

Institut für Hochfrequenztechnik

Universität Stuttgart

María de los Ángeles de la Cruz Barquero

Resumen del proyecto:

“Implementation of a Software Defined Ultra-
Wideband Impulse Radio Receiver”

Studienarbeit

Start Date: 20-12-2008

End Date: 18-08-2008

Supervisor:

Dipl.-Ing. Marcel Blech

En este documento se presenta una breve descripción del proyecto “Implementation of a Software Defined Ultra-Wideband Impulse Radio Receiver” el cual ha sido redactado en inglés.

En términos generales, un sistema de comunicación puede ser considerado como un sistema de Ultra-banda Ancha cuando posee al menos un ancho de banda de 500 MHz. La Comisión Federal de las Telecomunicaciones, encargada de regular el uso del espectro radio, dividió en espectro en pequeñas porciones destinadas a distintos usos. Esta división se convirtió en el enemigo de las señales de Ultra-banda ancha, ya que su uso se prohibió. En el año 2003 se aprobó su uso después de comprobar que podían coexistir con las comunicaciones de banda estrecha y que disponen de muchas ventajas. La tecnología de ultra banda ancha más utilizada hoy en día es la técnica OFDM, la cual está presente en algunos productos como ADSL

El tipo de señales de ultra-banda ancha utilizado en este proyecto es el impulso. La técnica consiste en transmitir pulsos de muy corta duración para enviar la información. Los sistemas que usan este tipo de técnica requieren tiempos de conmutación muy rápidos y son baratos de construir.

El receptor diseñado en esta investigación, ha sido adaptado a los requisitos de un transceptor implementado con anterioridad. El transceptor utiliza tecnología ultra-banda ancha, transmisión por impulsos y modulación B-/QPSK y ha sido implementado utilizando Matlab. Se pretende que el nuevo receptor sea implementado de modo que se el tiempo computacional sea incrementado significativamente y puedan ser simuladas comunicaciones en “tiempo real”. Para incrementar la velocidad, el receptor ha sido implementado usando lenguaje C.

El transceptor está caracterizado por trabajar a una radio frecuencia en el rango de los 2-3 GHz. El tipo de impulso utilizado para transmitir la información tiene una forma Gausiana con un ancho de banda de -10 dB de al menos 500 MHz tal y como es requerido por sistemas de ultra-banda ancha. La frecuencia de símbolo establecida oscila entre $f_{\text{sym}} = 0.5... 62.5$ MHz.

La arquitectura utilizada en el transceptor es la arquitectura de submuestreo, la cual permite eliminar la etapa con mezcladores. De esta forma se eliminan las distorsiones producidas por los mezcladores y sus características no ideales. Un esquema de la arquitectura empleada puede verse en la figura 1.

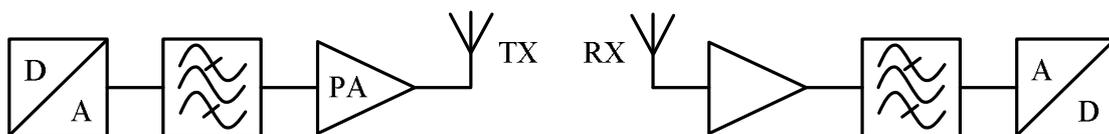


Fig. 1: Arquitectura de submuestreo.

Para conseguir este tipo de arquitectura se han empleado un muestreo paso banda y un convertidor analógico digital.

El muestreo paso banda es una técnica que permite reducir el régimen de muestreo. Esta técnica se consigue mediante dos pasos: primero es necesario filtrar la señal a muestrear mediante un filtro paso banda y segundo, la señal es muestreada con una frecuencia de muestreo dos veces el ancho de banda del filtro paso banda utilizado. De este modo, cuando la totalidad de la señal está localizada en un rango de frecuencias muy alto, es posible disminuir la frecuencia de muestreo y permitir el uso de distintos dispositivos hardware.

