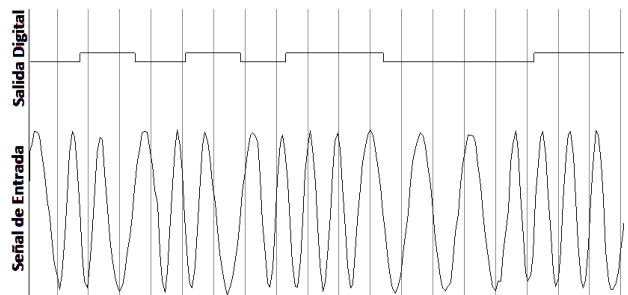


Fig. 8. Arriba señal demodulada, abajo señal FSK1200 40dB SNR.

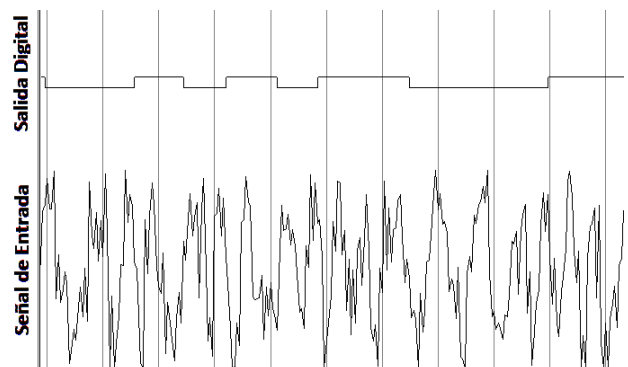


Fig. 9. Arriba señal demodulada, abajo señal FSK1200 5dB SNR.

Para poder poner a prueba al demodulador se ensaya con una SNR más baja, intentando simular los efectos producidos por el canal de comunicación. En la Fig. 9, se ve el



desempeño del demodulador frente a una condición de 5dB de relación señal a ruido.

IV. ENSAYO

Se realizaron los ensayos para calcular la tasa de error de bits (BER) y así tener una idea de cómo se comportará el sistema de demodulación frente a diferentes niveles de SNR y poder acercarnos a un escenario real (deformación de la señal debido a interferencias externas, ionosfera, EMI, etc.). Cabe destacar que, para los ensayos, se tomó un mínimo de SNR de 5dB debido a que el módem a implementar se conecta con transceptores de comunicación de voz y primero se verifica la correcta recepción de la misma, para luego conectar el módem.

A. Contexto del ensayo

La modulación y la generación del ruido se realizaron sobre el microcontrolador, aprovechando la capacidad de analizar el comportamiento en tiempo de ejecución (“Debug”).

Para que la muestra sea representativa, se generaron tramas aleatorias de 20000 bits. La señal entrada a la FPGA es de 12 bits con signo (valores desde -2047 a +2047). Esta entrada es un buffer de 12bit en paralelo. Se utilizó una frecuencia de muestreo de 24KHz.

El ruido Gaussiano Blanco agregado a la señal se generó con números aleatorios normalizados. El microprocesador utilizado, está provisto con módulo generador de números aleatorios reales (RNGs). Los números aleatorios son de suma importancia en algunos sistemas (como generación de certificados de red), por lo que es muy importante que la generación de esto sea, siendo redundante, aleatorio. Los numero aleatorios pueden ser creados mediante un algoritmo, normalmente conocido, implantando una semilla, que suele ser privada. La otra forma es en base a una fuente de ruido (ruido sobre una fuente de tensión continua) donde se generan números aleatorios, este es nuestro caso, y como la fuente es aleatoria los números lo son igual. Para poder generar un ruido del tipo Gaussiano, se deben normalizar los números bajo la curva normalizada. Para esto se usó el algoritmo de Box-Muller [13].

$$X_{i,normal} = \mu + \sigma\sqrt{-2 \ln U_{1,i}} \cos(2\pi U_{2,i}) \quad (5)$$

Donde  $U_{1,i}$  y  $U_{2,i}$  son los i-enésimo números aleatorios y con una media de  $\mu$  y un desvío de  $\sigma$ .

B. Capturas de señales

Para poder comprobar el funcionamiento del sistema, se analizó con un osciloscopio Siglent SDS1072CML, la señal generada (modulación) con la ayuda del conversor analógico digital (DAC) y la respectiva salida digital de la FPGA (demodulación).

Se pone a prueba el sistema con la peor condición de ruido que se impuso en el diseño, 5dB de SNR.

Para poder representar los resultados, se muestran un conjunto de la trama transmitida, con su respectiva trama demodulada, y una vista de la señal de modulación para poder visualizar el ruido en la señal.

Los ensayos se hicieron para ambos demoduladores, para BFSK1200, tenemos la Fig. 10, y para BFSK300, su representación en la Fig. 11.

