

5.- Simulación y estudio de las prestaciones.

5.1.- INTRODUCCIÓN.

En los primeros capítulos hemos ido exponiendo la base teórica para la realización de nuestro trabajo. En el capítulo 3 estudiamos diferentes aspectos de los PLL's para concluir con una aplicación a la recuperación de portadora que es de gran importancia para nosotros, y en el capítulo anterior a éste pudimos leer información teórica sobre el algoritmo de Sari y Moridi para la resolución del problema de la recuperación de la fase instantánea de la portadora, algoritmo en el que nos basamos para realizar las simulaciones.

Vamos a realizar la simulación de la modificación de un PLL para la recuperación de la portadora propuesta por Sari y Moridi, en un receptor digital de radio (paso de banda). El diagrama de bloques que representa nuestro bucle de recuperación lo podemos ver en la figura 5.1; en esta imagen se puede observar que el bucle que vamos a simular tiene una variación con el que se describía en la figura 4.10: tras demodular la señal, no tenemos dos ramas que filtran las componentes LP en fase y cuadratura separadamente, sino que filtraremos la envolvente compleja de la señal que recibimos. La entrada de nuestro sistema será la salida del demodulador en el receptor, esto es, trataremos con un sistema en banda base en el cual las señales no tienen su espectro perfectamente centrado en el origen, sino que existirá un offset de frecuencia no nulo en el sistema.

La simulación por ordenador se realizará a partir de la implementación de los sistemas que intervienen y sus interrelaciones mediante el programa de procesamiento matemático Matlab.

Veremos dos bloques distintos: el correspondiente al estudio del sistema en bucle abierto y el correspondiente al comportamiento en bucle cerrado. Nuestros objetivos se extienden a la simulación y posterior análisis del comportamiento del sistema simulado para señales de tipo QPSK y 8-PSK en cuanto a la extensión del rango de adquisición, dejando de lado el tema de la velocidad de adquisición.

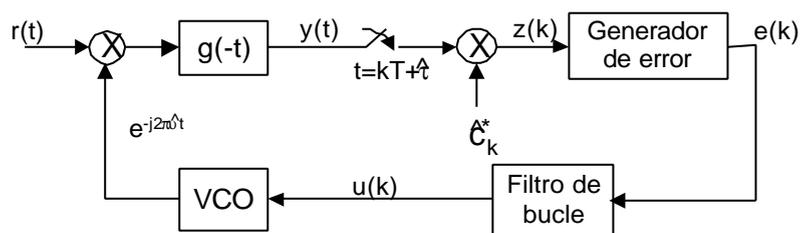


Figura 5.1.- Esquema del bucle de recuperación a realizar.

5.2.- HIPÓTESIS.

Hemos supuesto para la simulación una recuperación del reloj de símbolo perfecta, realizada en un sistema previo al nuestro. Esto simplifica mucho nuestra labor,

porque además de centrarnos únicamente en la recuperación de la frecuencia de la portadora en lugar de recuperar también la temporización, permite que supongamos:

- Que la estimación del instante de muestreo es perfecta, por lo que la decisión que tomemos sobre el símbolo que estamos recibiendo no verá incrementada su probabilidad de error por esta causa (ver figura 2.15).

- Que el desfase θ que aparece en (2.43) se puede obviar, lo que simplifica las expresiones (en adelante, entonces, $\theta = 0$)

5.3.- SISTEMA EN BUCLE ABIERTO. PRESTACIONES.

Para la simulación del sistema en bucle abierto, se han construido una serie de programas de Matlab que realizarán las operaciones de los distintos bloques constructivos del sistema de recuperación cuando el bucle no está cerrado. El estudio del sistema en bucle abierto se realiza para analizar el rango de adquisición de los distintos detectores de fase que vamos a simular; el objetivo en este punto es observar la salida del filtro de bucle para estudiar las prestaciones.

5.3.1.- REALIZACIÓN DEL SISTEMA EN BUCLE ABIERTO.

Recordaremos el esquema del receptor mostrado en la figura 2.11; este receptor lo considerábamos para la recepción en banda base, de forma que debía ir precedido de una etapa de demodulación como la que vemos en la figura 2.10, a la salida de la cual obtenemos la expresión para $r(t)$ dada por la ecuación (2.34).

Para simular la implementación del algoritmo que nos ocupa en sistemas PSK, hemos considerado que la señal se nos entrega ya demodulada, y con un cierto offset de frecuencia, es decir, la señal que recibimos obedece a una expresión dada por (2.43) más una componente de ruido. Esto es, en nuestro bucle de recuperación entrará una señal modulada a muy baja frecuencia.

Lo primero que debemos crear es, pues, una secuencia de símbolos equiprobables, que serán los emitidos por el transmisor, de naturaleza QPSK y 8-PSK. Para ello, se han creado dos bloques en Matlab, *datos.m* y *datos8.m*. Éstos se muestran, junto con todos los demás que se han realizado, en el anexo I. Aunque los datos de

salida de estos programas son de naturaleza PSK, su codificación ha sido realizada diferencialmente, por las razones que explicábamos en el punto 6 del capítulo 3.

Obtenemos entonces una secuencia codificada digitalmente según una codificación en fase cuyo alfabeto tiene 4 o bien 8 símbolos, y cuyas constelaciones son las mostradas en las figuras 2.7 y 2.8. Esa secuencia se introduce como parámetro de entrada en el módulo *pulso.m*, que la multiplica por pulsos en forma de raíz cuadrada de coseno alzado y con un periodo de símbolo T , cuyas ventajas en cuanto al efecto de la ISI ya tratamos en el capítulo 2. Obtenemos así una secuencia de datos en el ordenador que representan una señal continua en el tiempo.