Como todas las señales discretas, la señal muestreada tiene un espectro periódico. Si la señal está sintetizada en la primera zona de Nyquist, se generan también una serie de replicas atenuadas en zonas de Nyquist más altas. Teniendo en cuenta este fenómeno, es posible usar un filtro paso banda para directamente conseguir la señal en radio frecuencia deseada. El problema viene en que en la mayoría de las ocasiones las replicas están demasiado atenuadas para poder recuperarlas. Este problema es resuelto utilizando un convertor analógico digital multi-Nyquist. Este tipo de convertor es capaz de operar en distintos modos los cuales ofrecen distintos tipos de amplitudes siempre mayores que las obtenidas con un convertor normal. Esta situación puede observarse en la figura 2.

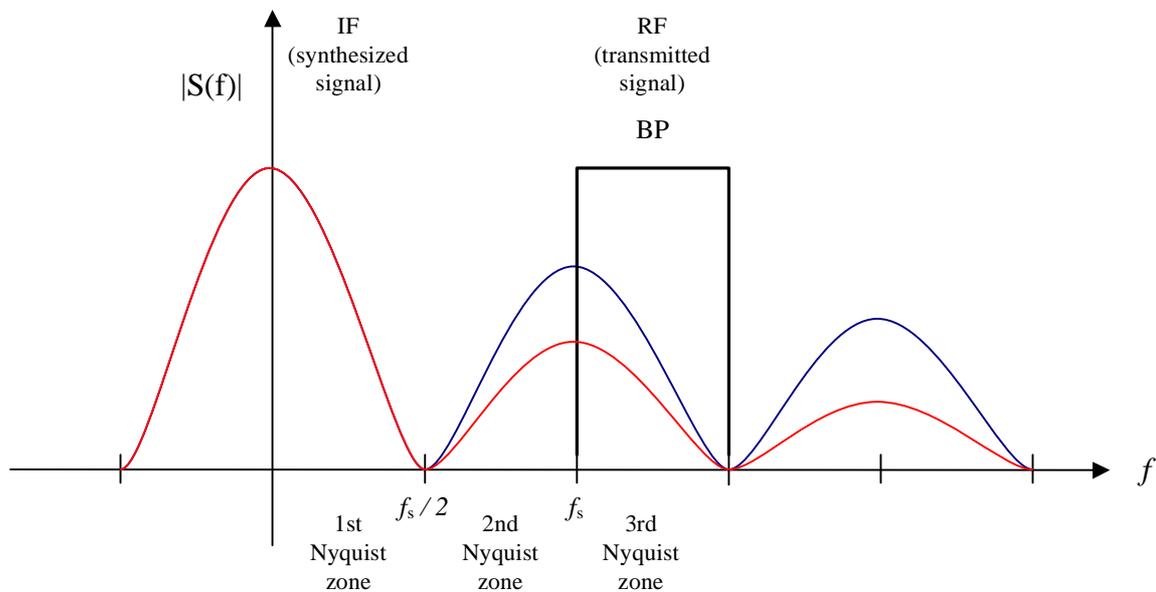


Fig. 2: Proceso de muestreo paso banda y comparación entre las salidas obtenidas por un convertor tradicional (rojo) y un convertor multi-Nyquist (azul).

Otra técnica utilizada es el muestreo de primer orden. Esta solución permite relajar los requerimientos del conversor. Utilizando los métodos anteriormente descritos es posible realizar una implementación hardware evitando circuitos más complejos que introducen distorsiones.

El transmisor está caracterizado por utilizar impulsos base y modulación B-QPSK. El impulso base utilizado para transmitir la información viene dado por

$$g(t) = Ke^{\ln(g_t)\left(\frac{t}{\tau}\right)^2} = Ke^{-\pi(at)^2}, \quad (1)$$

Donde K representa la amplitud de pico, g_t la amplitud mínima y τ el tiempo de descenso. Una representación del impulso en el dominio del tiempo y la frecuencia puede verse en la figura 3.

