

MATLAB: Interfaces Gráficas de Usuario destinadas al estudio de señales Radar y GNSS

Trabajo Fin de Carrera

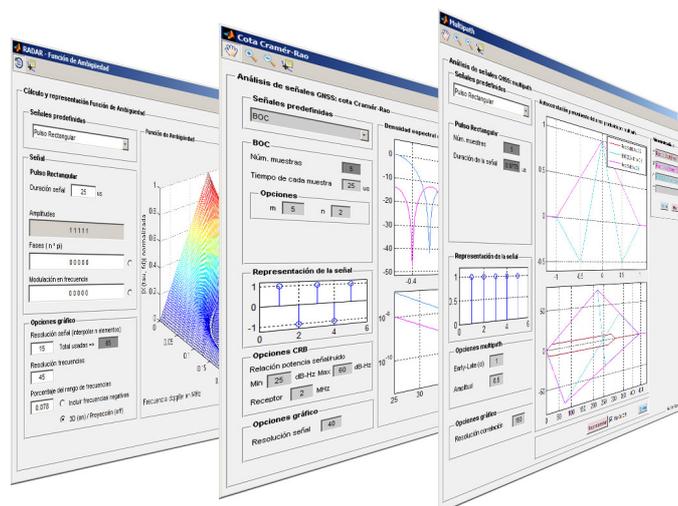
16 de junio 2011

Autor: Patrick Tamellini

Consultor: D. José Antonio López Salcedo



Universitat Oberta de Catalunya



Agradecimientos

Muchas personas han influido en mi formación académica y como consecuencia y de forma inevitable en mi formación personal. Quiero mencionar a D. Jesús Carretero Rubio que durante el bachillerato me enseñó física y el pensamiento racional así como técnicas de trabajo que sigo aplicando hoy en día, a D. Albert Gras Martí, que durante mis primeros meses en la UOC fue motivador increíble y con el que comparto el amor por las *baldufas* invertidas, y por supuesto a mi consultor y asesor para este proyecto, D. José Antonio López Salcedo por su generosidad en las explicaciones combinando rigurosidad con lenguaje claro. No me olvido de mis padres, Ingrid y Félix, por su infinita paciencia y su apoyo continuo. Sin ellos nada hubiera sido posible.

“La mente no es una vasija para llenar sino un fuego que hay que prender”.- Plutarco

Palabras Clave

Autocorrelación – Barker – BOC – CBOC – Coseno Alzado – Cramér-Rao – Función de Ambigüedad – Discriminador – Densidad Espectral en Potencia – Galileo – GNSS – GPS – Interfaz de Gráfica de Usuario – MATLAB – *multipath* – Navegación – Periodograma – Radar – Satélite – Señales Binarias

Resumen

Los sistemas de radar y los equipos de navegación globales basados en satélites (GNSS) tienen cada vez más presencia en nuestra vida cotidiana. Ya sea para guiar a las aeronaves que nos llevan a lugares remotos como para orientarnos cuando paseamos por las calles de una ciudad. Independientemente del uso final de los sistemas, la precisión en el posicionamiento es imperativa. Impulsado por ésta, el presente trabajo trata de las funciones y estimaciones que nos permiten realizar un análisis de las señales que forman parte integral de estos sistemas, siendo el fin último establecer y acotar los errores que se producen al usar un tipo de señal u otro. Serán objeto de estudio el funcionamiento del procesador de señales de un sistema radar y el concepto de Función de Ambigüedad. En el campo de los GNSS aplicaremos la Cota de Cramér-Rao para establecer una cota inferior en la varianza de la estimación de la posición. Así mismo se verá en que magnitud afectan al posicionamiento suministrado por un sistema GNSS la presencia de señales *multipath* en un receptor.

Con el fin de darle forma y una utilidad práctica, se usará el entorno de trabajo MATLAB para desarrollar tres aplicaciones con sus correspondientes interfaces gráficas de usuario, las cuales forman parte integral del trabajo. Se acompañarán los pertinentes manuales de usuario para que puedan ser empleadas por terceros.

Índice de Contenidos

PALABRAS CLAVE	I
RESUMEN.....	I
ÍNDICE DE CONTENIDOS.....	III
ÍNDICE DE FIGURAS.....	VII
ÍNDICE DE TABLAS.....	IX
LISTA DE ACRÓNIMOS.....	XI
1. INTRODUCCIÓN.....	1
1.1 MOTIVACIONES	1
1.2 OBJETIVOS	2
1.3 ENFOQUE Y METODOLOGÍA.....	3
1.3.1 ENTREGABLES	4
1.3.2 DIAGRAMA DE TAREAS.....	4
1.4 ESBOZO DE ESTA MEMORIA.....	5
2. FUNDAMENTOS TEÓRICOS.....	7
2.1 LOS SISTEMAS DE RADAR.....	7
2.1.1 CONCEPTOS BÁSICOS DEL RADAR	7
2.1.2 FRECUENCIA <i>DOPPLER</i>	8
2.1.3 FUNCIÓN DE AMBIGÜEDAD	9
2.2 LOS SISTEMAS DE NAVEGACIÓN GLOBALES POR SATÉLITE	11
2.2.1 CONCEPTOS BÁSICOS DE GNSS.....	11
2.2.2 ESTIMACIÓN DEL ERROR EN EL RETARDO.....	12
2.2.2.1 Densidad espectral en potencia	12
2.2.2.2 Cota de Cramér-Rao	13
2.2.3 SEGUIMIENTO Y SINCRONÍA CON LAS SEÑALES	14
2.2.3.1 Adquisición y seguimiento	14
2.2.3.2 El discriminador	15
2.2.3.3 El problema del <i>multipath</i>	17
2.3 SEÑALES.....	20
2.3.1 PULSO RECTANGULAR	20
2.3.2 <i>SQUARE-ROOT RAISED-COSINE</i>	20