Tras esto, esa secuencia es dada como parámetro de entrada al siguiente bloque, *rx.m*, que se encargará de modular a frecuencia ν (siendo ν el offset de frecuencia de la señal demodulada, que responde a la ecuación (2.42)), y de sumar un cierto ruido blanco gaussiano aditivo cuya potencia dependerá de la SNR que tengamos. Hecho esto, obtenemos a la salida de este módulo la señal continua ruidosa cuyo offset de frecuencia debemos compensar.

Esa señal ha de pasar por un bloque multiplicador, *mult.m*, que en bucle abierto no realizará ninguna acción debido a que anulamos el parámetro *nuestim* (ver anexo I); este bloque será comentado en más detalle cuando veamos la realización del sistema en bucle cerrado.

Tras esto, tenemos a la entrada del lazo directo un filtro adaptado, cuya función se realiza mediante *match.m*, cuya característica es también en forma de raíz de coseno alzado con el mismo factor de rolloff elegido para los pulsos que se transmiten, por razones que vimos en el capítulo 2. La salida corresponde a la señal $y(t)$ de la figura 2.11.

Nos encontramos ahora con una etapa de muestreo que, como ya hemos aclarado, no va a introducir ningún error adicional; su función se implementa mediante *muestreo.m*. Su salida corresponde a la señal $y(k)$ de la figura 2.11, y servirá como parámetro de entrada al decisor y al multiplicador que detecta el error de fase, bloques que describimos a continuación.

El siguiente programa es distinto según el tipo de codificación que estemos aplicando: tenemos *decisor.m*, que implementa un decisor para señales DQPSK cuyas regiones de decisión son las mostradas en la figura 2.13, y *decisor8.m*, con una función

análoga para sistemas 8-PSK diferenciales, que sigue un esquema de decisión como el de la figura 2.14. Obtendremos dos salidas en cada uno de ellos: la secuencia estimada, que en situación de ausencia de ruido y de desviación de frecuencia debe ser la misma que la secuencia PSK antes de realizar la codificación diferencial, y la secuencia detectada, secuencia de la que no hemos eliminado la codificación diferencial y en la que por lo tanto el símbolo k representa el valor nominal de la señal dentro de la región de decisión en la que se recibió $y(k)$.

La secuencia detectada de símbolos conjugada y la secuencia recibida $y(k)$, se multiplican para obtener la diferencia de fase entre ambas, que como se vio en el capítulo 3 en la sección 3.5.1 no excede de $\pi/4$ para el caso de señales QPSK (figura 3.9) y de $\pi/8$ para señales 8-PSK (figura 3.10). Esta operación se realiza en el módulo *zed.m*, el cual es el responsable de que la detección de fase tenga una característica en forma de diente de sierra.

La salida de *zed.m*, que será una secuencia cuya fase es la diferencia entre la fase recibida y la nominal de una determinada región de decisión, entra en el bloque generador de error, que será *err.m* para el caso QPSK y *err8.m* si tratamos con 8-PSK. Estos módulos emiten una característica de fase en diente de sierra *edk*, como la mostrada en las figuras 3.9, 3.10, pero truncada a partir de un cierto valor θ , tal como se veía en la figura 4.5 para un offset de frecuencia positivo y en la figura 4.6 para un offset de frecuencia negativo, con la diferencia de que en aquel caso se trataba de un detector de fase sinusoidal. La diferencia entre el bucle de recuperación de portadora basado en un decisor propuesto en la figura 3.8 y el bucle que hemos implementado nosotros para su simulación está en la inclusión de este dispositivo generador de error propuesto por Sari y Moridi. El conjunto de los bloques *decisor.m*, *zed.m* y *err.m* para el caso del receptor QPSK y de *decisor8.m*, *zed.m* y *err8.m* para el receptor 8-PSK se encarga de simular el detector de frecuencia y fase mostrado en la figura 4.10, utilizando en esta ocasión un PD en diente de sierra..

La salida del generador de error es una secuencia de fases que entra en el filtro de bucle *filtro.m*, que en la figura 4.8 llamamos $L(z)$, que realiza la media entre todas las muestras de esa secuencia que llega como parámetro de entrada para darnos su componente de continua o media. Esa media tendrá un valor $s(f_d)$, donde f_d corresponde a la expresión (3.38), con

$$\hat{\nu} = 0 \tag{5.1}$$

lo cual corresponde en nuestra realización a decir que el parámetro de salida del módulo que implementa el VCO, *nuestim*, se anula, ya que estamos viendo el sistema en bucle abierto. Entonces,

$$f_d = \nu \tag{5.2}$$

esto es, cuando el bucle no está cerrado el offset de frecuencia será en todo momento igual al inicial, dado por el parámetro *nu*.

El esquema de todo el sistema en bucle abierto se puede ver representado en las figuras 5.2 y 5.3, para sistemas QPSK y 8-PSK, respectivamente. El objetivo de este esquema es la consecución de la mencionada curva $S(f_d)$, para discutir los valores obtenidos y compararlos con los dados por los autores del artículo en el cual nos estamos basando.

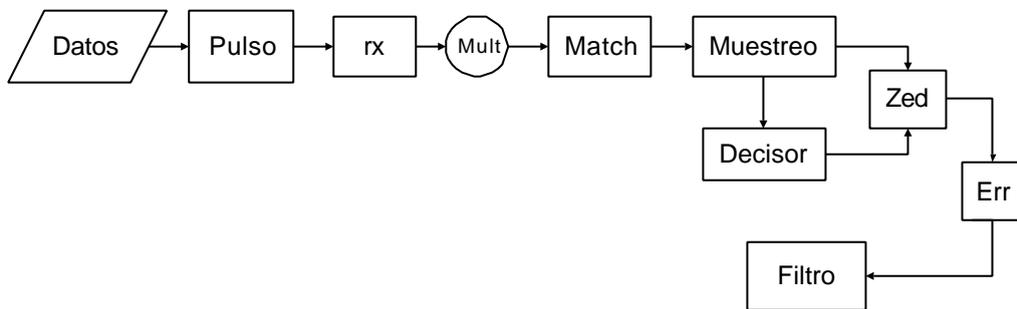


Figura 5.2.- Diagrama de los bloques empleados para el caso QPSK en bucle abierto.