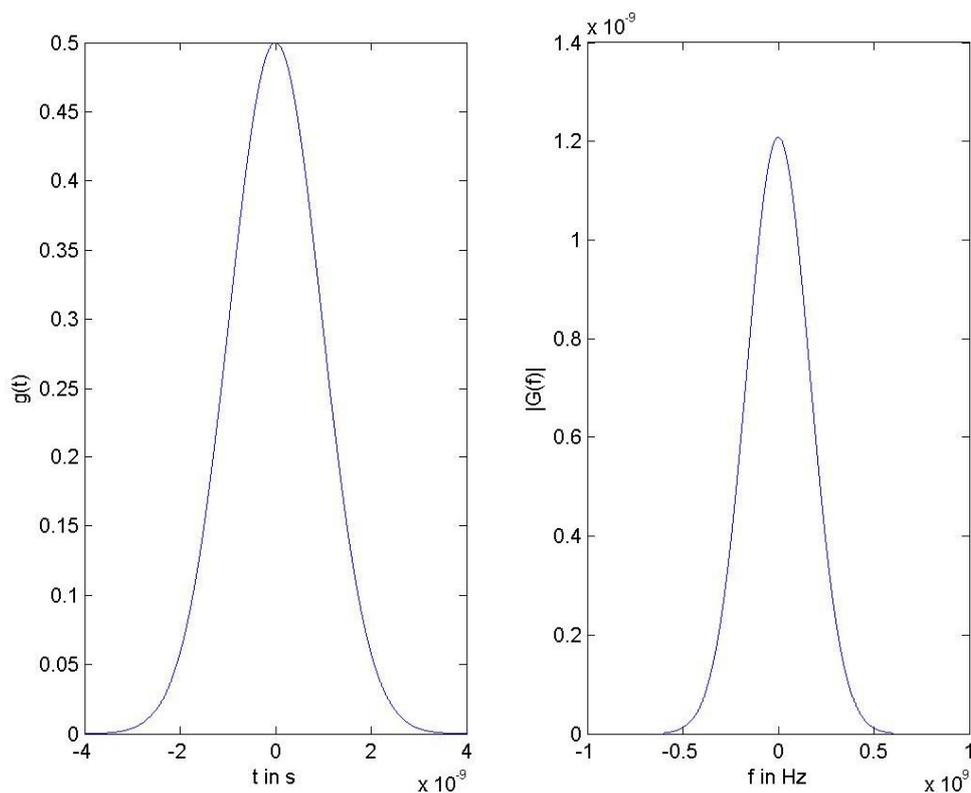


Fig. 3: Impulso base en dominio del tiempo y frecuencia.

El impulso base es modulado utilizando la modulación B-QPSK, habilitando de este modo la transmisión de distintos tipos de símbolos. La forma de onda resultante viene dada por

$$g_{m,PSK}(t) = K\sqrt{2}\cos(2\pi f_0 t + \theta_{PSK} + \varphi)e^{\ln(g_t)\left(\frac{t}{\tau}\right)^2}, \quad (2)$$

donde f_0 representa la frecuencia central, θ_{PSK} es el valor de fase para cada símbolo a transmitir y φ es una constante de fase de la modulación. La señal modulada tendría el aspecto representado en al siguiente figura.