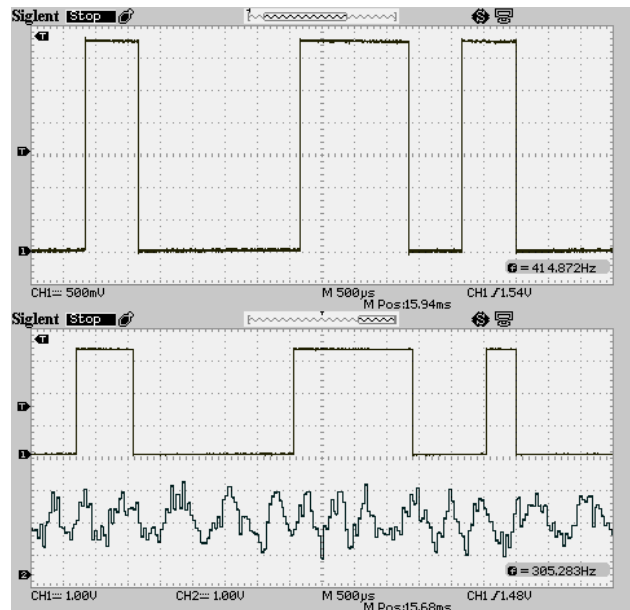


Fig. 10. FSK1200. Arriba. Trama transmitida. Abajo. Trama demodulada con señal modulada 5dB SNR.

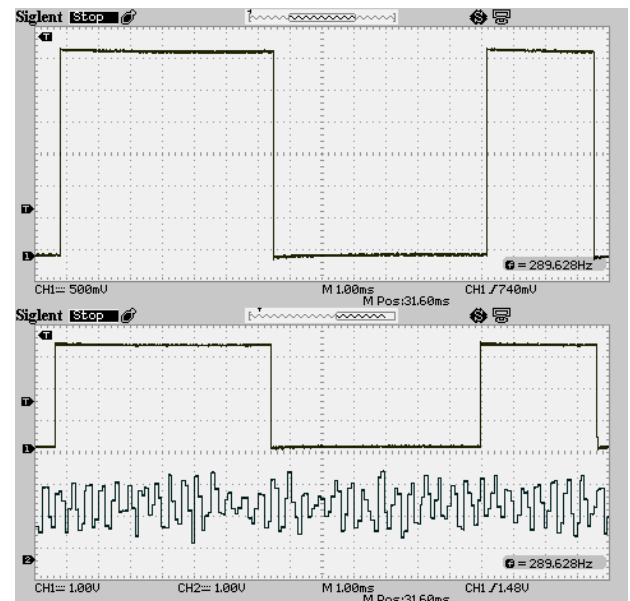


Fig. 11. FSK300. Arriba. Trama transmitida. Abajo. Trama demodulada con señal modulada 5dB SNR.

C. BER

El análisis de BER se basa en la relación de la cantidad de bit incorrectos con respecto a la totalidad de bit transmitidos. Por esta razón es necesario conocer los datos que se transmiten para poder compararlos con los datos recibidos.

Para poder detectar los bits recibidos, se emplea un algoritmo, donde se mide la diferencia de tiempo, entre las transiciones (detección de flancos) de la señal proporcionada por el demodulador, de este modo, sabiendo la tasa de bits, se calcula la cantidad de unos (“1”) o ceros (“0”) contenidos en intervalo medido, y de esta manera se confecciona la trama de datos recibidos.

En la Fig. 12, se muestra un diagrama de flujo para el algoritmo empleado en detección de la trama recibida.

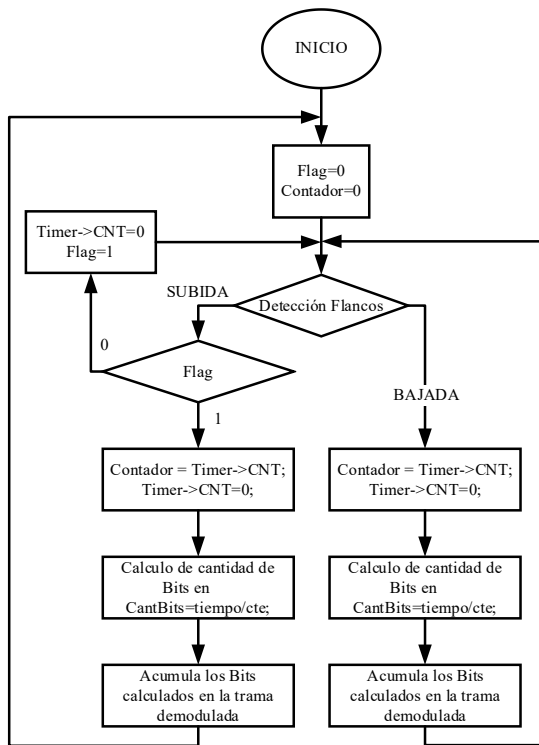


Fig. 12. Diagrama de flujo código detector.

Se pudo obtener los resultados del BER que se muestran en la Fig. 13, donde podemos observar el porcentaje de bits erróneos respecto al nivel de ruido de la señal moduladora.