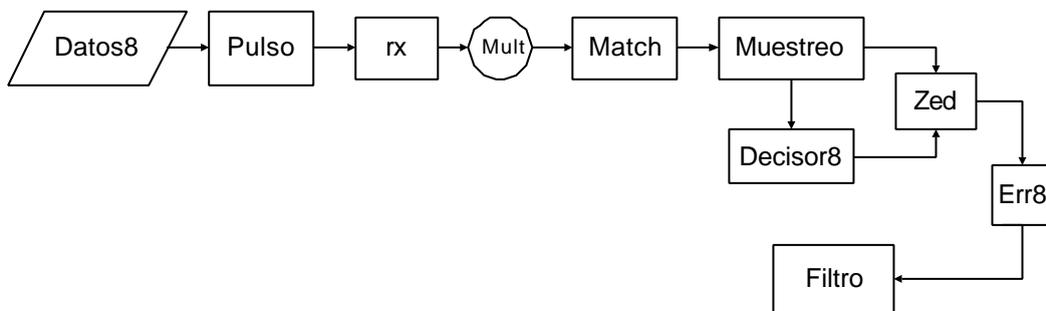


Figura 5.3.- Diagrama de los bloques empleados para el caso 8-PSK en bucle abierto.

Pasemos ahora al análisis de los resultados.

5.3.2.- SIMULACIÓN DEL SISTEMA EN BUCLE ABIERTO. ESTUDIO DE LAS PRESTACIONES.

Si emitimos una secuencia de símbolos QPSK con el módulo *datos.m*, la constelación que se genera es igual a la de la figura 2.3; en el caso de generar una secuencia con *datos8.m*, se obtiene una constelación como en la figura 2.4; estos resultados con Matlab se refleja en la figura 5.4; para esta simulación se ha escogido una longitud de la secuencia suficientemente larga como para que todos los distintos símbolos equiprobables generados aleatoriamente puedan aparecer.

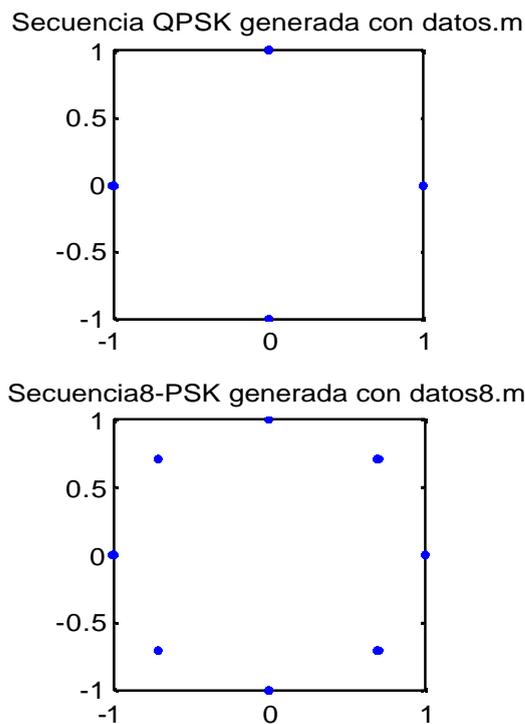


Figura 5.4.- Salida de *datos.m* y de *datos8.m*.

El pulso que obtenemos mediante *pulso.m* es la raíz cuadrada del que mostramos en la figura 2.8 para distintos valores del factor de caída, el cual se corresponde con el parámetro *rolloff*. La forma que obtenemos es muy parecida, como se puede apreciar en

la figura 5.5, donde la curva azul corresponde a $rolloff = 1$, la verde a $rolloff = 0.5$ y la roja a $rolloff = 0$.

El valor del factor de caída que utilizaremos para nuestras simulaciones es el que se utilizó en las pruebas que se describieron en el capítulo anterior: $rolloff = 0.5$.

Nos interesa fijarnos en los símbolos que recibimos en determinadas condiciones de ruido y de offset de frecuencia. Supongamos primero que la desviación de frecuencia es nula; entonces, obtendremos en el mapa de la constelación distintos puntos alrededor de los nominales que estarán más desplazados o menos según el nivel de ruido. En las figuras 5.6 y 5.7 mostramos, en estas condiciones de sincronización perfecta, cuales son los símbolos que recibimos (los valores que toma la secuencia $y(k)$) en distintas condiciones de ruido, para los sistemas QPSK y 8-PSK respectivamente. La longitud de la secuencia de símbolos es de 200. Los valores de la relación señal a ruido para los que se representa la señal recibida muestreada son de 10, 20, 30 y 60 dB.