2.3.3	<i>BINARY OFFSET CARRIER</i>	22
2.3.4	<i>COMPOSITE BINARY OFFSET CARRIER</i>	23
2.3.5	SEÑALES BASADAS EN CÓDIGOS BINARIOS	24
2.3.5.1	Códigos Barker	25
3.	<u>MATLAB Y LAS INTERFACES GRÁFICAS DE USUARIO</u>	27
3.1	EL ENTORNO DE TRABAJO	27
3.2	EL LENGUAJE DE PROGRAMACIÓN MATLAB.....	28
3.3	INTERFACES GRÁFICAS DE USUARIO.....	29
3.3.1	USO DE GUIDE.....	29
3.3.2	FICHERO .M DE GUIDE	31
4.	<u>IMPLEMENTACIONES Y ALGORITMOS.....</u>	33
4.1	IMPLEMENTACIÓN FUNCIÓN DE AMBIGÜEDAD	33
4.2	IMPLEMENTACIÓN COTA CRAMÉR-RAO	35
4.2.1	ELEMENTOS PRESENTES	36
4.2.2	FUNCIONES EMPLEADAS	36
4.3	IMPLEMENTACIÓN <i>MULTIPATH</i>	37
4.3.1	ELEMENTOS PRESENTES	37
4.3.2	FUNCIONES EMPLEADAS	38
5.	<u>CONCLUSIONES.....</u>	41
5.1	TRABAJO FUTURO	42
6.	<u>ANEXO: MANUALES DE USUARIO</u>	43
6.1	INICIO RÁPIDO. APLICACIÓN ‘LANZADOR’	43
6.2	APLICACIÓN FUNCIÓN DE AMBIGÜEDAD	43
6.2.1	DESCRIPCIÓN DE LA PANTALLA	43
6.2.2	DESCRIPCIÓN DE LAS ZONAS	44
6.2.2.1	Selector de señales predefinidas	45
6.2.2.2	Características de la señal	45
6.2.2.3	Opciones gráficas.....	47
6.2.2.4	Área de la gráfica y botón ‘Representar’	48
6.2.2.5	Herramientas para manejar la gráfica.....	49
6.3	APLICACIÓN COTA CRAMÉR-RAO.....	50
6.3.1	DESCRIPCIÓN DE LA PANTALLA	50

6.3.2	USO DEL PROGRAMA	51
6.3.2.1	Selección de la señal	51
6.3.2.2	Opciones CRB.....	52
6.3.2.3	Opciones gráficas.....	52
6.3.2.4	Representación gráfica.....	52
6.3.2.5	El almacén de señales	53
6.3.2.6	Herramientas para manejar la gráfica.....	54
6.3.3	INTERPRETACIÓN DE LOS RESULTADOS	54
6.4	APLICACIÓN <i>MULTIPATH</i>	55
6.4.1	DESCRIPCIÓN DE LA PANTALLA	55
6.4.2	USO DEL PROGRAMA	56
6.4.2.1	Selección de la señal	56
6.4.2.2	Opciones <i>multipath</i> y correlador	57
6.4.2.3	Opciones gráficas.....	57
6.4.2.4	Representación gráfica.....	58
6.4.2.5	El almacén de señales	58
6.4.2.6	Herramienta de manipulación	59
6.4.3	INTERPRETACIÓN DE LOS RESULTADOS	59
6.5	LISTA DE FICHEROS NECESARIOS PARA CORRER LAS APLICACIONES.....	61
6.6	FAQS: PREGUNTAS FRECUENTES	61
6.6.1	FAQS FUNCIÓN DE AMBIGÜEDAD.....	61
6.6.2	FAQS COTA CRAMÉR-RAO	62
6.6.3	FAQS <i>MULTIPATH</i>	63
<u>BIBLIOGRAFÍA.....</u>		65

Índice de Figuras

FIGURA 1: PLAN DE TRABAJO A ALTO NIVEL.....	4
FIGURA 2: PLAN DE TRABAJO A BAJO NIVEL.....	4
FIGURA 3: OBSTÁCULO ACERCÁNDOSE AL SISTEMA RADAR.....	8
FIGURA 4: ESQUEMA DE UN PROCESADOR DE SEÑALES DE UN SISTEMA RADAR.....	9
FIGURA 5: ESQUEMA DE UN RECEPTOR GNSS.....	14
FIGURA 6: BLOQUE DE ADQUISICIÓN EN UN RECEPTOR GNSS.	15
FIGURA 7: RESULTADO DE UNA BÚSQUEDA EN SERIE DONDE, PARA UN PRN DADO, HAY UNA SEÑAL GNSS PRESENTE [D. PLAUSINAITIS, 2009].....	15
FIGURA 8: PUNTOS <i>EARLY</i> Y <i>LATE</i> EN UN DISCRIMINADOR DE CÓDIGO.	16
FIGURA 9: CURVA DEL DISCRIMINADOR. SEPARACIÓN E-L UN <i>CHIP</i>	16
FIGURA 10: REPRESENTACIÓN GRÁFICA DEL FENÓMENO <i>MULTIPATH</i>	17
FIGURA 11: ERROR DE LA CURVA DEL DISCRIMINADOR POR LA PRESENCIA DE UN <i>MULTIPATH</i> . DESFASE 0°. SEÑAL PULSO RECTANGULAR.	18
FIGURA 12: ERROR DE LA CURVA DEL DISCRIMINADOR POR LA PRESENCIA DE UN <i>MULTIPATH</i> . DESFASE 180°. SEÑAL PULSO RECTANGULAR.....	18
FIGURA 13: ENVOLVENTE DE ERROR DEBIDO A <i>MULTIPATH</i> PARA LAS SEÑALES PULSO RECTANGULAR Y BOC(2,2).	19
FIGURA 14: MÚLTIPLES PUNTOS DE CORTE CON EL EJE TEMPORAL PARA UNA SEÑAL BOC(1,1).	19
FIGURA 15: AUTOCORRELACIÓN DE PULSO RECTANGULAR, CBOC(6,1,+) CBOC(6,1,-).....	20
FIGURA 16: SEÑAL SQRRRC CON DIFERENTES VALORES DE B. ARRIBA DOMINIO FRECUENCIAL, ABAJO TEMPORAL [M. ZOLTOWSKY, 2010].....	21
FIGURA 17: LA SINUSOIDE MODULADORA TOMADA SÓLO EN SU SIGNO. T_c ES EL PERIODO DE UN <i>CHIP</i> , EN ESTE CASO, BOC(5,2).	23
FIGURA 18: CÓDIGO MODULADO POR UN <i>CHIP</i> BOC(5,2).....	23
FIGURA 19: SEÑAL CBOC EN FASE O CBOC(+).	23
FIGURA 20: SEÑAL CBOC EN ANTIFASE O CBOC(-).....	24
FIGURA 21: UN CÓDIGO BARKER CUALQUIERA DE LONGITUD 11.....	25
FIGURA 22: FA DE UN CÓDIGO BARKER 13.....	25
FIGURA 23: CONSTRUCCIÓN DEL CÓDIGO BARKER COMPUESTO B_{54}	26
FIGURA 24: VENTANA DE TRABAJO TÍPICA EN EL ENTORNO MATLAB.	28
FIGURA 25: APLICACIÓN GUIDE.	29
FIGURA 26: MINI GUI CON GUIDE.....	30
FIGURA 27: ATRIBUTOS DEL BOTÓN <i>PUSHBUTTON1</i>	30
FIGURA 28: FICHERO <code>.m</code> GENERADO POR GUIDE.....	31
FIGURA 29: INTERPOLACIÓN DE LAS DOS MUESTRAS QUE CONFIGURAN UN PULSO RECTANGULAR PARA AUMENTAR LA RESOLUCIÓN A 6 MUESTRAS.....	34
FIGURA 30: INTERPOLACIONES. ARRIBA MUESTRAS ORIGINALES. EN EL MEDIO INTERPOLACIÓN ERRÓNEA. ABAJO INTERPOLACIÓN CORRECTA.....	35
FIGURA 31: SEÑAL BOC(1,1) AFECTA DE <i>MULTIPATH</i> CON MÚLTIPLES CORTES EN EL EJE TEMPORAL...	39
FIGURA 32 : LANZADOR DE LAS APLICACIONES.	43
FIGURA 33: PANTALLA DE INICIO DE LA APLICACIÓN FUNCIÓN DE AMBIGÜEDAD.....	44

