

Proyecto de Equipos de Radio Definidos por Software

Sistema de comunicación en de transmisiones BFSK

18 de enero de 2018

Ander Doncel Llamas

Daniel Montesano

Tabla de contenido

1. Especificaciones de proyecto	3
2. Comunicación Monoportadora	4
2.1. Transmisor	4
2.2. Canal	5
2.2.1. Retardo	5
2.2.2. Ruido	5
2.2.3. Dispersión	5
2.2.4. Interferencias intencionadas	5
2.3. Receptor	7
2.3.1. Recepción no coherente con detección de envolvente	7
2.3.2. Recepción heterodina no coherente	8
2.3.3. Recepción heterodina y coherente utilizando PLLs sintonizados	9
2.4. Recuperación de reloj	14
2.5. Resultados	14
2.5.1. Recepción no coherente con detección de envolvente	15
2.5.2. Recepción heterodina no coherente	17
2.5.3. Recepción heterodina y coherente utilizando PLLs sintonizados	19
3. Comunicación de espectro ensanchado	23
3.1. Transmisión	23
3.2. Recepción	23
4. Conclusiones y líneas futuras	24

1. Especificaciones de proyecto

Este documento describe el proceso de diseño e implementación de un sistema de comunicación basado en transmisiones BFSK desarrollado en MATLAB. Este esquema de modulación implementa una variación en frecuencia sobre la de la señal portadora para transmitir los símbolos, de manera que se utilicen diferentes frecuencias para transmitir cada uno de los mismos. Dado que este caso se corresponde a la implementación binaria de una modulación en frecuencia FSK, los símbolos a transmitir se corresponderán únicamente a '1's y '0's. La siguiente figura ilustra el concepto de modulación BFSK.

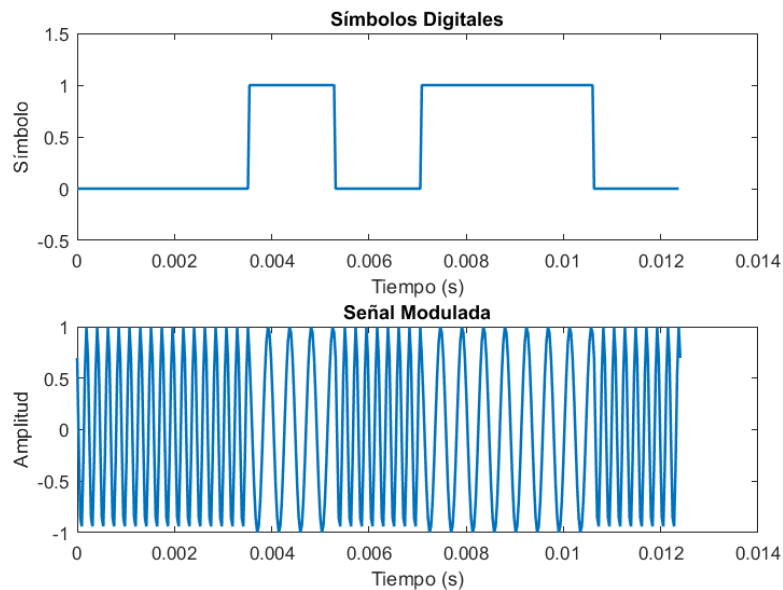


Figura 1. Modulación BFSK.

Los parámetros que describen el sistema de comunicaciones a implementar se describen a partir de los siguientes valores:

- Velocidad binaria (R_b) = 564,48 bps.
- Frecuencia de la portadora (f_c) = $\frac{f_1 + f_0}{2}$
- Los símbolos '1' estarán compuestos por 7 ciclos a frecuencia f_1 utilizada.
- Los símbolos '0' estarán compuestos por 8 ciclos a frecuencia f_0 utilizada.
- Separación de frecuencias (Δf) = $f_1 - f_0$ = 564,48 Hz.
- Duración de símbolo (T_b) = $\frac{1}{R_b}$ = 1,77ms.

Los valores de f_1 y f_0 utilizados se detallarán en las posteriores secciones para cada uno de los escenarios de diseño implementados, ya que vendrán definidos de la frecuencia de la señal portadora escogida.

En primer lugar, el sistema deberá mostrar un comportamiento tolerante ante los siguientes posibles efectos producidos por el canal de transmisión:

- Retardo de propagación arbitrario.
- Ruido blanco y gaussiano.
- Dispersión de hasta 180 μ s de exceso de retardo.
- Presencia de perturbación de banda estrecha intencionada.

Por otro lado, el sistema deberá tolerar posibles diferencias en las frecuencias de portadora (f_c) o de símbolo (f_1 o f_0) en el transmisor y en el receptor de acuerdo a unas tolerancias dadas. Estas posibles variaciones vienen dadas a partir de las siguientes indicaciones de tolerancia:

- Velocidad binaria (R_b) = 564,48 bps \pm 2,5%.
- Separación de frecuencias (Δf) = 564,48 Hz \pm 7%.

2. Comunicación Monoportadora

En primer lugar, se llevará a cabo el diseño de un transmisor y un receptor para los valores nominales de requisitos y portadora fija. Para ello, los valores utilizados desde el punto de vista de las prestaciones de transmisión vienen definidos a partir de los siguientes requisitos:

- La frecuencia portadora (f_c) utilizada deberá ser igual a $7.5 \cdot R_b = 4.233$ kHz.
- La frecuencia de los símbolos '1' deberá ser igual a $7 \cdot R_b = 3.951$ kHz.
- La frecuencia de los símbolos '0' deberá ser igual a $8 \cdot R_b = 4.515$ kHz.
- La separación de frecuencias (Δf) deberá ser igual a la velocidad binaria (R_b) del sistema.
- La frecuencia de muestreo (f_s) utilizada será de $56 \cdot R_b = 31.611$ kHz.

A lo largo de las siguientes secciones se describirá la arquitectura y el funcionamiento del sistema implementado.

2.1. Transmisor

Se generará el mensaje a transmitir de manera aleatoria, comprendido por '0's y '1's, con 50% de probabilidad para cada símbolo. El mensaje puede tener una longitud configurable, aunque dentro de unos mínimos explicados más adelante.

Una vez definidos tanto el mensaje de '0's y '1's a transmitir como las frecuencias de símbolo, el transmisor transformará la información binaria en una señal modulada en frecuencia. Además, se impondrá la condición de que en las transiciones entre pulsos la fase deberá ser continua, por lo que será necesario llevar el seguimiento de la misma.

Para llevar a cabo este procedimiento, se ha optado por llevar a cabo el seguimiento de la fase de la señal para cada uno de los símbolos, de manera que la fase inicial del siguiente símbolo sea la final del anterior símbolo. Para ello, se computará la variación de fase entre intervalos de muestreo de la siguiente manera:

$$\Delta\theta_1 = \frac{2\pi f_1}{f_s} = \omega_1 T_s$$

$$\Delta\theta_0 = \frac{2\pi f_0}{f_s} = \omega_0 T_s$$

Una vez calculada esta variación de fase, se establecerá de manera arbitraria la fase inicial de la señal a transmitir y el transmisor procederá a la generación un vector que contenga la fase de la señal para cada instante temporal de muestreo. Conociendo el intervalo de duración de cada símbolo (T_b) se irá sumando $\Delta\theta_1$ o $\Delta\theta_0$ (en función del símbolo que se esté enviando) a la fase de la muestra anterior N veces, donde N es la cantidad de instantes de muestreo contenidos en este intervalo T_b .

$$N = T_b * F_s = \frac{1}{R_b} * 56 * R_b = 56$$

La última actualización de fase del símbolo anterior se corresponderá a la fase inicial del siguiente símbolo, independiente de si es un '1' o un '0'. Este proceso se realizará iterativamente hasta finalizar la generación del vector de fase de la señal. Por último, la señal modulada se podrá obtener simplemente a partir del coseno del vector obtenido.

$$s_{mod}(t) = \cos \theta(t)$$

2.2. Canal

Como se ha visto previamente, el canal genera cuatro tipos de distorsiones que el sistema ha de ser capaz de soportar.

2.2.1. Retardo

Como ocurre en cualquier canal de transmisión, la señal se retarda de manera arbitraria. Sin embargo, en este sistema de comunicación, el retardo debería ser transparente, ya que no hay comunicación en doble sentido ni la comunicación es a tiempo real. La única complicación que causa es detectar el origen de tiempos. Para mitigarlo, habrá que incorporar tramas para detectar el inicio de la transmisión, que será detallado más adelante.

En el código el retardo del canal se calcula a partir del retardo de transmisión de la señal para una distancia dada, si bien se podría indicar directamente el retardo.

2.2.2. Ruido

El canal añadirá señales no deseadas a la transmisión. Como mínimo, se tratará de ruido blanco gaussiano. Se genera ruido blanco gaussiano de media cero, con una potencia arbitraria, y se suma a la señal. Cuanta más potencia de ruido sea capaz de soportar el sistema, mejor será el sistema. Es por esto que se estudian diferentes técnicas en la recepción, como se verá más adelante.

2.2.3. Dispersión

El canal introducirá una dispersión de hasta 180µs de retardo. Esto se modelará como dos deltas: la primera es la señal directa y la segunda tendrá la misma amplitud y estará retardada hasta 180µs. Sin embargo, el programa se ha planteado para un caso más amplio: arbitrarias señales de multitrayecto con amplitudes arbitrarias. Se especifica el retardo máximo que pueden tener las réplicas de las señales, el número de reflexiones y la amplitud de estas. Las reflexiones se distribuyen aleatoriamente en el rango temporal que se ha especificado, y luego se ajustan las amplitudes (un caso general puede ser amplitud aleatoria pero siempre menor que la señal directa).

2.2.4. Interferencias intencionadas

La técnica más común y sencilla de interferir en un sistema es mediante *jamming*. Esto se consigue introduciendo señales indeseadas de gran potencia en la banda usada por el sistema de comunicaciones. Las especificaciones indican que habrá interferencias que contengan como mucho la potencia original de la señal. Esto se entiende como que la potencia de transmisión será la misma, pero en todo el ancho de banda usado en el sistema, por lo que la densidad espectral de potencia de las interferencias será menor.