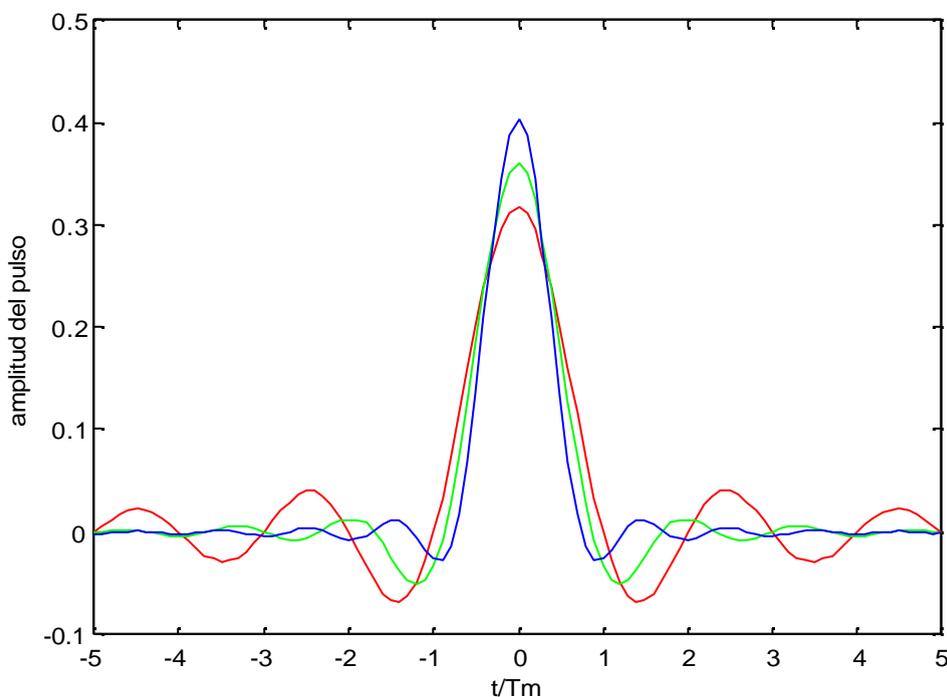


Figura 5.5.- Pulso raíz de coseno alzado normalizado con respecto al periodo de muestreo T_m , para distintos valores del parámetro $rolloff$.

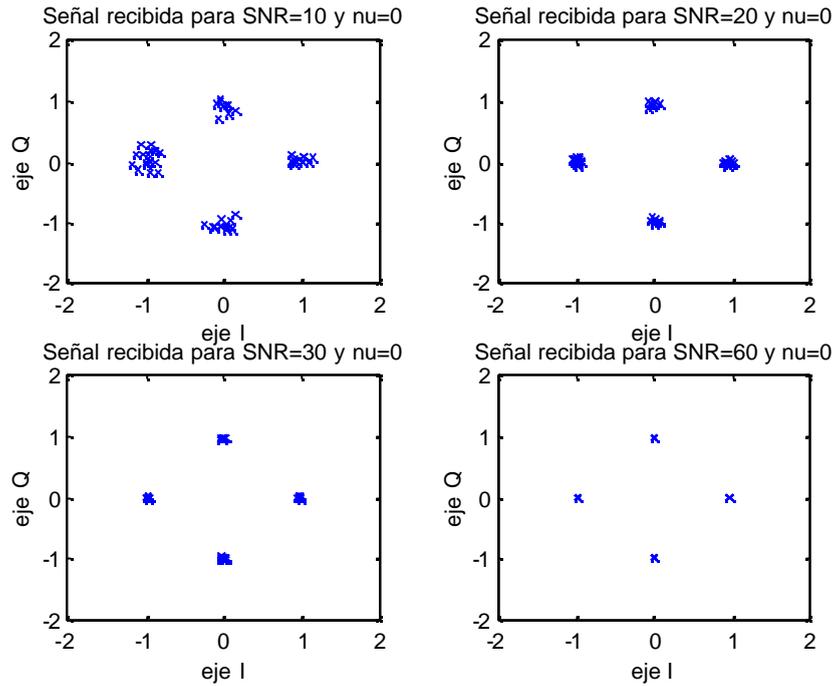


Figura 5.6.- Símbolos recibidos para una desviación de frecuencia nula y distintos valores de relación señal a ruido en un sistema QPSK.

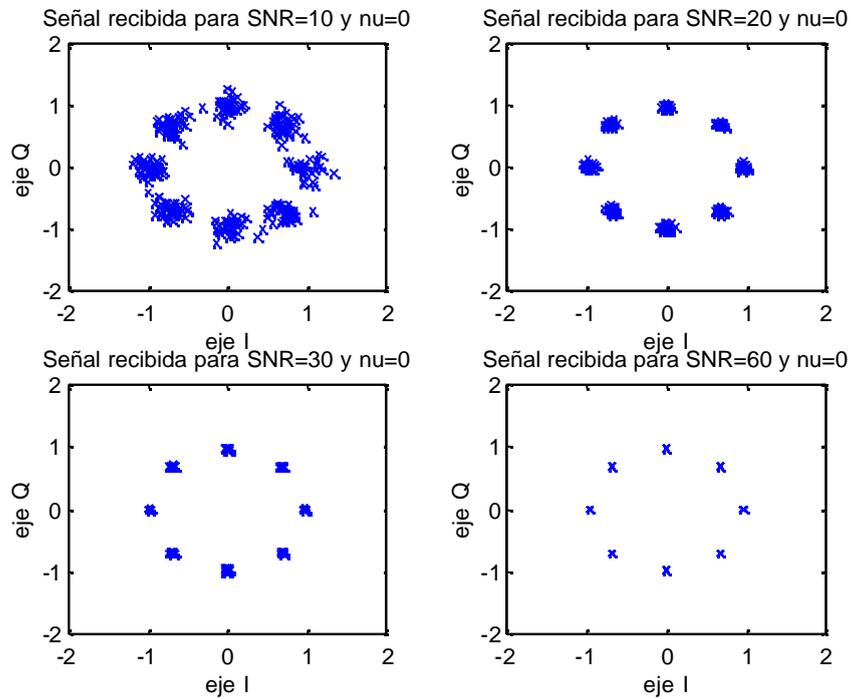


Figura 5.7.- Símbolos recibidos para una desviación de frecuencia nula y distintos valores de relación señal a ruido en un sistema 8-PSK.