FIGURA 34: MENÚ DESPLEGABLE CON LAS SEÑALES PREDEFINIDAS	45
FIGURA 35: VALORES DE LAS SEÑALES PREDEFINIDAS. DE IZQ. A DER.: PULSO RECTANGULAR, TREN DE PULSOS, PULSO MODULADO EN FRECUENCIA, CÓDIGO BARKER 13, BOC(3,2)	45
FIGURA 36: OPCIONES DE LA SEÑAL PULSO RECTANGULAR.	45
FIGURA 37: CÁLCULO DEL PULSO BOC(3,2).....	46
FIGURA 38: OPCIONES GRÁFICAS.	47
FIGURA 39: REPRESENTACIÓN DE LA FA DE UN PULSO RECTANGULAR RESTRINGIDO AL 8% Y AL 30% (IZQ.) DE LAS FRECUENCIAS POSIBLES.....	47
FIGURA 40: PROYECCIÓN DE LA FA DE UN PULSO RECTANGULAR USANDO EL 8% DE LAS FRECUENCIAS E INCLUYENDO LAS NEGATIVAS.	48
FIGURA 41: FA DE UN PULSO RECTANGULAR.....	48
FIGURA 42: MANIPULACIÓN DE LA GRÁFICA: ROTAR Y LEER EL VALOR DEL PUNTO.	49
FIGURA 43: VISTA LATERAL (FRECUENCIAS) Y FRONTAL (RETARDOS) DE LA FA DE UN TREN DE PULSOS RECTANGULARES.....	49
FIGURA 44: PANTALLA DE INICIO DEL PROGRAMA COTA CRAMÉR-RAO.	50
FIGURA 45: SEÑALES DISPONIBLES EN LA APLICACIÓN COTA CRAMÉR-RAO.....	51
FIGURA 46: LAS DIFERENTES SEÑALES CON SUS OPCIONES. DE IZQ. A DER. PULSO, SQRRRC, BOC, E1 Y GPS L1.	51
FIGURA 47: OPCIONES CRB.	52
FIGURA 48: OPCIONES GRÁFICAS.	52
FIGURA 49: LOS BOTONES PARA GENERAR LAS REPRESENTACIONES.....	52
FIGURA 50: DOS SEÑALES SUPERPUESTAS USANDO LAS OPCIONES 'FIX ON/OFF' EN ON Y 'COLOR'	53
FIGURA 51: EL ALMACÉN DE SEÑALES CON DOS SEÑALES GUARDADAS.....	54
FIGURA 52: RESERVANDO UNA SEÑAL EN EL CAJETÍN NUM. 2.	54
FIGURA 53: MANIPULACIÓN DE LA GRÁFICA. DE IZQ. A DER.: DESPLAZAR, ACERCAR, ALEJAR Y LEER EL VALOR DEL PUNTO.	54
FIGURA 54: PANTALLA DE INICIO DE LA APLICACIÓN <i>MULTIPATH</i>	55
FIGURA 55: SEÑALES DISPONIBLES	56
FIGURA 56: LAS DIFERENTES SEÑALES CON SUS OPCIONES. DE IZQ. A DER. PULSO, SQRRRC, BOC, CBOC, GPS L1 Y GALILEO E1.	56
FIGURA 57: OPCIONES <i>MULTIPATH</i> Y APERTURA DEL CORRELADOR.....	57
FIGURA 58: OPCIONES GRÁFICAS.	58
FIGURA 59: LOS BOTONES PARA GENERAR LAS REPRESENTACIONES.....	58
FIGURA 60: EL ALMACÉN CON TRES SEÑALES GUARDADAS.....	59
FIGURA 61: HERRAMIENTAS GRÁFICAS.....	59
FIGURA 62: ENVOLVENTE DE ERROR PARA VARIAS SEÑALES Y APERTURAS.....	60
FIGURA 63: LEYENDAS EN EL LUGAR MÁS APROPIADO.....	62
FIGURA 64: DEP CON LEYENDA.	63
FIGURA 65: PULSO RECTANGULAR CON $D = 1$ Y RESOLUCIÓN 10 (MORADO) Y RESOLUCIÓN 300 (MAGENTA).....	63
FIGURA 66: BOC(3,2) CON APERTURA $D=1$	64

Índice de Tablas

TABLA 1: CÓDIGOS BARKER26

Lista de Acrónimos

BOC	<i>Binary Offset Carrier.</i>
CBOC	<i>Composite Binary Offset Carrier.</i>
CRB	<i>Cramér-Rao Bound.</i> Cota de Cramér-Rao.
DEP	Densidad Espectral en Potencia.
DFT	<i>Discrete Fourier Transform.</i>
FA	Función de Ambigüedad.
FAQ	<i>Frequently Asked Question.</i> Pregunta frecuente.
GLONASS	<i>GLObal NAVigation Satellite System.</i>
GNSS	<i>Global Navegation Satellite System.</i>
GPS	<i>Global Positioning System.</i> Sistema de posicionamiento global.
GUI	<i>Graphic User Interface.</i> Interfaz gráfica de usuario.
IDFT	<i>Inverse Discrete Fourier Transform.</i>
ISI	<i>Inter Symbolic Interference.</i> Interferencia intersimbólica.
MATLAB	<i>Matrix Laboratory.</i>
PRN	<i>Pseudo Random Noise.</i>
RC	<i>Raised Cosine.</i> Coseno alzado.
SQRRRC	<i>Square Root Raised Cosine.</i> Raíz cuadrada coseno alzado.
TF	Transformada de Fourier.

1. Introducción

La presente memoria junto con los programas para MATLAB que la acompañan son el fruto de las tareas realizadas durante el Trabajo Fin de Carrera: *Herramientas desarrolladas en MATLAB aplicadas al estudio de los sistemas radar y de radionavegación*. Este constituye la sublimación de los conocimientos adquiridos durante el estudio de la carrera de Ingeniero Técnico en Telecomunicaciones cursada en la *Universitat Oberta de Catalunya*.