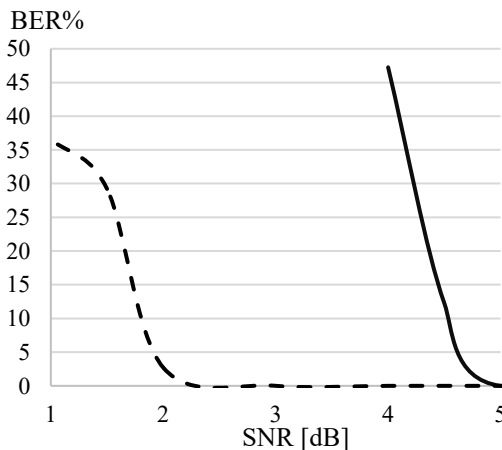


Fig. 13. BER (Línea de Trazo→FSK300 y Línea Continua→ FSK1200)

### V. CONCLUSIONES

Este demodulador es muy simple de desarrollar y se puede observar un buen funcionamiento con modulaciones que poseen pocos ciclos dentro del símbolo. El aspecto más importante es que no se necesita sincronizar con el símbolo binario, por lo que se ahorra en diseño. Otra característica importante es la robustez a la variación de fase que se observó a lo largo de la realización de los ensayos.

En cuanto al filtro utilizado, en un principio se implementó un filtro FIR de orden 20, pero se observó que a bajas frecuencias no respondía correctamente, ya que los

coeficientes calculados se salían de los rangos de los números flotantes usados por los ordenadores. Por esta razón, se cambió por un tipo IIR de bajo orden.

Es importante destacar que para el modo de transmisión de FSK1200, la inmunidad al ruido no es buena, esto es debido a la tasa de transferencia alta y su mínima cantidad de periodos de portadora por símbolo, este modo es usado habitualmente en canales de UHF y VHF. Sin embargo, para el modo de FSK300 muy usado en HF, el demodulador presenta una buena selectividad en cuanto a diferenciar las frecuencias en un estrecho ancho de banda.

Se comprobó una buena inmunidad al ruido para ambos esquemas, sumamente importante para trabajar en canales de HF, que suelen ser problemáticos en los sistemas de comunicaciones.

Finalmente, los resultados obtenidos fueron lo suficientemente satisfactorios para abordar la implementación en la FPGA. Al momento de presentar el trabajo se realizan las primeras pruebas del sistema integrado donde se adapta el audio mediante un circuito ad-hoc, se digitaliza la señal mediante los conversores del STM32F407 para finalmente procesar la señal digitalizada en la FPGA.

### VI. TRABAJO FUTURO

Como trabajo futuro se plantea el diseño y síntesis en FPGA de diferentes esquemas de demodulación para señales BFSK, de manera de poder comparar y mejorar el modem de comunicaciones.

### REFERENCIAS

- [1] *Interoperability and performance standards for data modems*, Department of Defense (DoD) of United States of America Std. MIL-STD-188-110B, 2000.
- [2] J. Torres Gómez, "Técnicas de demodulación para señales BFSK," *Telemática*, vol. 13, n° 3, pp. 35-46, Sep. 2014.
- [3] STMicroelectronics. (2020) STM32F407-417 product webpage. [Online]. Available: <https://www.st.com/en/microcontrollers-microprocessors/stm32f407-417/>
- [4] (1998) The Ax25 website. [Online]. Available: <http://www.ax25.net/>
- [5] M. Chepponis y P. Karn, "The KISS TNC: A simple Host-to-TNC communications protocol," in *ARRL 6th Computer Networking Conference*, 1987, pp. 38-42.
- [6] Xilinx. (2020) Spartan 6 product webpage. [Online]. Available: <https://www.xilinx.com/products/silicon-devices/fpga/spartan-6.html>
- [7] R. F. Campbell, "Analysis of various algorithmic approaches to software-based 1200 baud," M.S. CS thesis, California Polytechnic State University, San Luis Obispo, 2016.
- [8] J. Torres Gómez, F. Hernandez y J. Habermann, "Digital demodulator for BFSK waveform based upon correlator and differentiator systems," *Radioengineering*, vol. 23, n° 4, pp. 1161-1168, Dec. 2014.
- [9] Cheng-Yu Yeh y Shaw-Hwa Hwang, "Efficient detection approach for DTMF signal detection," *Appl. Sci.*, vol. 9, n° 422, Jan. 2019.
- [10] Pong P. Chu, *RTL hardware design using VHDL*, Ed. Hoboken, New Jersey: John Wiley & Sons, Inc., 2006.
- [11] Analog Devices. (2009) Fundamentals of Direct Digital Synthesis. [Online]. Available: <https://www.analog.com/media/en/training-seminars/tutorials/MT-085.pdf>
- [12] R. Howard, "White noise: A time domain basis," in *International Conference on Noise and Fluctuations (ICNF)*, 2015, paper 10.1109, pp. 1-4.
- [13] A. Klöckner, F. van der Linden y D. Zimmer, "Noise generation for continuous system simulation," in *Proceedings of the 10th International Modelica Conference*, 2014, paper 10.3384, pp. 837-846.