Como vemos, la señal que recibimos es tanto más difusa cuanto menor sea la relación SNR en la entrada del sistema, no mostrando movimiento apreciable respecto de los puntos nominales cuando la SNR = 60 dB, ya que ésta es una situación en la que a efectos prácticos se está libre de ruido. Esos desplazamientos de sus posiciones originales se traducen en nuestro dispositivo en desviaciones en la fase. Recordemos que el generador de error tiene la misión de detectar el desfase existente entre la señal recibida y la estimada por el detector; este generador pensará que esa desviación de fase proviene de un offset de frecuencia, lo cual sabemos que en este caso no es cierto, por lo cual dará una salida distinta de cero y el filtro del bucle hará una media que resultará ser no nula, si bien será más pequeña cuanto mayor sea la longitud de la secuencia recibida dada la naturaleza aleatoria del ruido a la entrada del receptor y, por tanto, de los desfases que produce; para los casos anteriores, las salidas de los generadores de error son las mostradas a continuación, para tres valores distintos (0.25, 0.5 y 0.75) del parámetro β que los caracteriza, siendo β un número comprendido entre 0 y 1 de forma que $\theta = \beta \cdot \pi/4$ en el caso del detector de fase QPSK y $\theta = \beta \cdot \pi/8$ en el caso del detector de fase 8-PSK. En primer lugar, en la figura 5.8 calculamos la salida del generador de error para el sistema QPSK cuando $beta = 0.25$; en la figura 5.9 mostraremos a continuación esa misma salida cuando $beta = 0.5$, por último, en la figura 5.10 veremos cuál es la salida del generador de error cuando $beta = 0.75$. Todo esto, para los valores de relación señal a ruido SNR=10, SNR=20, SNR=30 y SNR=60. En las tres figuras suponemos que el offset de frecuencia es nulo.

Estos tres valores del parámetro $beta$ los utilizaremos para hacer las distintas pruebas en bucle abierto; según los resultados que arrojen éstas elegiremos un valor u otro para el sistema en bucle cerrado.

Se ha calculado la media de estas secuencias y de las obtenidas con los otros generadores de error con el filtro del bucle; los resultados se muestran en la tabla 5.1.

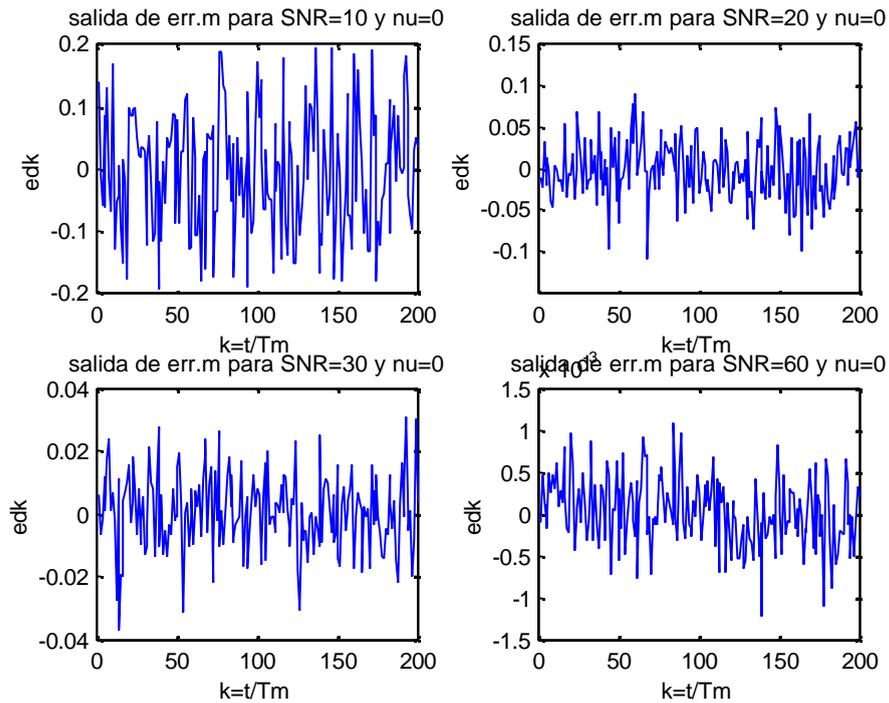


Figura 5.8.- Salida del generador de error con $\beta=0.25$ en el sistema QPSK para distintos valores de la relación señal a ruido y offset de frecuencia nulo.