1.1 Motivaciones

El fin del siglo XX y la primera parte del siglo XXI se conocerá como la era de las telecomunicaciones. Si bien los avances en esta área en los últimos años han sido exponenciales, no debemos olvidar que las personas y sobre todo las materias, ya sean primas o elaboradas, han de trasladarse de un confín a otro del planeta de una manera segura y eficiente. Para que ello sea posible se debe recurrir a la tecnología y es aquí donde entran en juego los equipos de radar y los sistemas de navegación por satélite (GNSS). Los primeros son responsables de la detección y determinación de la distancia de objetos (aviones, fondos marinos, montañas...) permitiendo desplazamientos seguros evitando colisiones. Los segundos permiten que las personas, automóviles, barcos y aviones puedan conocer su posición con precisión, siendo éste un factor de relevancia extrema en situaciones comprometidas como son los aterrizajes de aviones o maniobras de buques. Ambos elementos, radar y GNSS, constituirán el primer foco de nuestro estudio.

Por otra parte, es de dominio público que los estudios de Ingeniería son marcadamente multidisciplinarios. Requieren conocimientos técnicos como son las matemáticas, física, programación, electrónica... También requieren un cierto grado de conocimientos humanistas como la psicología reflejada en la interacción de los seres humanos con las máquinas y ordenadores. Por tanto, el buen ingeniero ha de tener una buena capacidad de análisis, abstracción y síntesis para poder finalmente entregar un producto definido y operativo a la vez que útil y usable por terceros. El Trabajo Fin de Carrera debe ser representativo de estos ámbitos del conocimiento.

Veremos a lo largo de la presente memoria como a elementos no tangibles como las ondas electromagnéticas les haremos un análisis matemático, extraeremos sus

características relevantes, y las sintetizaremos de forma que puedan ser tratadas por medio de un programa informático que habremos creado a tal efecto. Este programa deberá ser usable por una tercera persona o usuario final. Es decir, cubriremos casi todas las áreas estudiadas durante la carrera, siendo este el segundo foco para nuestro trabajo.

La tercera y última motivación la constituye el hecho que un caso particular de GNSS, el sistema europeo Galileo, está en pleno desarrollo. Algunos conceptos que analizaremos son de plena actualidad y de muy reciente implementación. Esto genera un tercer y último foco de interés.

1.2 Objetivos

Los objetivos finales son la obtención de 3 programas para el entorno de cálculo MATLAB, junto con sus correspondientes interfaces gráficas de usuario (GUI), que permitan el estudio de algunos casos particulares de señales radioeléctricas en virtud de algunas de sus propiedades matemáticas. Una estará orientada a los sistemas radar y las otras dos al entorno GNSS.

1. La primera aplicación permitirá el análisis de las ondas recibidas por una instalación de radar. Para ello calcularemos la Función de Ambigüedad (FA) asociada a las señales.
2. La segunda aplicación, ya dentro del ámbito GNSS, será una herramienta que permita determinar la precisión con la que se pueda estimar el tiempo de retardo de una señal en su llegada al receptor. El objetivo será el cálculo de la Cota de Cramér-Rao (CRB). Para su determinación deberemos obtener el ancho de banda cuadrático medio que a su vez se es función de la distribución espectral de potencia (DEP) característica de cada señal.
3. Finalmente, la tercera aplicación que desarrollaremos es una herramienta que nos permita obtener la envolvente de error en la estimación de la posición que se produce en un receptor de navegación por vía satélite cuando está afecto del fenómeno del *multipath*. Estableceremos como escenario de referencia el caso ideal de una señal directa más una única señal reflejada.

Durante el desarrollo de las aplicaciones obtendremos una panorámica del tipo de señales empleadas en los sistemas de radar y los receptores de navegación por satélite.

Finalmente queremos hacer mención que no son objetivo de este trabajo las interioridades y detalles de los GNSS, ni las de los sistemas radar pero deberemos tener unos elementales conocimientos acerca de ellos. También deberemos ser conscientes que el tema de trabajo es muy amplio y digno de un estudio mucho más elaborado y profundo pero imposible de realizar en el tiempo propuesto de apenas un trimestre. No obstante, unas adecuadas referencias bibliográficas pueden compensar estas deficiencias.

1.3 Enfoque y metodología

Una vez conocidos los requerimientos básicos propuestos por el plan de estudios hemos desarrollado un plan de trabajo en el ámbito temporal. En función de éste, el proceso ha quedado estructurado en tres etapas:

- 1. Familiarización con el entorno de trabajo:** de entrada ha habido que adquirir unas habilidades adecuadas en el entorno de trabajo MATLAB comenzado desde cero. En una primera fase hemos recurrido a un grupo de tutoriales hasta que nos hemos sentido confortables para luego entrar en detalles más específicos relacionados directamente con los casos que nos ocupan.
- 2. Desarrollo de las aplicaciones:** a continuación ha habido que generar las tres aplicaciones propiamente dichas. Estas se han creado secuencialmente y para cada una de ellas se ha seguido el mismo método. Recepción del enunciado de los responsables de área, análisis de éste, estudio y especificación de los requerimientos, obtención de la documentación. A continuación se ha desarrollado la aplicación y efectuado las pruebas. Finalmente se ha redactado una memoria parcial y todo se ha entregado al consultor para evaluación.
- 3. Integración:** una vez completados los tres programas ha llegado el momento de integrar todos los elementos. Si bien las aplicaciones son entes autónomos, la presente memoria constituye el nexo de unión entre ellas. Así, como tarea final se han aplicado las modificaciones y correcciones necesarias a los programas, se ha estructurado la memoria final y creado una presentación multimedia.

1.3.1 Entregables

Según esta planificación los entregables del proyecto son:

1. Tres aplicaciones, más un programa lanzador.
2. Esta memoria que incluye los manuales de usuario de las aplicaciones.
3. Una presentación tipo *Powerpoint* descriptiva del trabajo.

1.3.2 Diagrama de tareas

En las Figura 1 y la Figura 2 podemos observar el plan de trabajo tanto a alto nivel como a bajo nivel.



Figura 1: plan de trabajo a alto nivel.