La potencia de una sinusoide, como es el caso de la señal deseada, es su voltaje RMS al cuadrado dividido entre la resistencia en la que se disipa la potencia. La resistencia se asumirá 1 para simplificar los cálculos, ya que no es relevante al estar comparándose dos potencias medidas en

la misma resistencia. El voltaje RMS de un seno es su voltaje de pico entre $\sqrt{2}$. El voltaje de pico en el sistema es 1 V, por lo que la potencia serán 0.5 W.

$$V_{RMS} = \frac{V_{peak}}{\sqrt{2}}$$

$$P_S = \frac{V_{peak}^2}{R}$$

El sistema usa un ancho de banda (BW) de $F_c \pm R_b$, es decir, $2 \cdot R_b$. El sistema interferente tendrá que usar la misma potencia en ese ancho de banda. Esto nos deja una densidad espectral de potencia de interferencia:

$$PSD_I = \frac{P_s}{BW}$$

Para generar esta interferencia se generará ruido blanco gaussiano de media cero con varianza:

$$\sigma^2 = PSD_I$$

Este ruido será filtrado para que solo haya interferencia en la banda de trabajo con un filtro de un orden alto para que tenga cortes pronunciados.

Sin embargo, se puede apreciar que la interferencia tendrá muy poco efecto, estando la mayoría de las veces por debajo del ruido blanco. Esto se debe a que la señal concentra toda la potencia en una única sinusoide, mientras que la interferencia concentra la misma potencia en una banda bastante amplia.

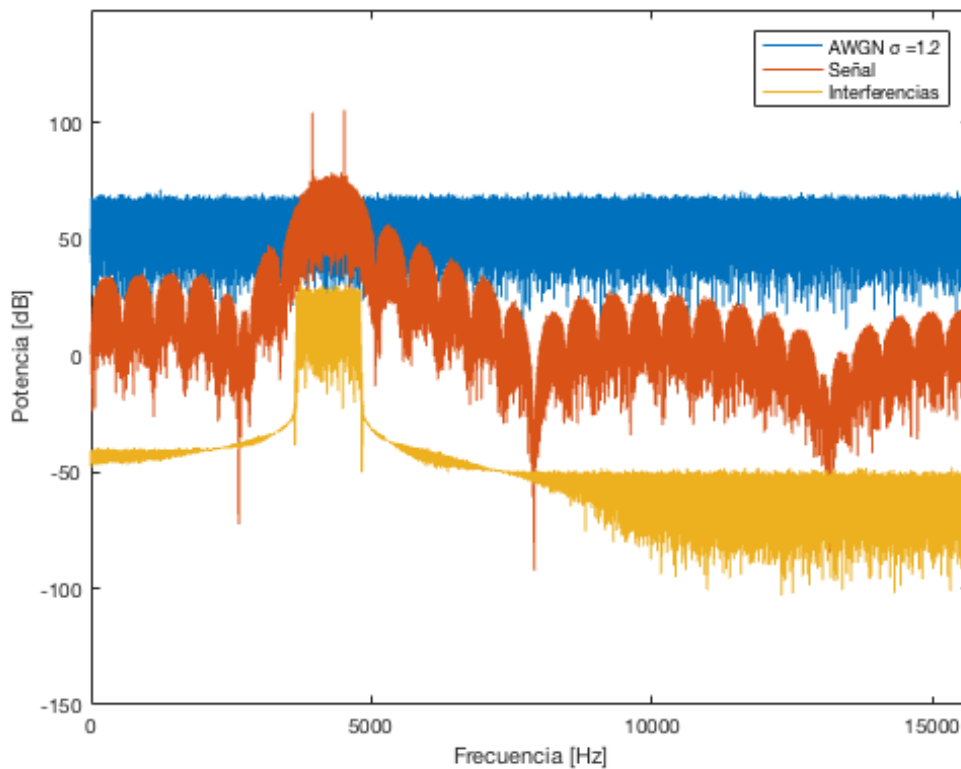


Figura 2: Espectro de las señales en el canal de transmisión.

Se puede ver que efectivamente, las interferencias se encuentran por debajo de la señal a transmitir y muy por debajo de un ruido blanco con $\sigma = 1.2$.

2.3. Receptor

Tras pasar por el canal de transmisión, la señal se recibirá en el receptor, que tratará de recuperar los símbolos enviados por el transmisor. Para ello, se han estudiado las 3 diferentes alternativas de recepción que se describen en los siguientes apartados.

2.3.1. Recepción no coherente con detección de envolvente

Con el fin de ser capaces de visualizar el funcionamiento del sistema completo, se ha optado por llevar a cabo un primer receptor simple que realizará una detección homodina no coherente de la señal. La siguiente figura esquematiza la implementación de este modo de recepción.

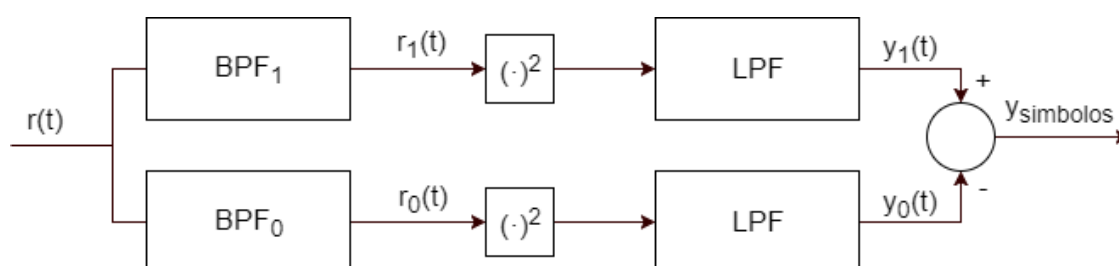


Figura 3. Esquema de Filtrado Paso Banda & Detección de Envolvente.

En primer lugar, se implementarán filtros paso banda para cada una de las frecuencias de símbolo. La señal recibida se filtrará de manera independiente con ambos filtros, de manera que se obtengan dos señales diferentes en las cuales estará contenida únicamente la señal correspondiente a uno de los dos símbolos. Finalmente, se buscará la envolvente de los símbolos utilizando un detector de ley cuadrática que combina una etapa de no-linealidad y un filtro paso bajo.

Las características de los 3 filtros implementados (el filtro paso bajo será el mismo para las dos señales) se resumen en la siguiente tabla:

	Filtro Pasobanda '1'	Filtro Pasobanda '0'	Filtro Pasobajo
Tipo	Butterworth	Butterworth	Butterworth
Orden	2	2	2
$f_{\text{corte},1}$ (Hz)	$f_c - R_b$	f_c	R_b
$f_{\text{corte},2}$ (Hz)	f_c	$f_c + R_b$	-

Tabla 1. Descripción de los parámetros de los filtros.

Las siguientes figuras ilustran la respuesta en magnitud de los filtros implementados.

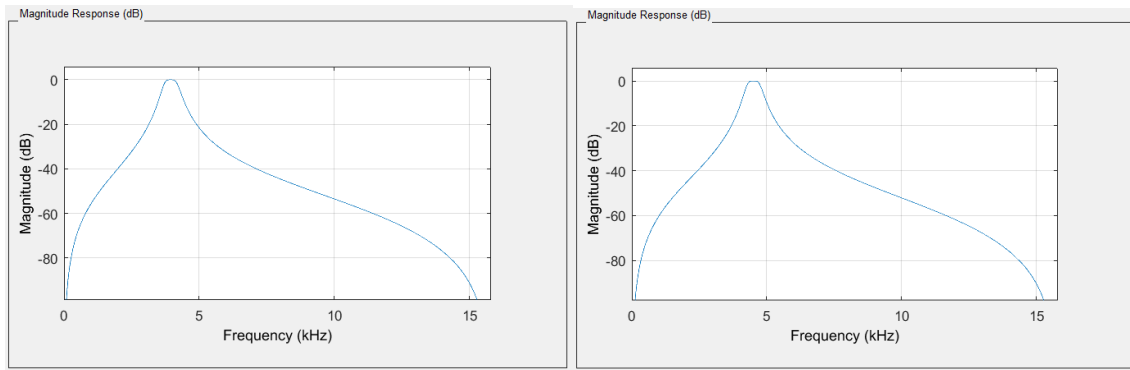


Figura 4. Respuesta en magnitud del filtro paso banda del símbolo '1' (izquierda) y símbolo '0' (derecha).

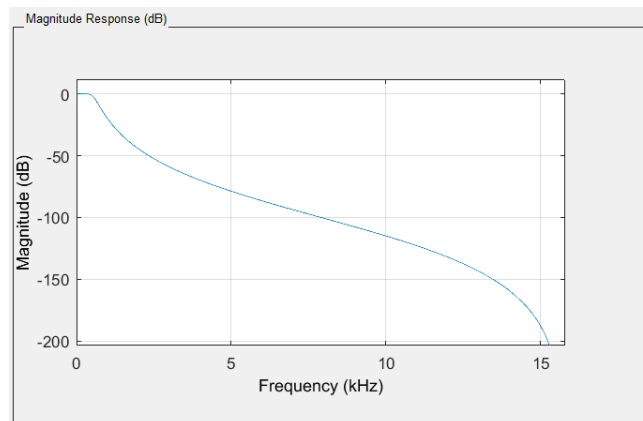


Figura 5. Respuesta en magnitud del filtro paso bajo.

2.3.2. Recepción heterodina no coherente

Este método sigue una arquitectura de receptor heterodino, usando osciladores locales para la recepción. Sin embargo, es un sistema no coherente, es decir, estos osciladores no están enganchados en fase con el receptor. Esto puede causar graves consecuencias si los osciladores están desfasados. Este método queda obsoleto con el método 2.3.3, que usa PLL. Se hizo como una primera versión, más rápida y sencilla que los PLL, para poder probar el sistema completo.

En el siguiente diagrama se puede ver la arquitectura del receptor.