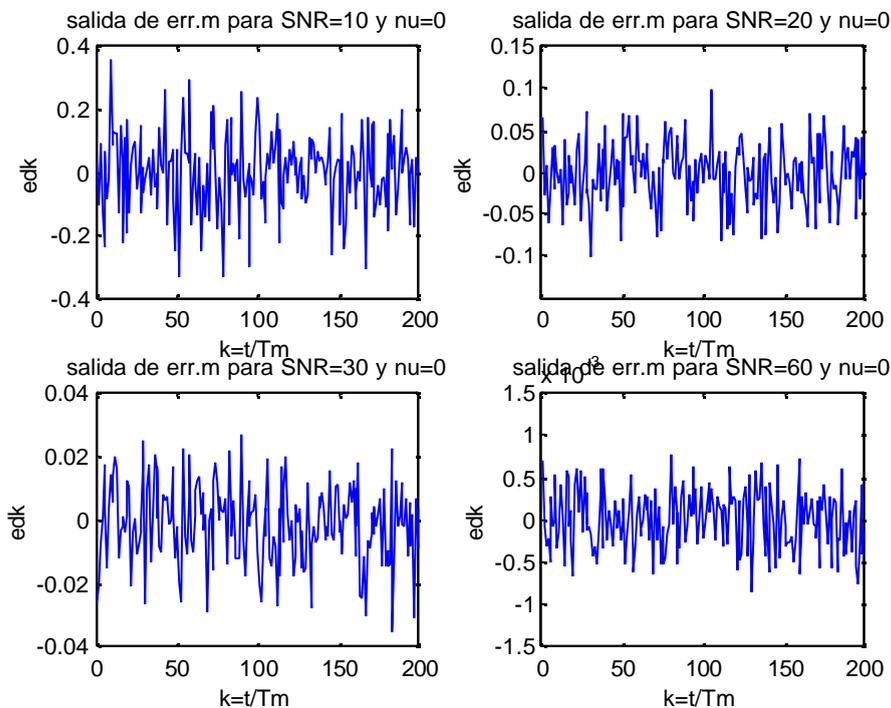


Figura 5.9.- Salida del generador de error con $\beta=0.5$ en el sistema QPSK para distintos valores de la relación señal a ruido y offset de frecuencia nulo.

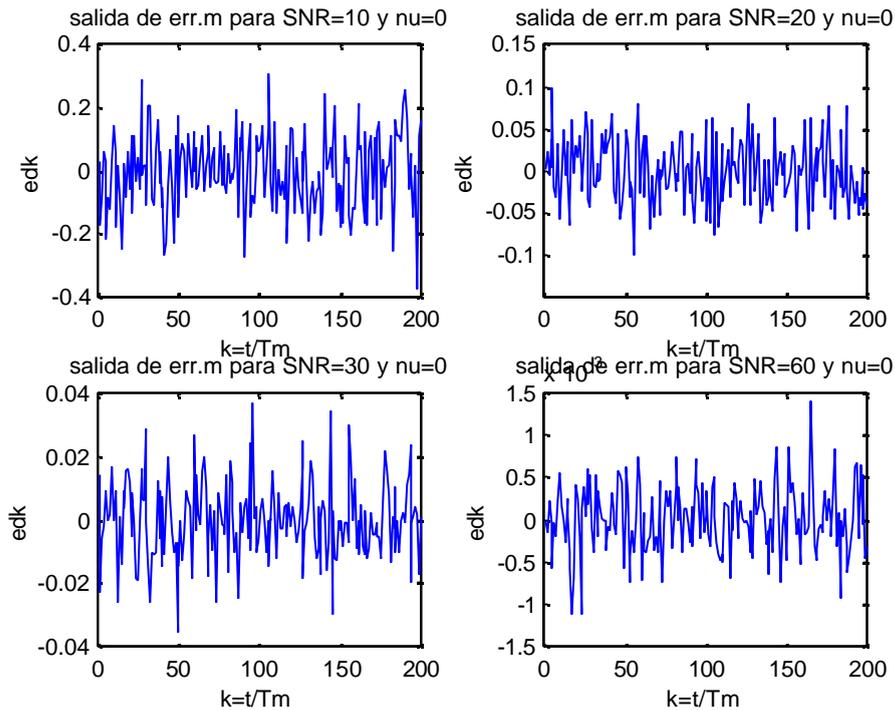


Figura 5.10.- Salida del generador de error con $\beta=0.75$ en el sistema QPSK para distintos valores de la relación señal a ruido y offset de frecuencia nulo.

	SNR=10	SNR=20	SNR=30	SNR=60
$\beta=0.25$	-0.0093	-0.0012	-0.0016	-2.5621e-005
$\beta=0.5$	-0.0076	-0.0067	-9.9104e-004	-6.4975e-006
$\beta=0.75$	-0.0078	-0.0015	-3.5312e-004	4.6309e-007

Tabla 5.1.- Medias obtenidas por el filtro del bucle en el sistema QPSK con las distintas SNR y los distintos generadores de error que estamos considerando, para $\nu=0$.

En cuanto al sistema 8-PSK, la salida del generador de error para cada SNR especificada se muestra en las figuras 5.11 ($\beta = 0.25$), 5.12 ($\beta = 0.5$) y 5.13 ($\beta = 0.75$), para una desviación de frecuencia nula. Las respectivas salidas del filtro de media aparecen en la tabla 5.2.

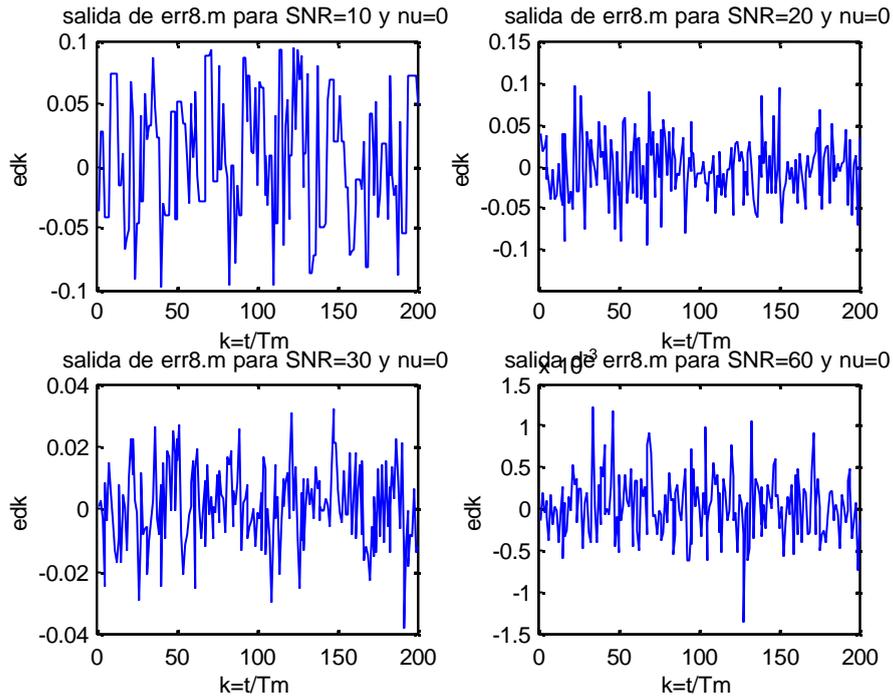


Figura 5.11.- Salida del generador de error con $\beta=0.25$ en el sistema 8PSK para distintos valores de la relación señal a ruido y offset de frecuencia nulo.