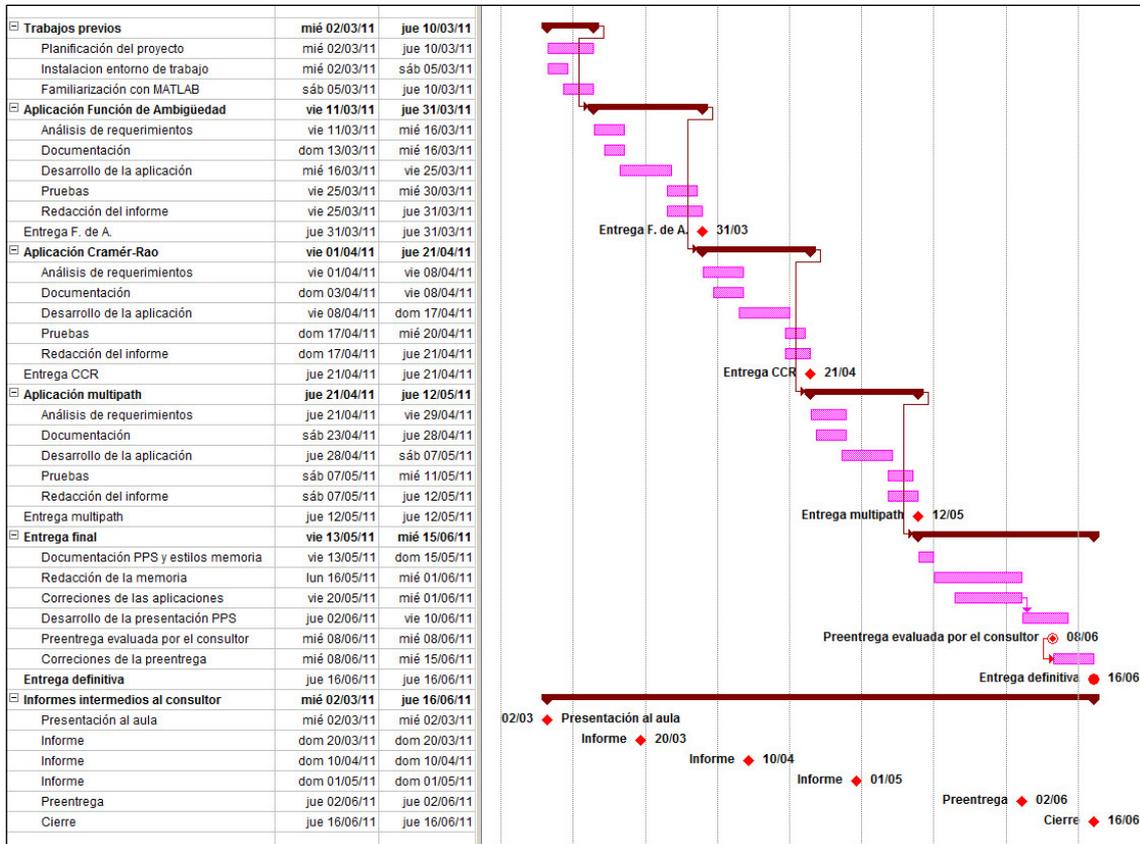


Figura 2: plan de trabajo a bajo nivel.

1.4 Esbozo de esta memoria

La memoria está estructurada en los siguientes bloques principales.

- **Capítulo 1: Introducción.** Agrupa las motivaciones, objetivos, planes de trabajo y metodologías empleadas.
- **Capítulo 2: Fundamentos teóricos.** En este bloque plantearemos las bases teóricas de los elementos que se desarrollarán en las GUI. Inicialmente trataremos los sistemas radar y la Función de Ambigüedad. A continuación expondremos los GNSS y desarrollaremos la teoría para la estimación del error en el retardo y el error en el posicionamiento debido a la presencia de interferencias *multipath*. Terminaremos describiendo algunas de las señales que usaremos.
- **Capítulo 3: MATLAB y GUIs.** Veremos una descripción de las interioridades del desarrollo y programación de una GUI para el entorno de trabajo MATLAB.
- **Capítulo 4: Algoritmos e implementaciones.** Haremos referencia a las funciones empleadas, técnicas usadas y problemas encontrados.
- **Capítulo 5: Conclusiones.** Veremos hasta donde hemos llegado y que se puede hacer tomando como base el presente trabajo.
- **Anexo: Manuales de usuario.** Para que un programa sea completo debe dotársele un manual de instrucciones. Será la guía de referencia a la que deba acudir el usuario para saber como se manejan las tres aplicaciones.

Con el fin de no saturar el documento con pies de página haremos las citas que hagamos las referenciaremos a la bibliografía por medio de un corchete indicando el autor y el año. Por ejemplo: [M. Irsigler, 2009].

2. Fundamentos Teóricos

En esta sección primero describiremos los elementos básicos de los sistemas radar y de la Función de Ambigüedad asociada. Seguidamente veremos los GNSS y los conceptos de densidad espectral en potencia, la cota Cramér-Rao, los discriminadores de los receptores GNSS y el problema del *multipath*. Terminaremos una descripción de las señales sobre las que trabajaremos

2.1 Los sistemas de radar

En esta sección definiremos los conceptos elementales de los sistemas de radar y la Función de Ambigüedad.

2.1.1 Conceptos básicos del radar

Radar es el acrónimo en inglés de *radio detection and ranging*. Esto es, detección y medición de distancias por ondas radioeléctricas. Básicamente, el procedimiento radar consiste en transmitir una onda y analizar esta misma onda una vez que un obstáculo la ha devuelto por medio de una reflexión.

La forma mas elemental del uso del radar consiste en la medición del tiempo que tarda una onda en ir desde el emisor hasta el obstáculo y volver (τ_R). En virtud de esta información somos capaces de discernir la distancia (R) del emisor al obstáculo puesto que el pulso se transmite a la velocidad de la luz (c)

$$R = \frac{c \cdot \tau_R}{2} \quad (1)$$

Los radares que operan de esta manera generan el problema del alcance máximo no ambiguo. Consideremos un obstáculo situado mas allá de una distancia que llamaremos distancia máxima de no ambigüedad (R_{amb}) y que entre la emisión de un pulso y el siguiente transcurre un tiempo t . En estas condiciones sucede que la onda reflejada por el obstáculo correspondiente al primer pulso llega al receptor después de la transmisión del segundo pulso. Para el observador es imposible discernir si la onda recibida corresponde al primer o al segundo pulso. Por tanto, un radar sólo puede operar unívocamente con obstáculos localizados a una distancia inferior a:

$$R_{amb} = \frac{c \cdot t}{2} \quad (2)$$

2.1.2 Frecuencia *doppler*

Aparte de las aparentemente elementales operaciones de detección y cálculo de la distancia a la que está un obstáculo, el radar se puede usar también para medir las variaciones de la distancia en el tiempo (\dot{R}) de dicho obstáculo. Notemos que esta variación, aún teniendo unidades de velocidad, no corresponde con la velocidad a la que se mueve el obstáculo, salvo que este se mueva alejándose o acercándose al observador radialmente.

Si tenemos un obstáculo a una distancia R el número de longitudes de onda (λ) contenidas en el recorrido de ida y vuelta es $2R/\lambda$. Como cada longitud de onda cubre 2π radianes, el recorrido angular total (φ) viene dado por:

$$\varphi = \frac{4 \cdot \pi \cdot R}{\lambda} \text{ radianes} \quad (3)$$

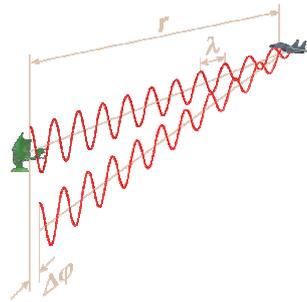


Figura 3: obstáculo acercándose al sistema radar.