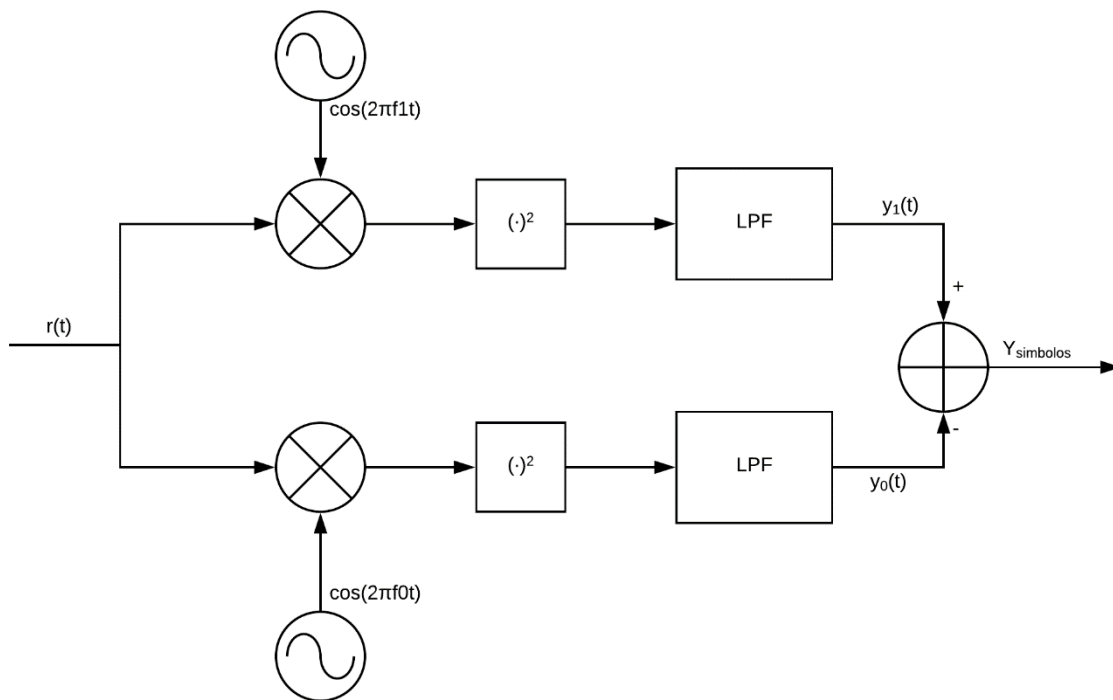


Figura 6. Demodulación Heterodina No Coherente.

Se generan tonos a la frecuencia de cada símbolo, que al ser mezclados con la señal la bajan a banda base. Sin embargo, también mezclarán componentes no deseadas, que hay que filtrar con un filtro paso bajo. Este filtro ha de ser lo suficiente rápido como para dejar pasar los símbolos, pero nada más. Es por esto, que la frecuencia de corte del LPF será R_b .

2.3.3. Recepción heterodina y coherente utilizando PLLs sintonizados

Al igual que en método descrito en la sección anterior, esta modalidad implementará una arquitectura de recepción heterodino, usando osciladores locales para la recepción. Sin embargo, este modo implementará un método de recepción coherente en el que se tendrá en cuenta la fase con la que se recibe la señal.

Dado que los '1's y los '0's enviados corresponden a diferentes frecuencias, se implementarán dos PLLs sintonizados a las frecuencias de cada uno de los símbolos, de manera que cada uno de ellos identifique el offset en fase con el que se reciben los símbolos.

Para llevar a cabo esta tarea se ha realizado llevado a cabo dos distintas implementaciones, de manera que se pueda estudiar el comportamiento de ambas y elegir la que mejor se adapta a las necesidades del sistema. La Figura 6 ilustra el esquema de la primera modalidad de PLL implementada para el símbolo '0'. Su diseño sería análogo para el símbolo '1', cambiando únicamente la frecuencia de sintonización f_0 a f_1 .

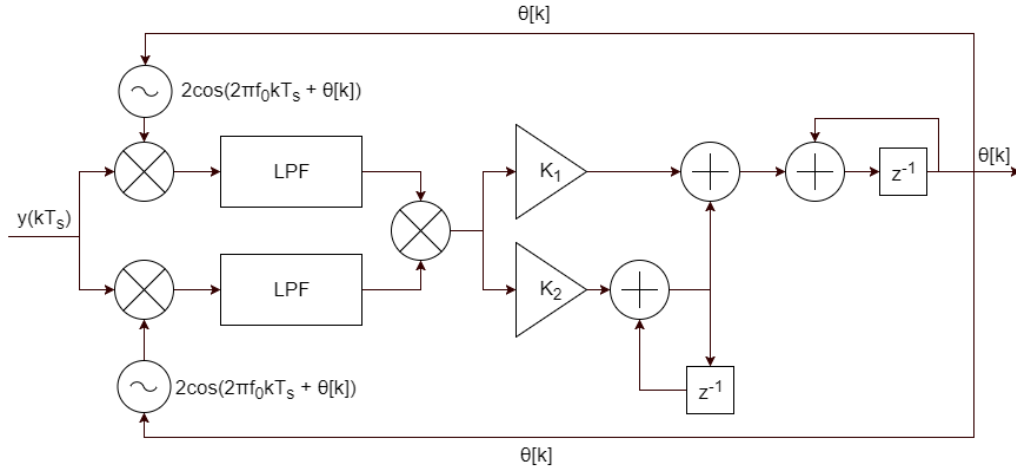


Figura 7. Diseño del PLL.

Como puede verse, el PLL implementado consiste en un bucle de Costas. Sin embargo, a fin de obtener mejores resultados, la parte integradora del algoritmo se ha sustituido por la de un PLL más realista. Para ello, será necesario definir los siguientes parámetros:

- Ganancia del NCO (K_0) → El valor de este parámetro se ha fijado a 1.
- Ratio de amortiguación (ζ)
- Ancho de banda de ruido (B_n)

A partir de estos valores, se calcularán los valores de K_1 y K_2 siguiendo las siguientes ecuaciones:

$$K_0 K_1 = \frac{4\zeta\omega_n}{1 + 2\zeta\omega_n + \omega_n^2}$$

$$K_0 K_2 = \frac{4\zeta\omega_n}{1 + 2\zeta\omega_n + \omega_n^2}$$

$$\omega_n = \frac{B_n T}{\zeta + \frac{1}{4\zeta}}$$

Por su parte, la segunda modalidad implementada se corresponde a un PLL, cuyo diseño se muestra en la Figura 7, también particularizada para el símbolo '0'. Puede verse que el Loop Filter y el NCO (*Numerically Controlled Oscillator*) son equivalentes a los de la implementación basada en el Bucle de Costas y difieren únicamente en la detección de fase.

Para ambos casos, utilizando el vector de fases estimado por el bucle, se generará la señal coseno que se utilizará como oscilador local para demodular la señal a banda base al igual que en la sección anterior, donde el coseno utilizado estará sincronizado en fase con la de la señal.

$$PLL_{out,1} = \cos(2\pi f_1 k T_s + \theta_1[k])$$

$$PLL_{out,0} = \cos(2\pi f_0 k T_s + \theta_0[k])$$

Tras probar ambos diseños, finalmente se ha decidido utilizar la implementación de PLL (Figura 7), ya que ha presentado resultados de convergencia ligeramente superiores a los de la implementación basada en el Bucle de Costas, además de ofrecer unas mejores prestaciones en tiempo computacional.

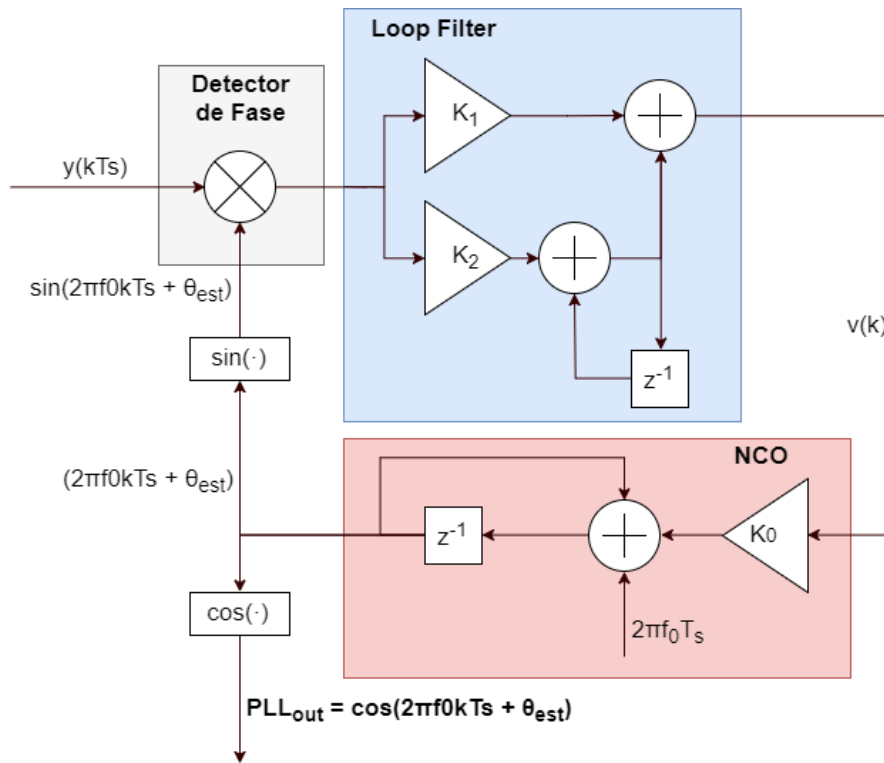


Figura 8. Esquema del PLL. Diseño 2.

Finalmente, la Figura 8 presenta el esquema del receptor. Utilizando las señales $PLL_{out,1}$ y $PLL_{out,0}$ como señal demoduladora, se llevará la señal a banda base utilizando, por un lado, el coseno de salida del PLL sintonizado a la frecuencia del símbolo '1' y, por otro, el coseno de salida del PLL sintonizado a la frecuencia del símbolo '0'. A continuación, se llevará a cabo un filtrado paso bajo de ambas señales, de manera que se tengan de manera independiente la señal contenedora de los símbolos '1' y la contenedora de los símbolos '0'. Sumando estas señales, con una de ellas invertida en signo, se tendrá finalmente la señal de símbolos a muestrear.

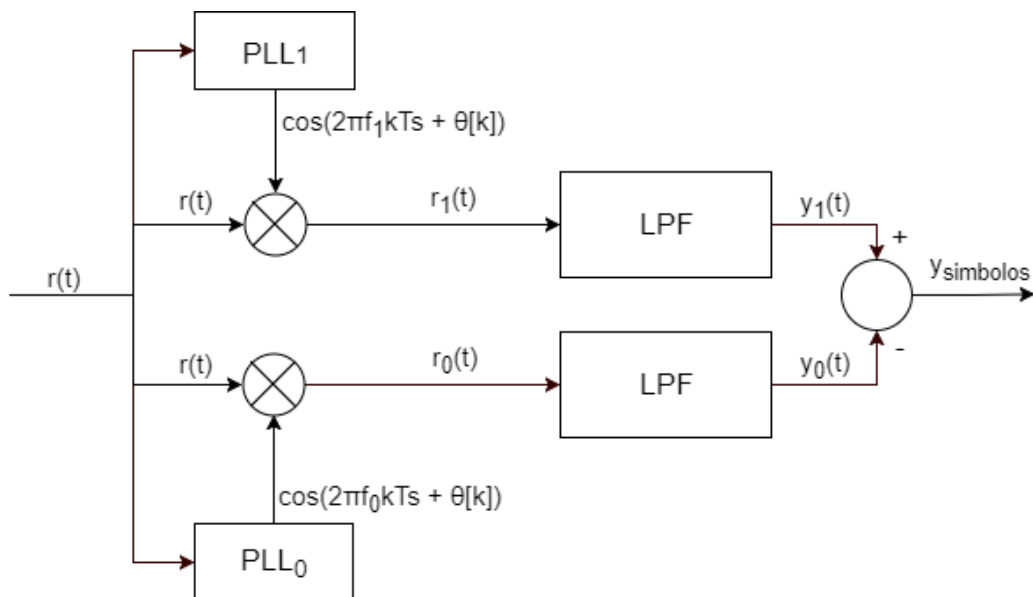


Figura 9. Demodulación Coherente utilizando PLLs sintonizados.