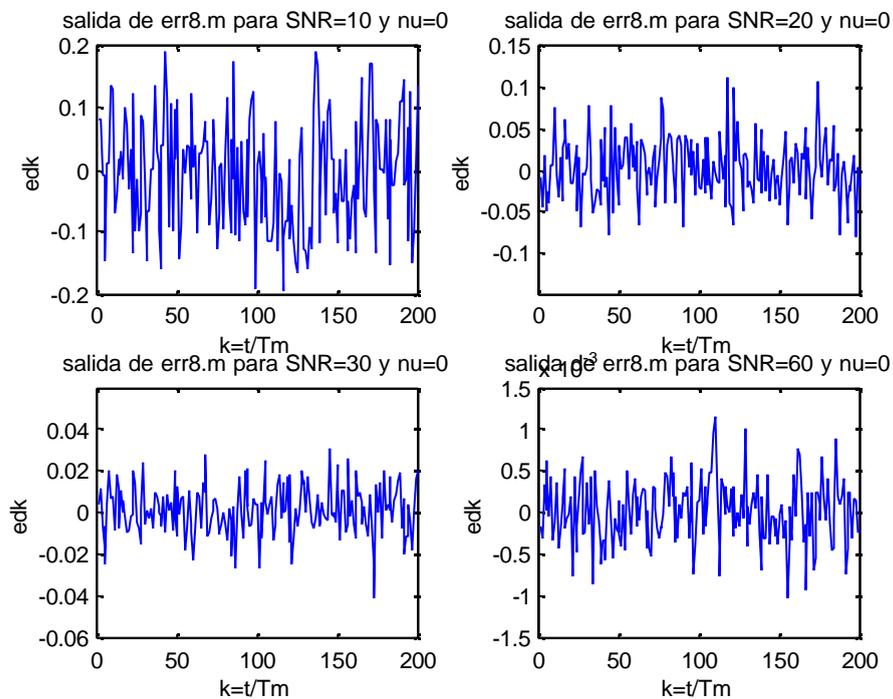


Figura 5.12.- Salida del generador de error con $\beta=0.5$ en el sistema 8PSK para distintos valores de la relación señal a ruido y offset de frecuencia nulo.

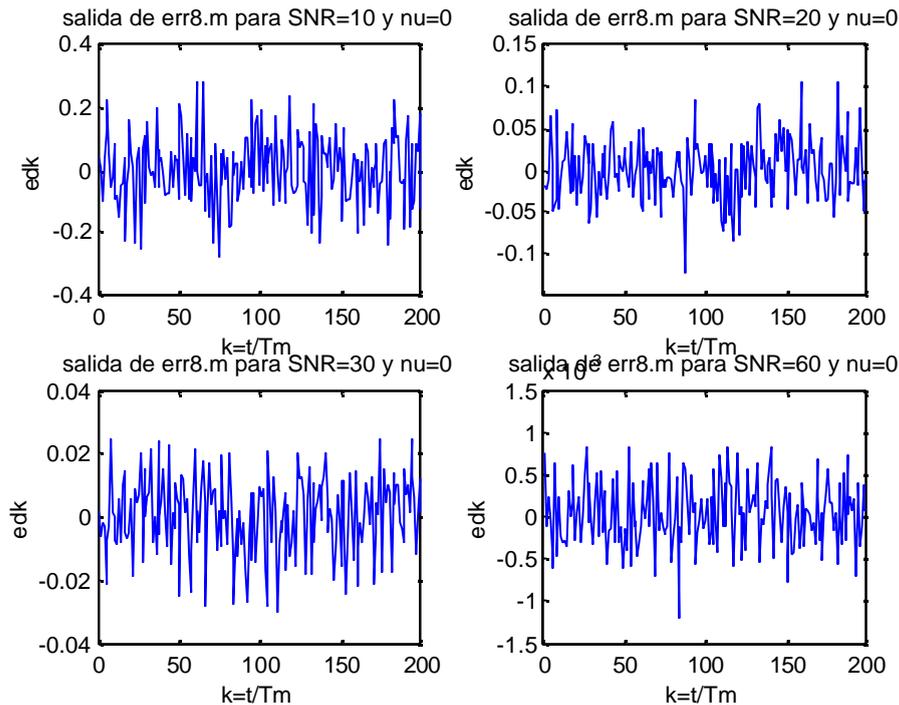


Figura 5.13.- Salida del generador de error con $\beta=0.75$ en el sistema 8PSK para distintos valores de la relación señal a ruido y offset de frecuencia nulo.

	SNR=10	SNR=20	SNR=30	SNR=60
$\beta=0.25$	-0.0068	0.0021	-0.0015	2.5064e-006
$\beta=0.5$	-0.0119	-0.0022	0.0013	-2.5734e-005
$\beta=0.75$	-0.0129	2.2866e-004	6.5591e-004	1.4564e-005

Tabla 5.2.- Medias obtenidas por el filtro del bucle en el sistema 8-PSK con las distintas SNR y los distintos generadores de error que estamos considerando, para $\nu=0$.