En el caso que el obstáculo esté en movimiento, por ejemplo moviéndose a una velocidad radial v_R , R y por tanto φ , variarán en el tiempo. Observemos que si v_R es positiva (movimiento hacia el receptor) dR/dt será negativo. Así, podemos expresar la frecuencia angular (ω) como:

$$\omega = 2 \cdot \pi \cdot f = \frac{d\varphi}{dt} = \frac{-4 \cdot \pi}{\lambda} \frac{dR}{dt} = \frac{-4 \cdot \pi}{\lambda} \dot{R} \Rightarrow f = -2 \frac{\dot{R}}{\lambda} \quad (4)$$

Es decir, apreciamos una diferencia entre la frecuencia de la onda emitida y la recibida. A esta variación en la frecuencia se la conoce por **frecuencia doppler**.

$$f_D = -2 \frac{\dot{R}}{\lambda} \quad (5)$$

Obviamente $f_R = f_C + f_D$ donde f_R y f_C son las frecuencias de las envolventes de las ondas recibidas y emitidas.

2.1.3 Función de Ambigüedad

Como hemos visto un sistema radar basa su utilidad en el análisis comparativo de una señal de características conocidas después de ser reflejada por un obstáculo. Para ello el equipo radar dispone de un procesador de señales. Este, en su versión más elemental es un filtro pasa banda sintonizado a la frecuencia f_c de la portadora de la señal transmitida ($s(t)$) con el filtro adaptado¹ a la misma.

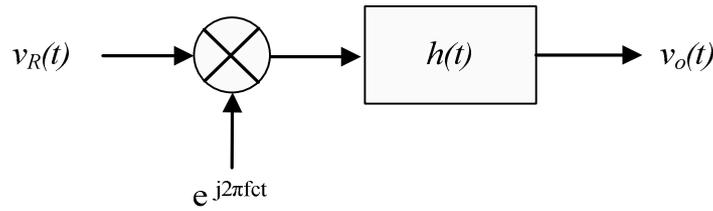


Figura 4: esquema de un procesador de señales de un sistema radar

Por tanto la FA no es más que la respuesta del procesador de señales del sistema a una señal recibida ($s_R(t)$). La FA adopta la forma de una función bidimensional $\chi(\tau, f)$ cuyas variables son el retardo en el tiempo (τ) y la frecuencia (f). Démosla una expresión analítica. Sea la onda transmitida expresada en su equivalente paso bajo complejo:

$$s(t) = u(t) e^{j2\pi f_c t} \quad (6)$$

En estas condiciones, la onda recibida tendrá la misma forma que la onda enviada pero desplazada en el tiempo y afectada en frecuencia por el desplazamiento *doppler*. Asumimos que en el proceso no hay atenuación en amplitud. Así:

$$s_R(t) = u(t - \tau_R) e^{j2\pi(f_c + f_d)(t - \tau_R)} \quad (7)$$

Si tenemos en cuenta que la respuesta impulsional del filtro adaptado viene dada por $h(t) = s^*(-t)$ podemos decir que la respuesta impulsional completa del procesador es:

$$h_{TOTAL}(t) = \left[e^{j2\pi f_c (-t)} u(-t) \right]^* \quad (8)$$

¹ En el caso que no empleemos un filtro adaptado, a la FA se la denomina FA cruzada. En nuestro desarrollo nos limitaremos a tratar con filtros adaptados.

Para calcular la señal obtenida a la salida del procesador deberemos hacer la convolución de la señal de recibida y la respuesta impulsional:

$$\begin{aligned} salida &= \int_{-\infty}^{\infty} s_R(t) h_{TOTAL}(\tau - t) dt = \int_{-\infty}^{\infty} \left(u(t - \tau_R) e^{j2\pi(f_c + f_d)(t - \tau_R)} \right) \left(e^{j2\pi f_c(t - \tau)} u(t - \tau) \right)^* dt = \\ &= \int_{-\infty}^{\infty} u(t - \tau_R) u^*(t - \tau) e^{j2\pi(f_c + f_d)(t - \tau_R)} e^{-j2\pi f_c(t - \tau)} dt \end{aligned} \quad (9)$$

Ahora bien, si hacemos el cambio de variable $z = t - \tau_R$ obtenemos:

$$salida = \int_{-\infty}^{\infty} u(z) u^*(z + \tau_R - \tau) e^{j2\pi(f_c + f_d)z} e^{-j2\pi f_c(z + \tau_R - \tau)} dz \quad (10)$$

Observamos que, para el resultado del procesamiento de la señal, es irrelevante cuándo se ha transmitido la señal. Lo importante es la diferencia $\tau_R - \tau$. Por otra parte, según [M. I. Skolnik, 1980] es usual tomar $f_c = 0$. Llamemos $\chi(\tau, f)$ a la señal obtenida y su expresión es:

$$\chi(\tau, f) = \int_{-\infty}^{\infty} u(t) u^*(t - \tau) e^{j2\pi f_d t} dt \quad (11)$$

Llegados a este punto tenemos una función que representa la respuesta del procesador de señales del radar a una señal recibida. Unos autores definen la FA como el cuadrado del módulo de $\chi(\tau, f)$ y otros sólo el módulo. A efectos prácticos, para nuestro trabajo, es indiferente una opción que otra. Optaremos por la versión sin cuadrado:

$$|\chi(\tau, f)| = \left| \int_{-\infty}^{\infty} u(t) u^*(t - \tau) e^{j2\pi f_d t} dt \right| \quad (12)$$

La FA tiene algunas propiedades muy interesantes:

- Alcanza su máximo en $|\chi(0, 0)| = |2E|$ donde E es la energía contenida en la señal recibida.
- Propiedad de simetría: $|\chi(-\tau, -f)| = |\chi(\tau, f)|$
- Si trabajamos con $|\chi(\tau, 0)|$ estaremos frecuencia *doppler* nula, esto es, la componente de velocidad radial del obstáculo es cero. Obviamente corresponde con la correlación de la señal transmitida.
- Si empleamos $|\chi(0, f)|$ estaremos observando la salida del procesador en un tiempo igual al tiempo al de la distancia al obstáculo.

2.2 Los Sistemas de Navegación Globales por Satélite

En esta sección veremos someramente el funcionamiento de un GNSS y los errores que les afectan. Nosotros abordaremos dos de estas indeterminaciones. Por un lado veremos como el tipo de señal afecta la estimación del retardo de una señal. Por otro lado valoraremos el error que se produce en el posicionamiento debido a la presencia del fenómeno *multipath*.

2.2.1 Conceptos básicos de GNSS

GNSS es un conjunto de sistemas basados en satélites artificiales que permiten que un receptor conozca su ubicación espacial y temporal. En la actualidad hay varios GNSS operativos de los cuales debemos destacar el GPS de origen americano y bajo control militar y el GLONASS, que es su equivalente ruso, el cual, tras unos inicios vacilantes está alcanzando el estado de operativo. Ambos sistemas comenzaron su despliegue a finales de los '70 y están en una amplia fase de modernización. La gran apuesta global de futuro es el sistema europeo Galileo que surge con un espíritu marcadamente civil y de compatibilidad con GPS.