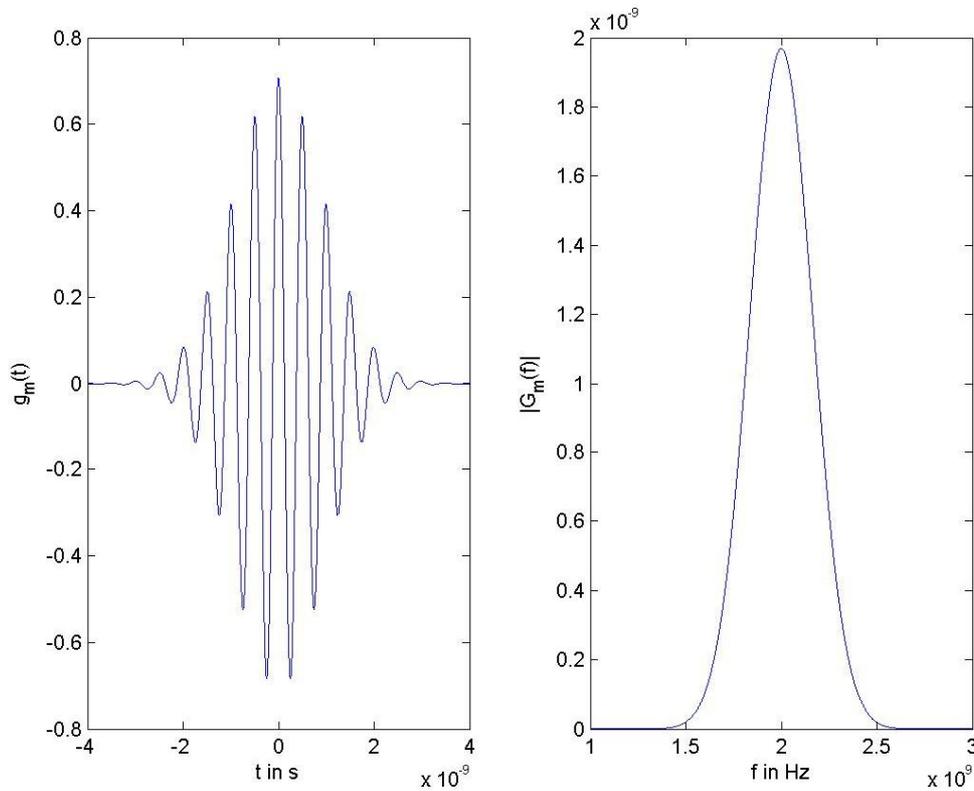


Fig. 4: Impulso modulado en dominio del tiempo y frecuencia.

Con este tipo de modulación pueden utilizarse en modo binario o cuadratura, los cuales permiten transmitir 2 o 4 símbolos respetivamente. En la solución software también está contemplado el caso en el que se puedan transmitir 8 o 16 símbolos, pero en la práctica no es utilizado ya que ambos modos son muy poco robustos.

La FCC no restringe el uso de distintos tipos de modulación, pero si lo hace con la potencia permitida para transmitir. En este caso, la señal transmitida puede tener una potencia máxima de -41,3 dB por cada MHz.

El receptor ha sido diseñado teniendo en cuenta tanto aspectos teóricos como hardware. En el siguiente esquema puede observarse un diagrama que muestra las distintas etapas de las que está formado el receptor.