Los filtros paso bajo usados deben tener el ancho de banda R_b , para filtrar componentes suma de la mezcla. Es por esto que las exigencias del filtro no son elevadas, ya que las componentes “suma” o incluso la frecuencia de portadora están muy por encima de la banda base.

Sin embargo, como toda la señal es mezclada con las salidas del PLL, incluido los símbolos que no son de esa frecuencia, si vemos las componentes en frecuencia, vemos un tono justo en R_b , que es la diferencia entre portadoras. En la siguiente imagen se muestra el espectro de la señal recibida y la señal desmodulada con uno de los PLL antes de entrar a los filtros.

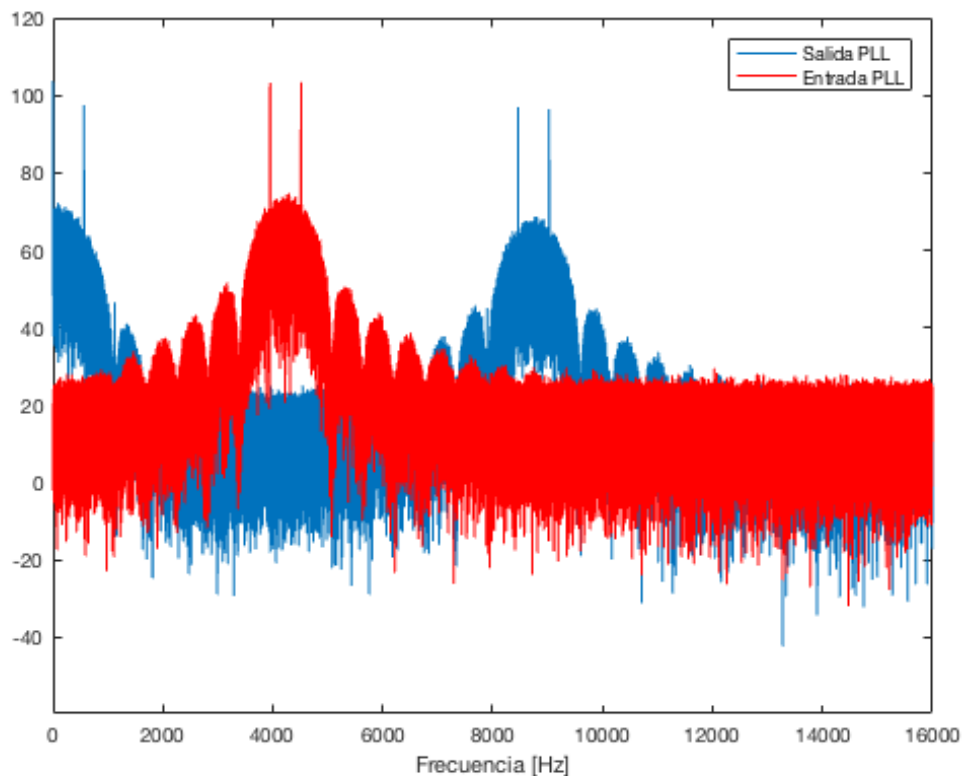


Figura 10: espectro de la entrada y de la salida del PLL

En rojo podemos ver la señal con las dos portadoras a la frecuencia de transmisión. En azul, vemos la señal tras ser mezclada con la salida de un PLL. Se ven la componente suma y la componente resta de cada una de las frecuencias. En 0 Hz esta la portadora del PLL en cuestión, y en R_b un tono correspondiente a la otra frecuencia. Y en $2 \cdot F_c$ hay una réplica de los tonos.

Si la frecuencia de corte del filtro es menor a R_b , el filtro no será suficiente rápido para símbolos alternados. Sin embargo, si es mayor que R_b , incluirá el tono de R_b y puede tener efectos indeseados. En la siguiente imagen, vemos estos efectos. En azul se muestra la señal filtrada con una frecuencia de corte de $2 \cdot R_b$, es decir, incluye el tono de R_b Hz. En rojo esta la señal filtrada con una frecuencia de corte de $0.9 \cdot R_b$, es decir, no incluye el tono de R_b Hz.

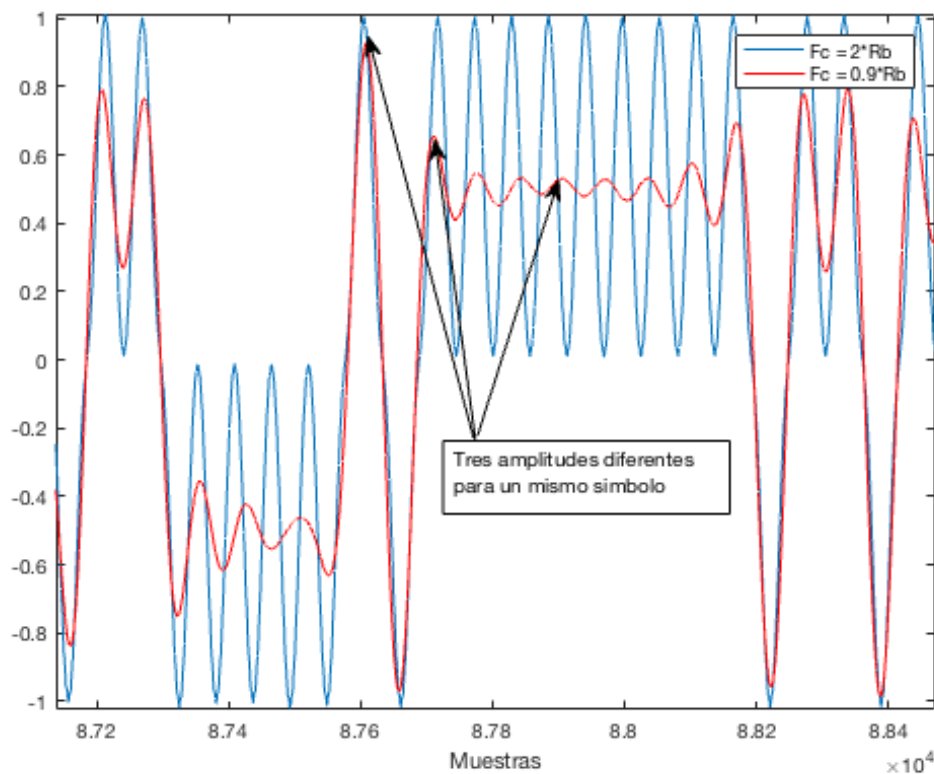


Figura 11: Salida del filtro con diferentes frecuencias de corte.

En principio, convendría usar la señal sin el tono R_b . Sin embargo, vemos que la respuesta temporal de los filtros hace que haya más de dos niveles en la constelación. Esto es debido a la interferencia entre símbolos, no teniendo la misma amplitud si los símbolos son alternados o repetidos. Además, la amplitud varía una oscilación amortiguada, cuya frecuencia depende del filtro.

Sin embargo, si se observa la señal incluyendo R_b , podemos ver que solo habría dos niveles para cada símbolo, uno o cero. Seleccionar el cero podría ser nefasto para la señal. Pero teniendo en cuenta que tiene frecuencia R_b , y la frecuencia de muestreo de símbolo será de R_b , se podría encontrar un punto en el que siempre se muestree en el nivel más alto. Esto se consigue con la recuperación de reloj que se explicará más adelante.

Tras realizar varias pruebas y comprobaciones, si usa la señal con la componente R_b , la recuperación de reloj converge más fácilmente y ofrece mejores resultados. Además, el filtro paso bajo tendrá que ser menos restrictivo. Por todo esto, se escoge no filtrar R_b , usando un filtro de orden bajo cuya respuesta en frecuencia puede verse a continuación.

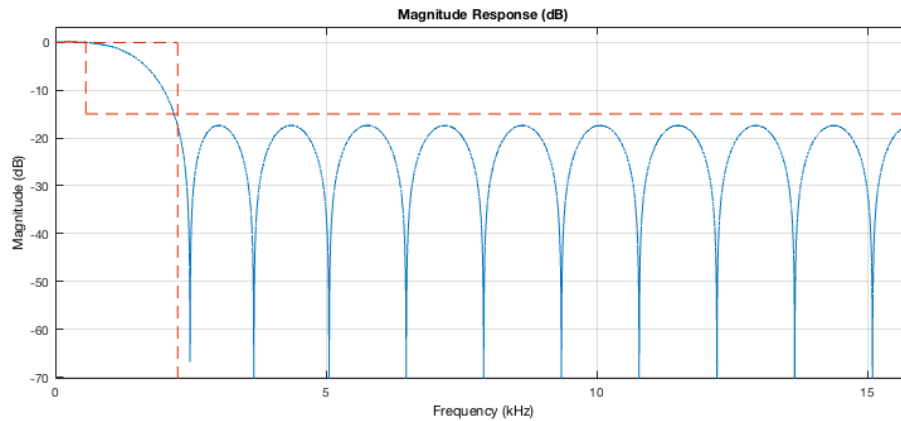


Figura 12: Respuesta en frecuencia de LPF de los PLL

2.4. Recuperación de reloj

Una vez la implementada la demodulación de la señal y se haya obtenido la señal $y_{simbolos}$ que contiene la representación analógica de la señal que transporta los símbolos, se deseará hallar el instante de muestreo ideal de manera que se pueda recuperar el mensaje transmitido.

Para llevar a cabo este proceso se ha utilizado el método de recuperación de tiempos dirigido por decisión. Este método tratará obtener el instante de muestreo en la amplitud de la señal recibida tiende a los niveles de la constelación. Esto se conseguirá interpolando la señal a instantes temporales desplazados un determinado tiempo $\pm\delta$ con respecto al instante de muestreo seleccionado. De esta manera, el algoritmo podrá comprobar en qué dirección se tiene que desplazar para encontrar el instante de muestreo óptimo. Para ello se interpola usando una *sinc*. Con esto se consigue recuperar la forma de la señal, sin que la amplitud se vea afectada debida al muestreo una vez el algoritmo converge.