Para hacer posible la mencionada localización cada satélite envía una señal de códigos pseudoaleatorios (PRN) cuyo mensaje de navegación contiene una marca temporal que depende del momento en que se transmite tomando como referencia unos relojes atómicos de alta precisión a bordo de los mismos. El receptor recogerá estas señales, las decodificará y en virtud de la marca temporal sabrá en que momento fueron enviadas. Conociendo el tiempo transcurrido entre emisión y recepción determinará la distancia al satélite. Si esto lo hace para un número suficiente de satélites, mínimo cuatro, el receptor será capaz de triangular su posición espacial y temporal con una precisión aceptable. Podemos encontrar detalles precisos relacionados con esta temática en [E. D. Kaplan et al. 2006].

No obstante, la evaluación del tiempo que tardan en viajar las señales se puede ver afectada por diferentes fenómenos como los atmosféricos, cambios en la ionosfera, interferencias *multipath*, errores de efemérides y de reloj... La gran mayoría de estos de errores se pueden corregir por medio de sistemas de aumentación como WAAS o EGNOS, los cuales, por medio de satélites geoestacionarios y una red terrestre, envían

mensajes de corrección a los receptores. Esto deja, como veremos más adelante, el fenómeno *multipath* como la mayor fuente de error de los GNSS.

2.2.2 Estimación del error en el retardo

Las señales en su viaje desde el satélite hasta el receptor están afectadas por interferencias aleatorias conocidas como ruido. Esto genera un error en el cálculo del tiempo que tardan en llegar al receptor (retardo). Usaremos la Cota de Cramér-Rao para delimitar la precisión con la que se pueda estimar este tiempo. Veremos a continuación que para su cálculo deberemos obtener el ancho de banda cuadrático medio que a su vez es función de la distribución espectral de potencia característica de cada señal.

2.2.2.1 Densidad espectral en potencia

De manera genérica la densidad espectral en energía o potencia (DEP) de una señal, nos muestra como está distribuida la energía o la potencia a través del rango de frecuencias que la componen. Una función $z(t)$ está definida en energía cuando su energía media es finita (y por tanto su potencia media es nula). Estará definida en potencia cuando su potencia media es finita, esto es, cuando la siguiente expresión existe y es finita:

$$P_z = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\infty}^{\infty} |z(t)|^2 dt \quad (13)$$

Aplicando la relación de Parseval

$$P_z = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\infty}^{\infty} |Z(f)|^2 df = \int_{-\infty}^{\infty} \lim_{T \rightarrow \infty} \frac{|Z(f)|^2}{T} df \quad (14)$$

El integrando es la definición de la **densidad espectral en potencia** $S_{zz}(f)$. La DEP se expresa en unidades de vatios por hertzio (W/Hz) o bien en dBm/Hz. En el caso de procesos aleatorios estacionarios en sentido amplio el teorema de Wiener-Khinchine afirma que:

$$S_{zz}(f) = R_{zz}(f) = \int_{-\infty}^{\infty} R_{zz}(\tau) e^{-j\omega\tau} d\tau \quad (15)$$

Siendo $R_{zz}(f)$ la TF de la autocorrelación de la señal. No obstante se genera el problema que el proceso es estacionario y por tanto la TF no es convergente. Se puede

demostrar ([R. G Brown et al, 1997]) que para procesos finitos en un periodo de tiempo T suficientemente amplio, la siguiente expresión es una estimación adecuada de la DEP:

$$E \left[\frac{1}{T} |Z(f)|^2 \right] \Rightarrow S_{zz}(f) \text{ cuando } T \rightarrow \infty \quad (16)$$

Donde $Z(f)$ es la TF de la señal $z(t)$ a lo largo del periodo T . Al término entre corchetes se le conoce con el nombre de **periodograma**.

2.2.2.2 Cota de Cramér-Rao

Como hemos visto, los receptores GNSS proporcionan la información de posición haciendo cálculos sobre el tiempo que tarda una señal concreta en llegar desde un satélite hasta el receptor. Estas señales, en su propagación, se ven alteradas por el ruido electromagnético. Este ruido está caracterizado por ser blanco, aditivo, gaussiano de media nula y potencia finita.

Para estimar este tiempo de retardo deberemos recurrir a estimadores insesgados, esto es, que su estimación tienda al valor real del parámetro en cuestión. El objetivo es encontrar aquel estimador con varianza mínima. La pregunta que surge de inmediato es: ¿existe algún límite para esa varianza mínima? En efecto, así es. Este límite viene determinado por la Cota (inferior) de Cramér-Rao (CRB). En [S. M. Kay, 1993] se demuestra que en el caso de las señales que estamos tratando la varianza mínima de la estimación del tiempo de retardo de una señal viene dado por:

$$\text{var}(\tau) \geq \text{CRB}(\tau) = \frac{1}{2 \frac{E_s}{N_o} B_{ms}^2} \quad (17)$$

Donde $\frac{E_s}{N_o}$ es la relación entre la energía de la señal y el ruido, B_{ms}^2 es el ancho de banda de Gabor o ancho de banda cuadrático medio definido por:

$$B_{ms}^2 = \frac{\int_{-\infty}^{\infty} (2\pi f)^2 |S(f)|^2 df}{\int_{-\infty}^{\infty} |S(f)|^2 df} \quad (18)$$

Donde $S(f)$ es la TF de la señal $s(t)$. Nosotros relacionaremos la CRB con la potencia de la señal (C) recibida a la salida del receptor frente al ruido. Esto es:

$$\frac{C}{N_o} = \frac{E_s / T}{N_o} = \frac{E_s}{N_o} B \quad (19)$$

Donde B es el ancho de banda del receptor. Así:

$$CRB(\tau) = \frac{B}{2 \frac{C}{N_o} \cdot B_{ms}^2} \quad (20)$$

2.2.3 Seguimiento y sincronía con las señales

En este apartado veremos cómo hace el seguimiento de las señales un receptor GNSS, analizaremos las interferencias fruto del *multipath* y cómo éstas generan un error en la estimación de la posición.

2.2.3.1 Adquisición y seguimiento

Como hemos comentado en el apartado dedicado a aspectos básicos de los GNSS, para que este pueda hacer su función debemos estar “conectados” a al menos cuatro satélites. Este proceso de conexión se denomina adquisición de la señal. Para ello sea posible, los satélites emiten un código llamado *Coarse Acquisition Code* (C/A code). Este código y la frecuencia de la portadora son diferentes para cada satélite. Así el procedimiento de adquisición consiste en identificar la frecuencia y el código que estamos recibiendo.

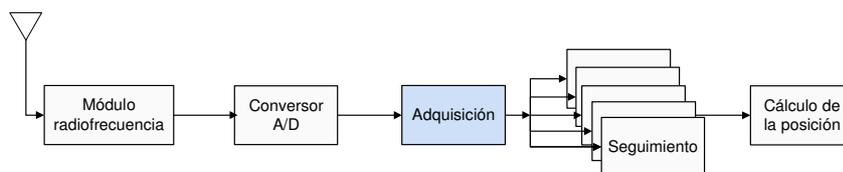


Figura 5: esquema de un receptor GNSS.