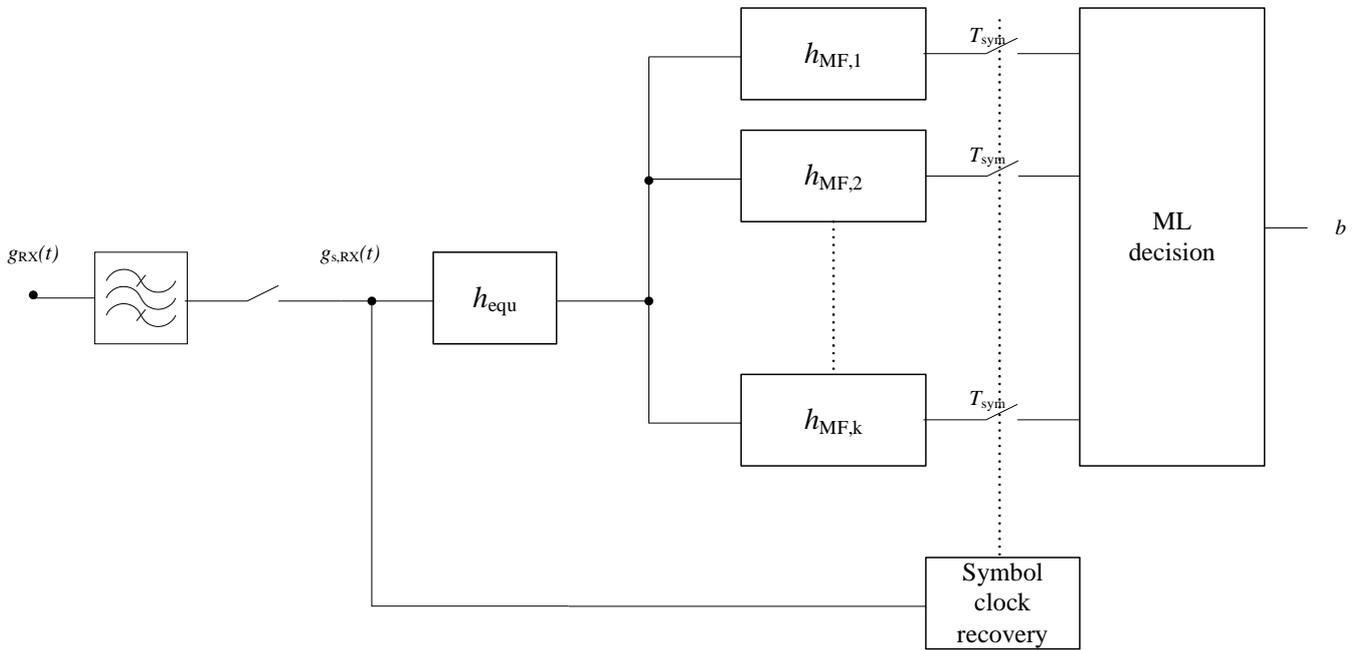


Fig. 5: Diagrama de diseño del receptor.

La señal recibida $g_{RX}(t)$ contiene los efectos del canal y del ruido aditivo gaussiano blanco. La fórmula que describe dicha señal puede escribirse de la siguiente forma:

$$g_{RX}(t) = g_{m,PSK}(t) * h(t) + n(t), \quad (3)$$

Los efectos del canal están representados mediante $h(t)$ mientras que los del ruido vienen dados por $n(t)$. Esta señal es procesada siguiendo el mismo procedimiento realizado en el transmisor, el muestreo paso banda y conversor mult-Nyquist. Como se realiza ya en el transmisor, la señal recibida es filtrada usando un filtro paso banda y es muestreada utilizando una frecuencia de muestreo que es dos veces el ancho de banda del filtro paso banda. La señal muestreada estará en la tercera zona de Nyquist por lo que es fácil obtener la réplica de dicha señal para ser procesada en la primera zona de Nyquist.

Una vez muestreada la señal, los efectos del canal son eliminados mediante un ecualizador cuyo diseño no se tiene en cuenta en este proyecto. Después de pasar por el ecualizador, la señal atraviesa un filtro adaptado. La función de este filtro es detectar la presencia de una señal conocida dentro de una señal con ruido recibida. Este sistema además maximiza la relación señal a ruido. La señal a la salida del filtro adaptado se obtiene calculando la correlación cruzada entre la señal desconocida y la “señal plantilla” o señal conocida. Esta operación viene dada por la siguiente formula

$$g_{MF}(t) = g(t) * h_{MF}^*(-t) = \int g(t) * h_{MF}^*(\tau - t) dt. \quad (4)$$

De este modo, tras atravesar el filtro adaptado, tendremos tantas señales como posibles símbolos esperamos recibir.

El criterio de decisión utilizado es el criterio de máxima verosimilitud. Como se ha dicho anteriormente, el filtro adaptado mide la similitud entre la señal recibida y una plantilla. Estas salidas son muestreadas según el periodo de símbolo y teniendo en cuenta la sincronización. El símbolo elegido para cada instante de símbolo es aquel que tiene mayor verosimilitud, es decir, el que más se parezca a la plantilla correspondiente. Es muy importante el momento de la sincronización puesto que dependiendo de ella, la secuencia obtenida puede ser totalmente correcta o totalmente errónea.

A partir de ahora se va a abordar como llevar a cabo la solución software y como comunicar matlab con el lenguaje C.

El problema de la comunicación entre C y Matlab fue resuelto utilizando Mex-files. Mex-file es una interfaz externa de Matlab que permite llamar tu propia rutina escrita en C o Fortran directamente desde Matlab como si fueran funciones escritas en dicho lenguaje. De este modo habilita la posibilidad de transferir datos desde Matlab ha C evitando guardar los datos como Mat-files y el consecuente tiempo necesitado para ello. De este modo es posible reducir el tiempo computacional de distintas operaciones que en Matlab serían muy lentas. Para nuestra aplicación, la principal funcionalidad del Mex-file es comunicar Matlab con C y transferir los datos de uno a otro.