2.5. Resultados

Esta sección mostrará los resultados que se han obtenido para cada uno de los tipos de recepción implementados, de manera que podrá comprobarse la robustez que ofrece cada uno de ellos. Se ha establecido un valor de BER = 1% como límite a partir del cual se considerará que la recepción no es aceptable.

La señal transmitida utilizará cabeceras que servirán tanto para detectar el inicio y final del mensaje así como para comenzar la estimación de fase en los PLLs o llevar a cabo la recuperación del reloj a lo largo de los símbolos de la cabecera en lugar de hacerlo directamente sobre el mensaje. Para todas las simulaciones considerará:

- Una cabecera pseudoaleatoria de 300 símbolos que conocen tanto el transmisor como el receptor previa al mensaje. Con esto se consigue detectar el inicio del inicio de la información mediante correlación.
- Una cabecera pseudoaleatoria de 300 símbolos que conocen tanto el transmisor como el receptor posterior al mensaje. Con esto se consigue detectar el final del inicio de la información mediante correlación.
- Un mensaje que contiene un total de 10000 símbolos ('0's o '1's) generados aleatoriamente.
- Un retardo en recepción arbitrario. Se calcula este retardo en función de la distancia, aunque podría ser cualquier retardo ya que es transparente para el sistema.

2.5.1. Recepción no coherente con detección de envolvente

En primer lugar, se ha llevado a cabo una simulación considerando únicamente un ruido blanco de desviación típica $\sigma = 0.01$. Al tratarse de un ruido considerablemente bajo, el resultado obtenido ha sido satisfactorio ya que el receptor ha sido capaz de obtener una BER del 0%. La Figura 8 muestra tanto la constelación con los valores muestreados, así como la trayectoria del parámetro de offset de reloj τ . Puede apreciarse como a medida que el parámetro de offset converge a su debido valor, los '1's y '-1's se muestran bien diferenciados en la constelación, dado que se estará muestreando en el instante óptimo.

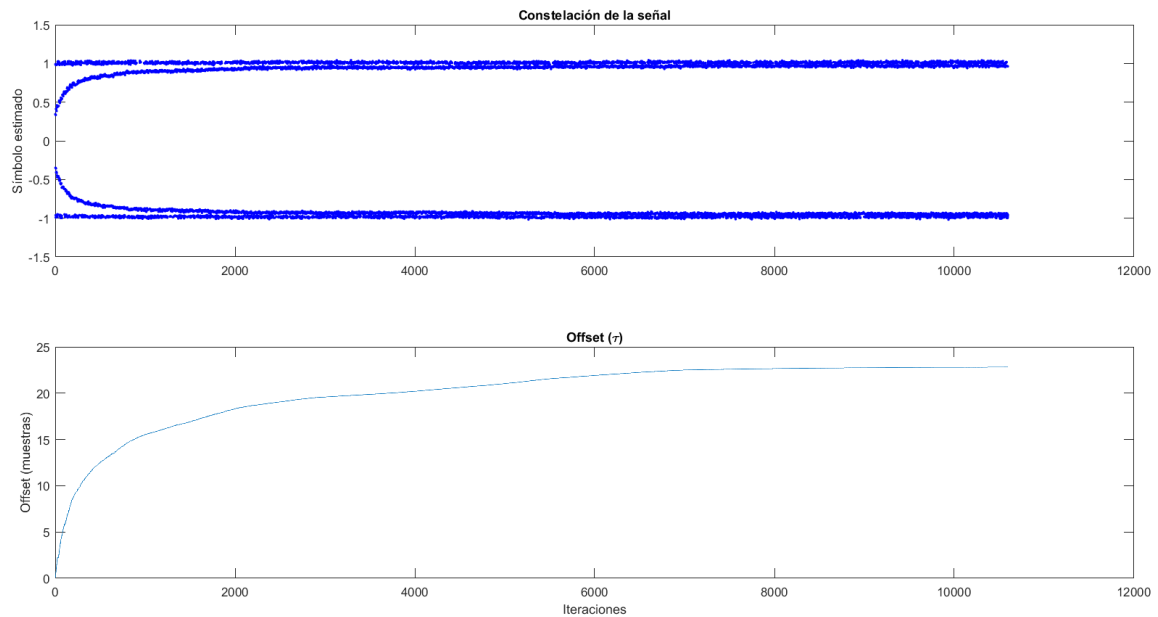


Figura 13. Receptor 1: Simulación de bajo ruido.

Cabe destacar que, durante el tiempo de convergencia, da la impresión de que hay dos niveles símbolos por cada símbolo. Esto se debe a que, si hay varios símbolos iguales, el instante de muestreo no será crítico, y siempre mostrará el valor correcto. Solo se apreciará la convergencia en aquellos símbolos que sean alternados.

En segundo lugar, se ha probado a aumentar la potencia de ruido de la señal recibida. Los resultados obtenidos para una desviación típica $\sigma = 1.2$ se muestran en la Figura 9. Como puede comprobarse, la constelación obtenida es considerablemente más ruidosa y '1's y '-1's se distinguen más difícilmente que en el caso de bajo ruido. En consecuencia, el BER ha aumentado su valor a 0.2%.

Cabe destacar que ha podido comprobarse que a partir de una desviación típica de $\sigma = 1.3$ el BER obtenido sobrepasa el 1% que hemos establecido, por lo que este sería el límite de ruido que este sistema podría contemplar.

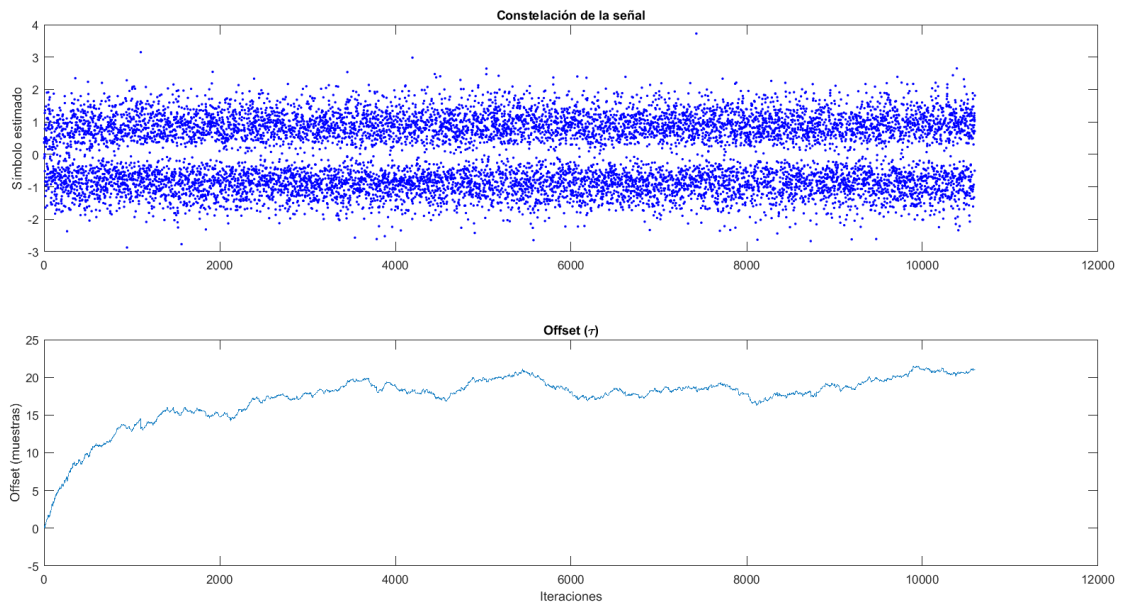


Figura 14. Receptor 1: Simulación de alto ruido.

Por último, se comprueba el efecto de la dispersión. Como se ha comentado antes, la dispersión tiene valores aleatorios en cada transmisión. Sin embargo, para esta demostración, se toman los valores de caso peor: dispersión de 180us con la misma amplitud que la señal. En la siguiente imagen podemos ver los efectos de la dispersión en la constelación.

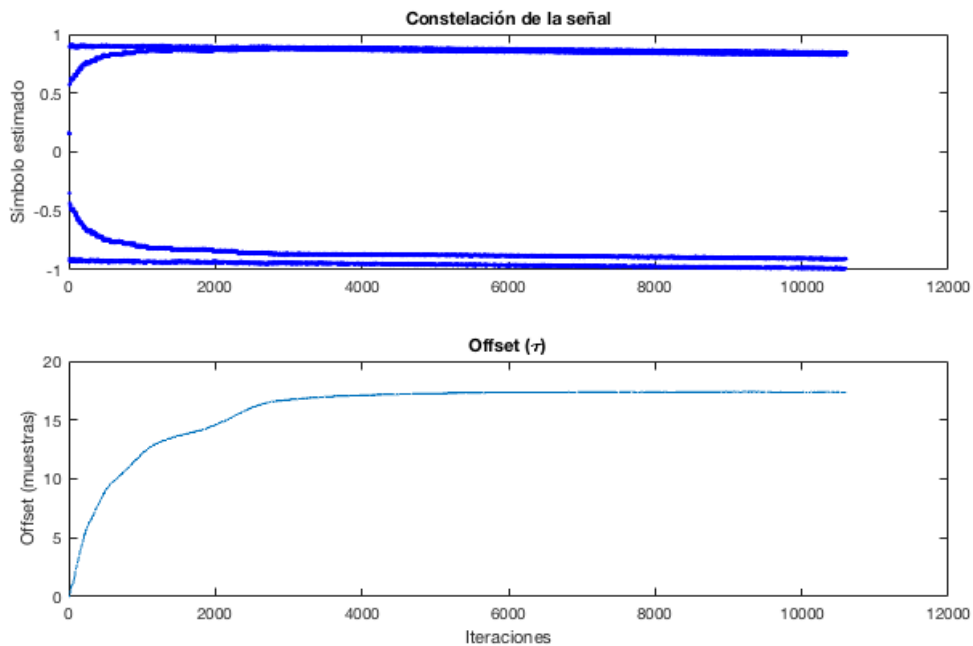


Figura 15: Receptor 1: Simulación de alta dispersión.

Se puede apreciar que aparecen numerosas muestras que no están centradas en los valores nominales de la constelación.