Lo primero que se observa es que cuando no existe offset de frecuencia la señal de salida del detector de frecuencia y fase oscila (aleatoriamente, a causa del ruido) alrededor de cero, de forma que la componente de continua debe aproximarse a cero. La presencia del ruido da lugar a secuencias de fases aleatorias, cuyas muestras serán mayores o menores según sea mayor o menor la potencia de ruido que interfiere con la señal; la media que se obtiene es más cercana a cero cuanto mayor es la relación señal a ruido, aunque no se cumple estrictamente en las simulaciones anteriores según se

observa en las tablas 5.1 y 5.2, donde vemos que podemos obtener medias de igual orden o incluso mayores para secuencias donde hay una menor potencia de ruido. Esto se debe al carácter aleatorio de la señal de ruido; ya hemos comentado que, por esta razón, con una secuencia más larga de símbolos deberíamos obtener medias más cercanas a cero, que es el número que debería medir, en ausencia de ruido, el filtro de bucle. En la tabla 5.3 mostraremos, para distintas SNR y distintas longitudes de secuencia, la salida del filtro del bucle en el sistema QPSK cuando el valor del parámetro *beta* del generador de error es igual a 0.5; en la tabla 5.4 se pueden ver los resultados de una simulación análoga realizada sobre el sistema 8-PSK. Se observa que el resultado no se llega a anular del todo, si bien se obtienen magnitudes muy bajas. Por lo demás, estos resultados no arrojan una información significativa.

	SNR=10	SNR=20	SNR=30	SNR=60
200 símbolos	-0.0076	-0.0067	-9.9104e-004	-6.4975e-006
1000 símbolos	0.0034	-8.0745e-004	-9.5320e-004	-2.6759e-006
5000 símbolos	-0.0025	5.5394e-004	-2.8058e-004	2.0426e-006
15000 símbolos	0.0011	2.5779e-004	8.9512e-005	1.9516e-006

Tabla 5.3.- Media obtenida por el filtro de bucle en sistema QPSK para secuencias de distinta longitud, con $nu = 0$

	SNR=10	SNR=20	SNR=30	SNR=60
200 símbolos	-0.0119	-0.0022	0.0013	-2.5734e-005
1000 símbolos	-0.0053	0.0013	-5.7380e-004	-2.6759e-006
5000 símbolos	0.0060	0.0026	2.7042e-004	2.0426e-006
15000 símbolos	0.0031	5.5394e-004	1.7333e-004	-4.0122e-007

Tabla 5.4.- Media obtenida por el filtro de bucle en sistema 8-PSK para secuencias de distinta longitud, con $nu = 0$

Hasta ahora hemos comentado el comportamiento en bucle abierto del detector cuando no existe desviación de frecuencia; sin embargo, como venimos diciendo desde

el principio, lo utópico de esa situación es precisamente lo que nos lleva a implementar nuestro sistema en el receptor, es decir, esa situación ideal no es lo que nos vamos a encontrar. En lugar de esto, el receptor no estará totalmente sincronizado con el transmisor y será necesaria la actuación de nuestro bucle de recuperación para poder acceder a los datos correctamente tras la demodulación. La decodificación correcta de los datos será posible sólo si nos encontramos dentro de un rango de adquisición, esto es, existe un límite para el offset de frecuencia a partir del cual el nuestro sistema no funcionará.

Supongamos entonces que la señal demodulada llega al PFD afectada por un cierto offset de frecuencia; la señal muestreada $y(k)$ ya no es como en las figuras 5.6 y 5.7, sino que aparece una rotación debido al incremento de fase, producido por ese offset, dado por $\Delta\varphi = 2\pi f_d T_m$. (ya que $\varphi(t) = 2\pi f_d t + \varphi(0)|_{t=k \cdot T_m} = 2\pi f_d k T_m + \varphi(0) = \Delta\varphi \cdot k + \varphi(0)$, según lo que vimos en el apartado 4.2). Para ilustrar esto podemos recordar la figura 2.17.

La desviación de frecuencia se medirá en términos porcentuales del periodo de símbolo (dado que siempre aparecen estos dos términos en un producto); veamos a continuación gráficas que representan la salida del detector de frecuencia y fase, para unos valores de la SNR de 10, 20, 30 y 60 dB, cuando la señal entra en el PFD con distintas desviaciones de frecuencia tanto en sistemas QPSK como en sistemas 8-PSK. Aquí ya podremos observar la forma de diente de sierra truncado según el valor de β que elijamos. En las figuras 5.14, 5.15 y 5.16 vemos la salida del generador de error en una comunicación QPSK cuando la desviación de frecuencia es pequeña, de un 1% (es decir, $T_m \cdot f_d = 0.01$, para β 0.25, 0.5 y 0.75 respectivamente. En las figuras 5.17, 5.18 y 5.19 obtenemos análogos resultados para otro valor de la desviación de frecuencia, 4%. Repetimos el proceso para un offset de frecuencia del 7% y mostramos los resultados obtenidos en las figuras 5.20, 5.21 y 5.22; por último, las figuras 5.23, 5.24 y 5.25 muestran la salida del generador de error para los valores de β mencionados cuando el offset alcanza un valor del 10%. Realizamos las pruebas ahora para sistemas 8-PSK, mostrando en las figuras 5.26, 5.27 y 5.28 las salidas de los tres generadores de error que estamos testeando cuando existe un offset del 1%, en las figuras 5.29, 5.30 y 5.31 las mismas salidas para una desviación del 2.5% y en las figuras 5.32, 5.33 y 5.34 lo mismo para un offset de frecuencia del 5%.