La Mex-function funciona de la siguiente forma:

- La Mex-function es llamada desde Matlab transfiriéndole los parámetros necesarios. Estos parámetros son dos arrays que contienen los parámetros de entrada y de salida (*plhs y *prhs) y dos números enteros que indican la cantidad de parámetros de salida y entrada disponibles (nlhs y nrhs).
- Dentro de la Mex-funtion, es comprobado el número de parámetros recibidos lanzando una excepción si no son correctos.
- Se obtienen las variables almacenadas en los punteros.
- Se llama a la propia función escrita en C transfiriéndole los parámetros necesarios, en este caso sería la función qpsk_receiver().

La función donde se lleva a cabo todo el proceso de recepción y demodulación es la función “qpsk_receiver()”. La función comienza creando los modelos de los impulsos base y los distintos pulsos modulados posibles. Una vez creados los impulsos modulados, son utilizados para crear las plantillas de los filtros adaptativos. Usando las plantillas de los filtros adaptativos, la sincronización es llevada a cabo y finalmente la decisión de los símbolos recibidos tiene lugar.

La sincronización es una parte muy importante en el proceso de recepción. Dependiendo de su desempeño, el resultado obtenido puede ser satisfactorio o

desastroso. La sincronización es realizada usando una pequeña sección del principio de la señal recibida. El procedimiento consiste en coger una muestras a la frecuencia de símbolo en cada plantilla del resultado obtenido en el filtro adaptativo. Una vez se tienen las muestras, la decisión es tomada según el criterio de máxima verosimilitud explicado anteriormente. Después de este paso, se calculan las tasas de error. Este procedimiento es llevado a cabo una serie de veces variando una constante de tiempo que desplaza ligeramente el tiempo en el que se cogen las muestras. Finalmente, la sincronización correcta viene dada por el desplazamiento de tiempo que genera menos errores.

Una vez obtenida la sincronización, se desmodula la señal completa. Además, ha sido necesario implementar funciones existentes en matlab y que eran necesarias para realizar distintos cálculos como por ejemplo una consolación. Estas funciones son las siguientes:

- void add_awgn(double* rx_in, double snr, int long_rx, double* final_signal).
- void tukeywin(int L, double r, double* window).
- void conv(double* v, int n, double* u, int m, double* result).
- void fliplr(double* vector, int n_v, double* invert).
- double maxi(double *vector, int n_v).
- double randn(void).

Ya que el principal objetivo de esta investigación es mejorar el tiempo necesario para demodular la señal, en la siguiente figura se muestra una comparación del tiempo computacional requerido por las simulaciones realizadas con C y Matlab para diferentes longitudes de datos. A simple vista puede verse que el código escrito en C es mucho más rápido que el de Matlab, y a medida que la longitud de datos se incrementa, esta diferencia se ve aumentada. De esta forma, para una longitud de 100000 datos, la diferencia es de casi 30 segundos, lo que significa un ahorro del 75% del tiempo.