2.5.2. Recepción heterodina no coherente

Se ha llevado a cabo una simulación considerando únicamente un ruido blanco de desviación típica $\sigma = 0.01$. El BER es del 0%, pero se ve que la constelación es peor que en los otros receptores para el mismo caso.

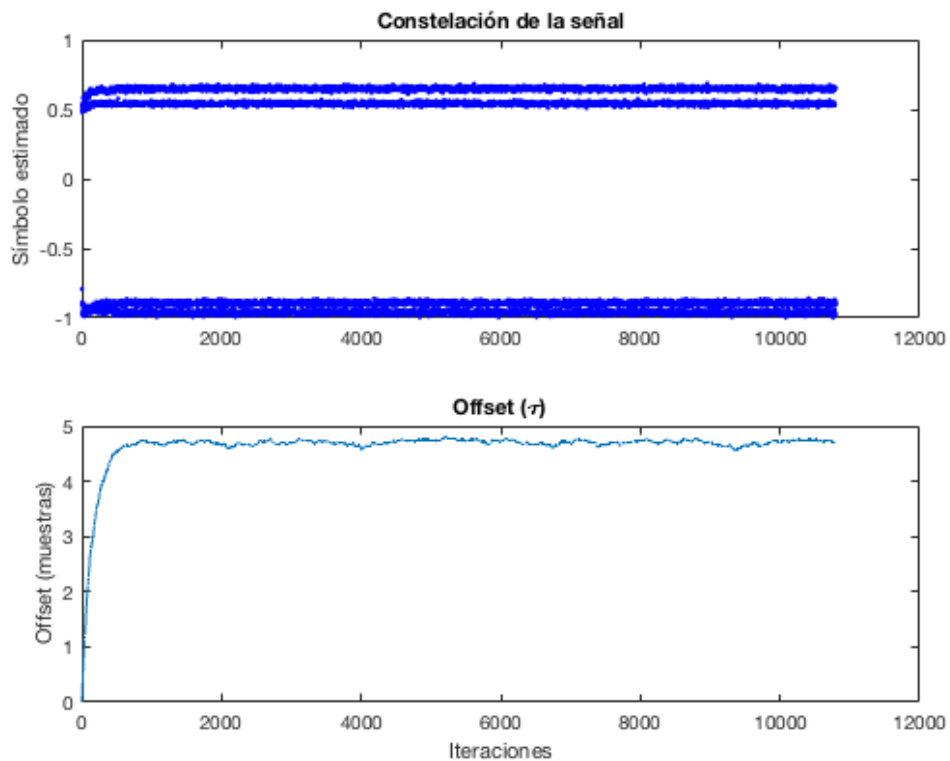


Figura 16: Receptor 2: Simulación de bajo ruido.

A continuación, se realiza la prueba con ruido blanco con ruido con $\sigma = 1$, con lo que se consigue una BER de 1%. Se puede ver que se ha empeorado un poco respecto al caso peor, aunque no es demasiado grande. Los resultados se pueden ver en la siguiente imagen.

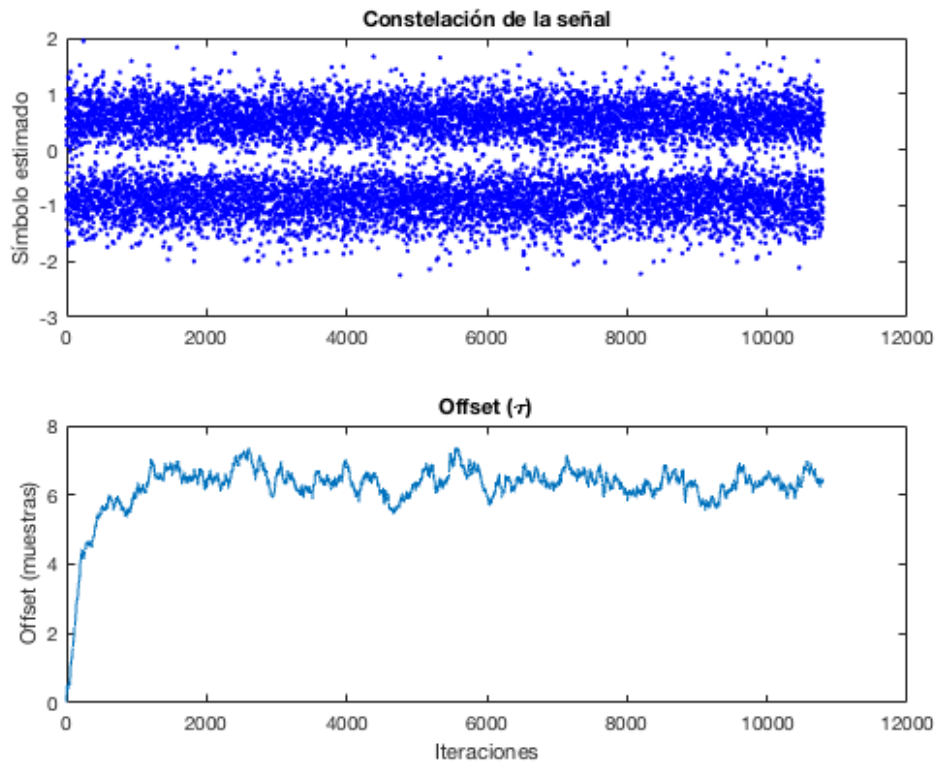


Figura 17: Ruido de 1

La siguiente imagen muestra los resultados cuando se incorpora dispersión en el caso peor, como en casos anteriores.

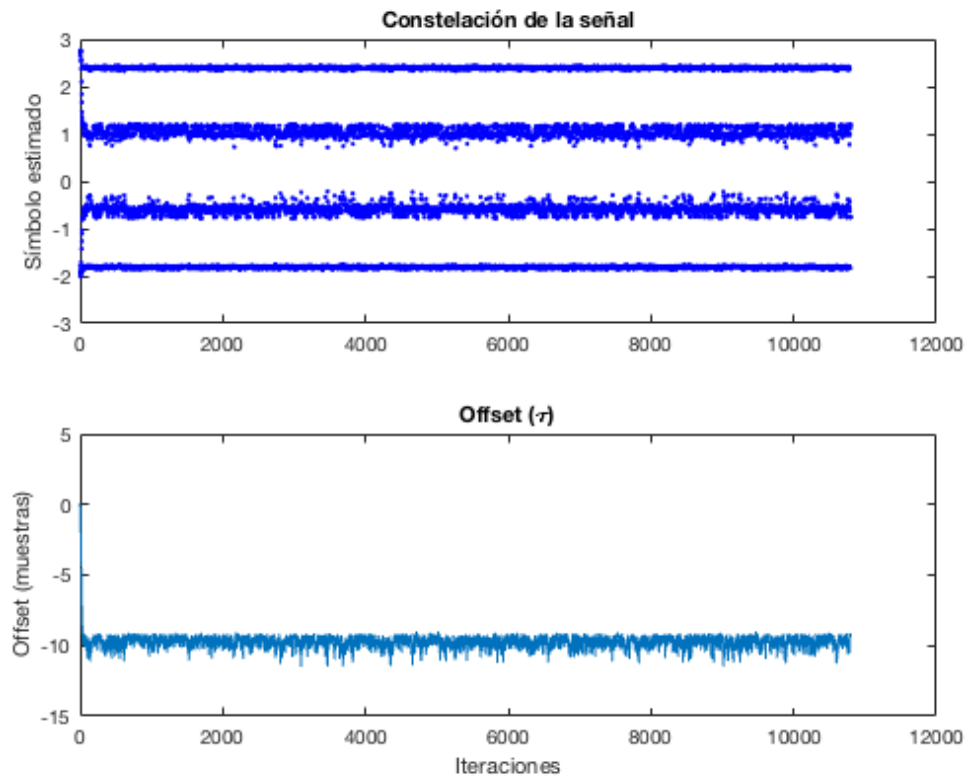


Figura 18: Dispersión de $0.5 \cdot 180$

Se puede apreciar que tiene efectos mucho más graves que en casos anteriores. No se ha intentado mejorar el funcionamiento de este receptor ya que se sabía que los otros sistemas ofrecerían mejor cualidades.

Por último, se realizan pruebas que cambian el offset de fase. Como se ha comentado anteriormente, pese a ser un receptor heterodino, los osciladores no están enganchados. Así que cualquier desfase entre estos (ya sea desfasando la creación del oscilador o por el retardo producido por el sistema) tiene efectos nefastos en la recepción. En la siguiente imagen podemos ver los efectos de un offset de solo 0.2 radianes.

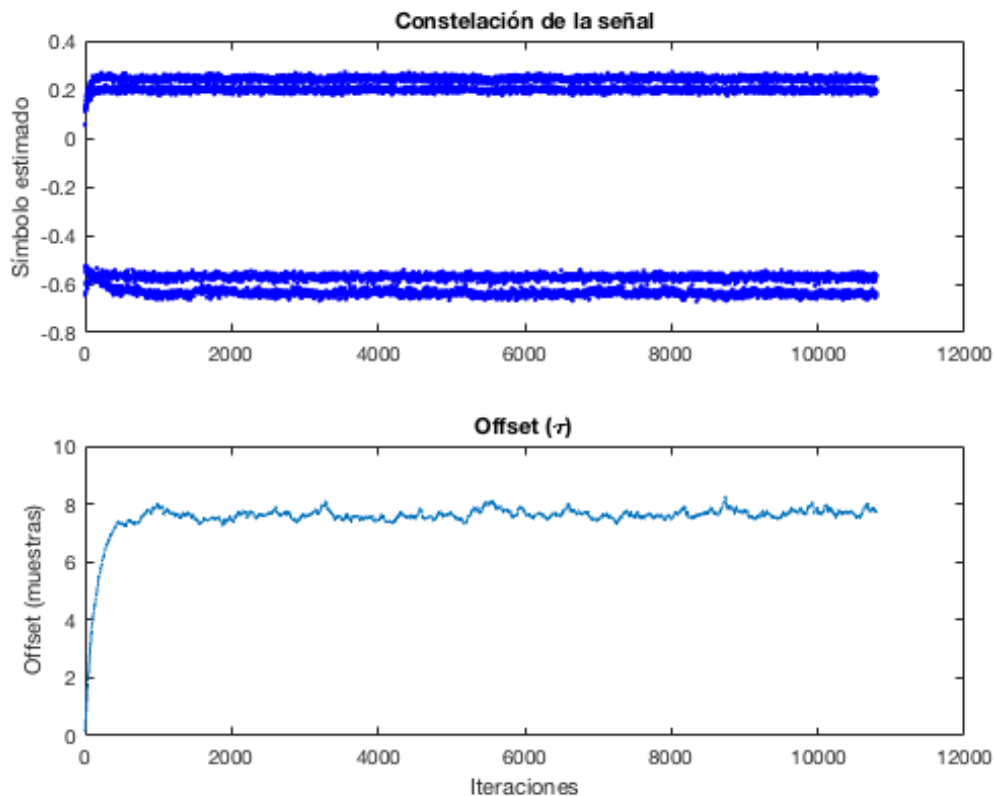


Figura 19: Desfase de 0.2 rad

Se puede ver como tiene gran efecto en la amplitud de la señal.

2.5.3. Recepción heterodina y coherente utilizando PLLs sintonizados

En primer lugar, se ha llevado a cabo una simulación considerando únicamente un ruido blanco de desviación típica $\sigma = 0.01$. Al tratarse de un ruido considerablemente bajo, el resultado obtenido ha sido satisfactorio ya que el receptor ha sido capaz de obtener una BER del 0%.

Como se ha comentado con anterioridad, los PLL tardan un tiempo en engancharse a la frecuencia. Es por esto, que es necesario transmitir cabeceras antes de transmitir información, ya que los primeros símbolos son más propensos a sufrir errores. En la siguiente figura se puede apreciar como los PLL convergen al offset de fase.

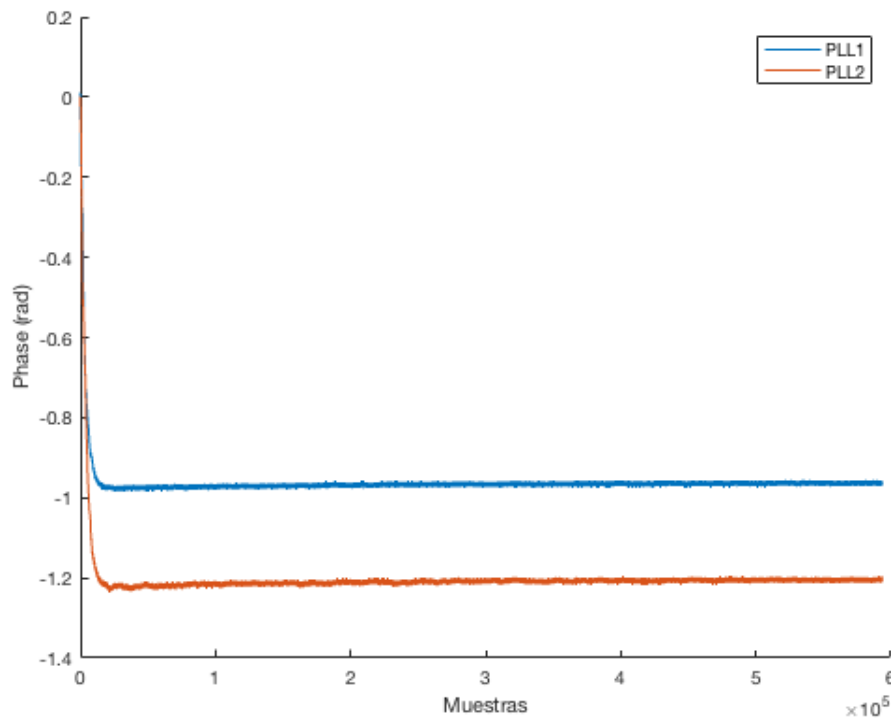
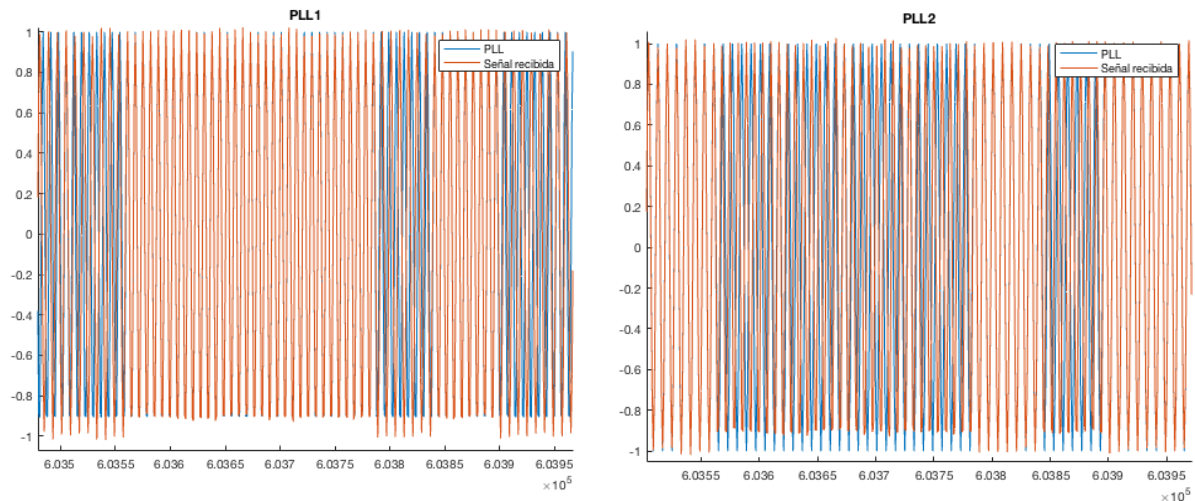


Figura 20: Convergencia de los PLL.

Se puede apreciar que se necesitan unos 400 símbolos hasta que se consigue la convergencia. En un principio se probó a crear largas secuencias de 1s o 0s, para preparar cada respectivo PLL. Sin embargo, se comprobó que con símbolos aleatorios también se alcanzaba la convergencia. Es por esto que se decidió aprovechar las tramas de sincronización para este propósito. Para obtener buenos resultados, las tramas de sincronización necesitaban unos 300 símbolos, muy cercano a los 400 necesarios del PLL, así que no supone un incremento muy significativo.

En la siguiente imagen se ve la salida de los PLL en comparación con la señal recibida. A la izquierda comparamos el PLL de f_0 con la señal recibida y en la derecha el PLL de f_1 con la señal recibida. Cuando las dos trazas se superponen, es que están enganchadas en fase. Se puede ver como cada PLL se engancha en fase cuando hay su respectivo símbolo, mientras que para el símbolo contrario no están enganchados en fase.



A continuación, se muestra la constelación de la señal recibida sin ruido ($\sigma = 0.01$) que consigue una BER de 0%.

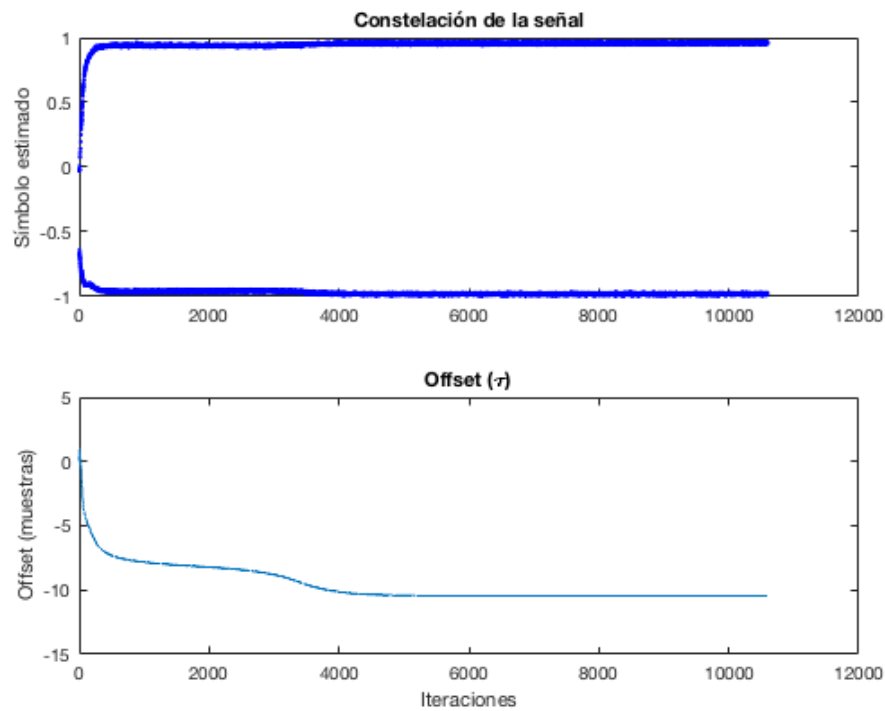


Figura 21: Receptor 3: Simulación de alta dispersión.

Se puede apreciar el tiempo de convergencia de los PLL en la constelación, así como el tiempo de convergencia del offset de muestreo.

En segundo lugar, se ha probado a aumentar la potencia de ruido de la señal recibida. Los resultados obtenidos para una desviación típica $\sigma = 1$ se muestran en la siguiente figura. Como puede comprobarse, la constelación obtenida es considerablemente más ruidosa y, en consecuencia, el BER ha aumentado su valor a 1%.