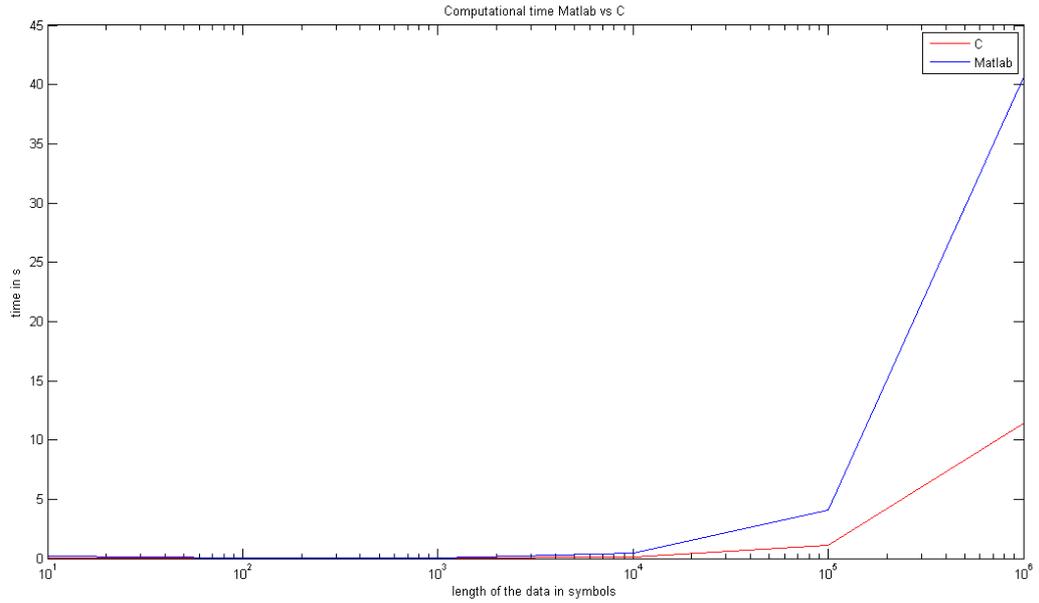


Fig. 6: Comparación de tiempo computacional de Matlab y C.

Además de aumentar la velocidad de ejecución del código, otro aspecto debe ser considerado. Este aspecto es la calidad de la recepción medida en tasas de error. Tras realizar distintas simulaciones, se ha comprobado que las tasas de error obtenidas por matlab se mantienen. La siguiente figura muestra una comparación de las tasas de error obtenidas para ambas simulaciones con distintas relaciones de señal a ruido.

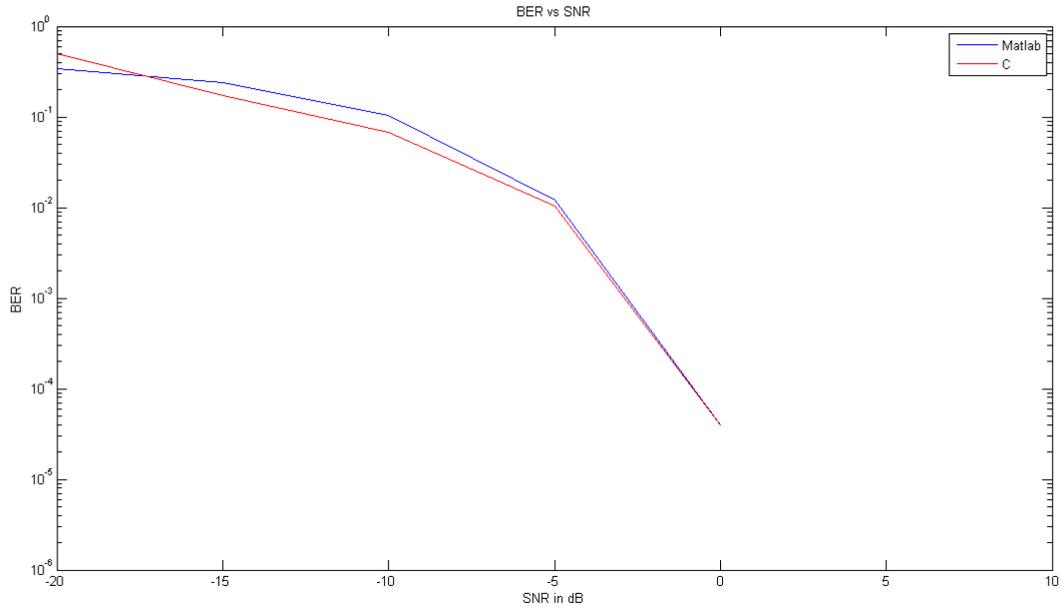


Fig. 7: Comparación de calidad entre Matlab y C.

El objetivo del proyecto ha sido cumplido ya que la implementación del receptor ha sido realizada con éxito disminuyendo el tiempo de computación necesario. Esta implementación puede ser útil para posibles aplicaciones en tiempo real e incluso como herramienta de medida para investigaciones de otros estudiantes.