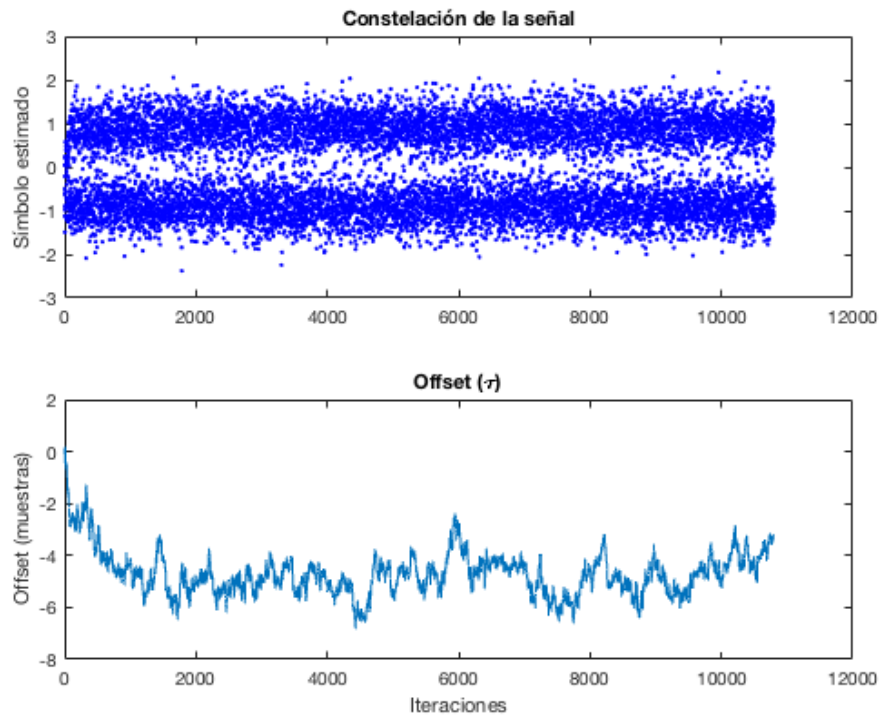
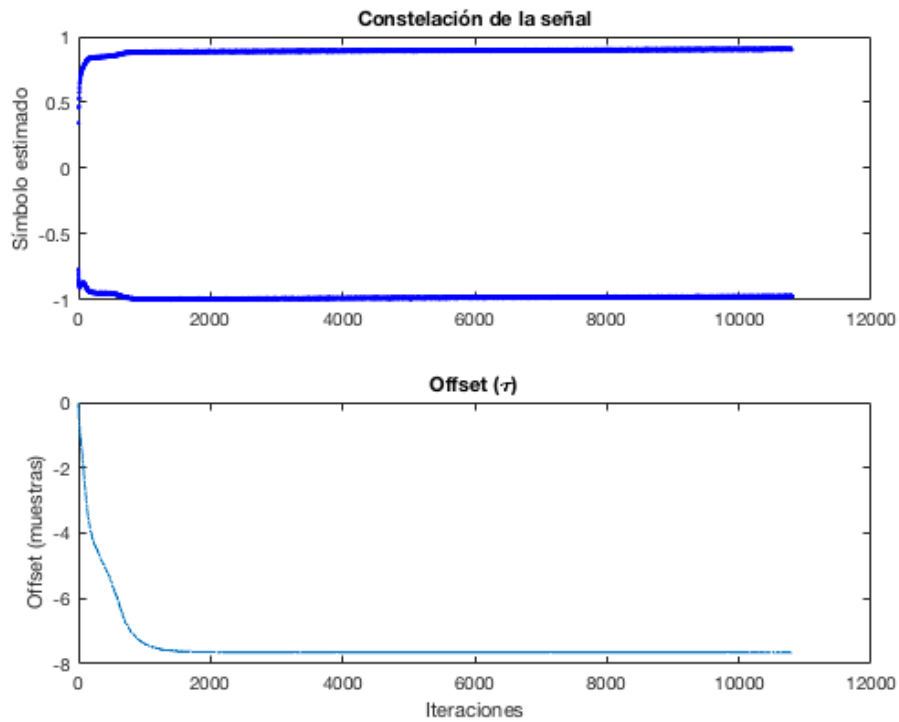


Figura 22: Receptor 1: Simulación de alto ruido.

Por último, se procede a simular sin ruido y con dispersión. Al igual que en el caso anterior, se prueba el peor caso para la dispersión: misma amplitud que la señal original y retardo de



180us.

Figura 23: Receptor 3: Simulación de alta dispersión.

Al igual que en el caso anterior, la dispersión no supone un gran problema.

3. Comunicación de espectro ensanchado

Una vez comprobado que el sistema funciona correctamente para una comunicación portadora, se tratará de extender el sistema a un entorno de comunicaciones tácticas. Para ello, la frecuencia portadora cambiará de forma pseudoaleatoria transcurrido un determinado tiempo T_h que se escogerá de manera arbitraria. La frecuencia portadora podrá tomar cualquiera de los siguientes valores:

$$f_c \in \{4.5R_b, 5.5R_b, 6.5R_b, 7.5R_b, 8.5R_b, 9.5R_b, 10.5R_b\}$$

En primer lugar, se definirá de manera arbitraria el orden de permutación de frecuencias, que será conocido tanto por el transmisor como por el receptor. Por otro lado, se definirá también el número de símbolos tras los cuales se realizará el cambio de portadora, que definirán el tiempo de salto T_h . Para los resultados mostrados en esta sección se han definido saltos de portadora tras 500 símbolos.

3.1. Transmisión

Para formar la trama de transmisión será necesario calcular el número de “bloques frecuenciales” que se van a tener. Conociendo tanto el número de símbolos a transmitir ($N_{\text{simb}, \text{total}}$) y los símbolos a transmitir por cada intervalo frecuencial ($N_{\text{simb}, \text{cambio}}$), el número de “bloques frecuenciales” se calcula redondeando hacia arriba la siguiente expresión.

$$N_{\text{bloques}} = \frac{N_{\text{simb}, \text{total}}}{N_{\text{simb}, \text{cambio}}}$$

Este número también indicará cuantos cambios en frecuencia se van a realizar. Conociendo el orden pseudoaleatorio de transmisión, el modulador generará la señal correspondiente a 500 símbolos para las frecuencias correspondientes, tras lo cual realizará un salto en frecuencia y hará lo propio para los siguientes 500 símbolos a las nuevas frecuencias de ‘0’ y ‘1’. Será importante tener en cuenta la continuidad en fase de las señales tras estos saltos de frecuencia.

3.2. Recepción

Dado que en la comunicación monoportadora se ha comprobado que el método de recepción que mejores resultados ofrece es la recepción coherente utilizando PLLs sintonizados, será la implementación escogida para el receptor de comunicaciones de espectro ensanchado.

Conociendo la secuencia de saltos de frecuencia pseudoaleatoria y el número de símbolos que se transmiten por periodo de salto (T_h), el receptor irá demodulando la señal realizando saltos en frecuencia cada 500 símbolos. Al igual que en la comunicación monoportadora, tras el proceso de demodulación se procederá a muestrear la señal, por lo que se implementará nuevamente la recuperación de reloj para hallar el instante óptimo de muestreo. Por tanto, podría decirse que la recepción se realizará de una manera análoga a la comunicación monoportadora, con la diferencia de que, tras un determinado número de muestras, se realizará un salto en frecuencia y, por tanto, se obligará al receptor a demodular la señal recibida con el valor correcto de portadora en diferentes instantes.

Para un número abundante de símbolos recibidos, se dispondrá de un número de saltos o “bloques frecuenciales” mayor a 7, por lo que los diferentes valores de portadora se utilizarán repetidas veces. Por ello, para cada valor de frecuencia portadora, será recomendable almacenar el último estado de fase estimada por el PLL para ambos símbolos (‘1’ y ‘0’) previa al

salto en frecuencia. De esta manera, cuando la portadora vuelva a realizar un salto a esta determinada frecuencia, los PLLs se inicializarán utilizando estos valores almacenados, que serán iguales o, al menos, muy próximos a la fase de la señal. De esta manera, se evitará tener que reestimar completamente la fase de la señal en cada salto, considerándola desconocida cuando, en realidad, ya se dispone de información sobre la misma.

Este mismo criterio se aplicará para el offset de la sincronización de tiempos, que ayudará a que el muestreo se realice en instantes idóneos desde el principio una vez se conozca información sobre el mismo en las diferentes frecuencias. Su efecto puede verse en el ejemplo de la Figura 24, en la que el mensaje se está transmitiendo por un canal de ruido blanco de desviación típica de 0.01. La izquierda, se muestra la primera secuencia de 500 símbolos recibida en $f_c=9.5\text{MHz}$, mientras que a la derecha se muestra la segunda secuencia recibida con esta portadora cuando, tras pasar por el resto de portadoras de manera pseudoaleatoria, vuelve a tomar este valor. Puede verse que el hecho de haber guardado el valor del offset provoca que se cometa un error menor en los primeros símbolos muestreados de la segunda secuencia en comparación con las de la primera.

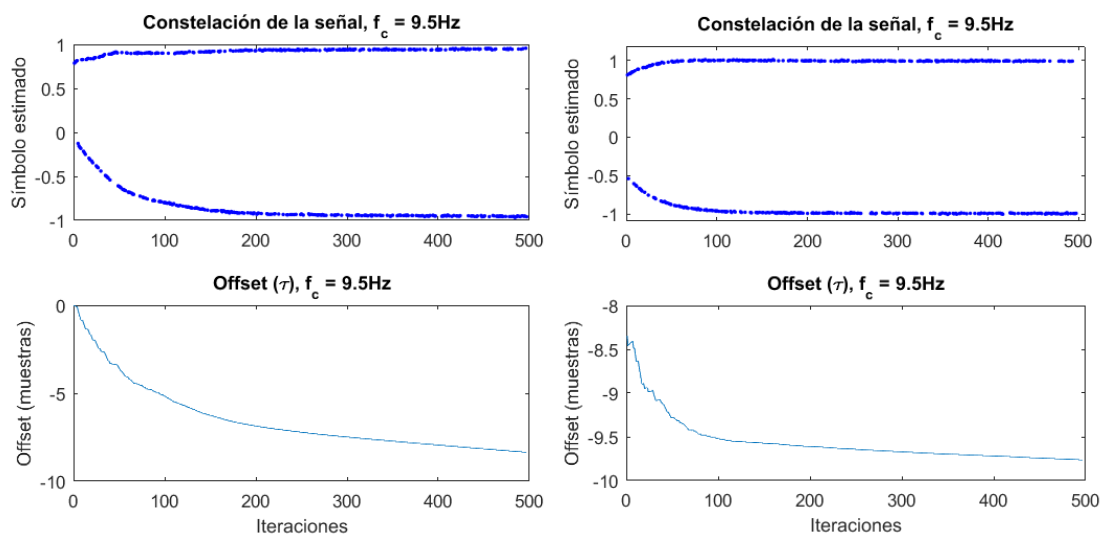


Figura 24. Primer bloque (izquierda) vs Segundo bloque (derecha) de 500 símbolos recibidos con $f_c = 9.5\text{KHz}$.

4. Conclusiones y líneas futuras

Tras realizar este sistema de transmisión y recepción, se concluye que la mejor técnica de recepción es la recepción coherente mediante PLLs. Es el receptor que soporta más potencia de ruido, así como soportar mejor la dispersión.

El trabajo pendiente en este sistema sería completar el receptor multiportadora. Pese a que se ha podido unir varias partes y demostrar que pueden trabajar juntas, no se ha llegado a implementar el sistema completo. Tampoco se han realizado pruebas con dispersión ni retardos, es decir, no hay detección del inicio ni el final de la transmisión. El sistema funciona con ruido blanco, aunque tampoco se han llegado a probar los límites de funcionamiento.

Otra funcionalidad interesante a implementar sería un control automático de ganancia. Esto ayudaría a la detección del instante de muestreo, ya que funciona mediante convergencia a los niveles prefijados de la constelación. Si estos niveles cambian, pueden dificultar la convergencia. Un CAG permitiría mantener constante estos niveles en un conjunto de situaciones más amplio.