Hay diferentes maneras de hacer esta adquisición ([D. Plausinaitis, 2009]) pero la más sencilla computacionalmente, aunque no más eficiente, es generar una réplica local de cada una de las combinaciones C/A code – Frecuencia de Portadora posible y realizar la correlación con la señal entrante. A esta técnica se la denomina búsqueda en serie.

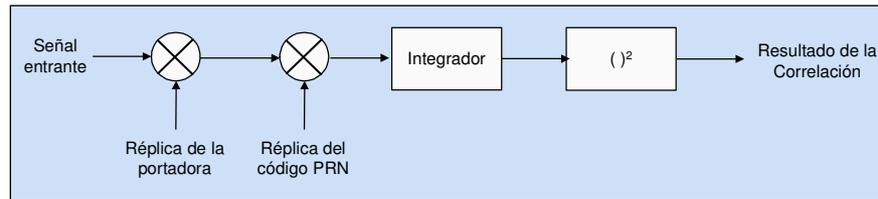


Figura 6: bloque de adquisición en un receptor GNSS.

En la Figura 1 podemos observar el resultado de una búsqueda en serie que revela la presencia de una señal GNSS.

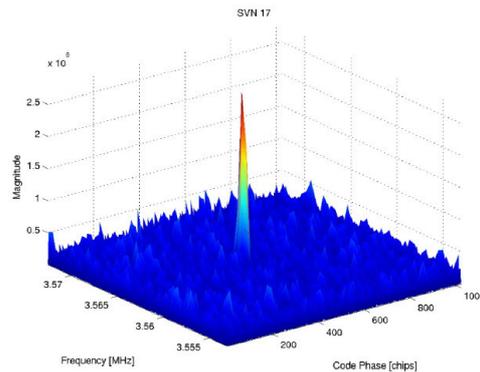


Figura 7: resultado de una búsqueda en serie donde, para un PRN dado, hay una señal GNSS presente [D. Plausinaitis, 2009].

2.2.3.2 El discriminador

Una vez que hemos adquirido la señal deberemos determinar la posición. La base para ello es la medición del tiempo que tarda la señal en viajar desde el emisor al receptor. El cálculo de la diferencia de tiempos no se puede hacer tomando directamente como referencia el reloj que incorpora el receptor ya que exigiría una sincronía perfecta entre este y el reloj del satélite. Para establecer esta sincronía, el receptor genera una señal idéntica a la que recibe con tiempo local y realiza la correlación entre ambas. Así determinará el retraso o adelanto de su propio reloj respecto al satélite y actuará en consonancia para mantener la sincronía con la señal que recibe. Este procedimiento se denomina *tracking* o seguimiento. Para efectuar esta tarea el receptor GNSS delega en el discriminador, el cual forma parte de los bloques de procesamiento del mismo. Existen dos tipos de discriminadores, los de portadora y los de código. Sin restarle importancia a los primeros, nosotros nos centraremos en estos últimos por ser conceptualmente más intuitivos.

Su forma de funcionamiento más elemental consiste en generar dos copias locales de la correlación de la señal esperada desplazadas en el tiempo. Un punto se

adelanta en el tiempo y es la correlación *early* (E) y otra se atrasa y constituye la correlación *late* (L).

Esta situación la podemos ver reflejada en la Figura 8. Por ejemplo, tomemos un pulso cuadrado modelado como la sucesión de 8 muestras de amplitud 1. La correlación de dicha señal es un triángulo de 17 muestras como el de la figura. Definimos dos puntos de análisis (marcas magenta) o correladores E y L, en este caso separados 6 muestras. La separación entre E y L se la denomina apertura del correlador (d) y se suele expresar en fracciones de $chip^2$. Usualmente 0.1 $chip$ para un correlador estrecho y un $chip$ para un correlador estándar o ancho.

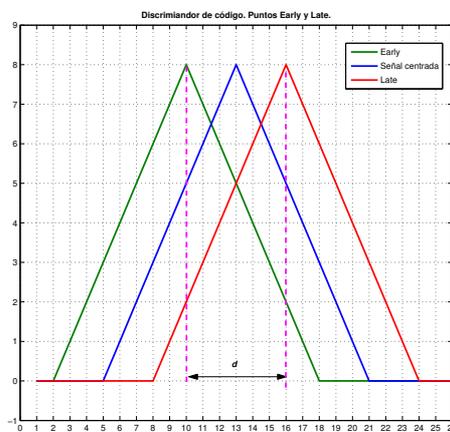


Figura 8: puntos *Early* y *Late* en un discriminador de código.

Si realizamos la diferencia entre ambas correlaciones nos dará una imagen como la de Figura 9. A la función resultante la denominaremos **curva del discriminador** o simplemente discriminador de la señal. Notemos que esta no es la única implementación posible del discriminador [E. D. Kaplan et al. 2006].

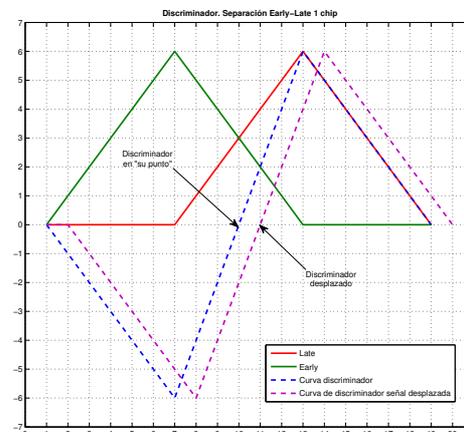


Figura 9: curva del discriminador. Separación E-L un $chip$.

² Veremos el concepto de *chip* en el apartado dedicado a las señales.

Cuando se recibe una señal, el sistema calcula su curva del discriminador. Comparando ambas curvas determinará la diferencia temporal entre ambas correlaciones viendo si está adelantada o retrasada. El bloque de seguimiento se encargará de corregir esta desviación permitiendo el *tracking* adecuado de la señal.

2.2.3.3 El problema del *multipath*

El problema del *multipath* o multicamino se origina cuando una misma señal llega al receptor en tiempos diferentes por haber recorrido distancias distintas. Podemos observar en la Figura 10 como el frente de onda llega al receptor y al suelo simultáneamente. En el suelo se produce una reflexión llegando esta hasta el receptor después de haber recorrido un camino R' adicional. La reflexión produce un cambio en las características de la señal tales como atenuación y cambios de fase. La llegada de esta señal genera una interferencia en la señal primigenia. Este fenómeno se apreciaba comúnmente, antes de la era de la Televisión Digital Terrestre, en los receptores de televisión, manifestándose por medio de una réplica de la imagen, difusa y desplazada superpuesta a la imagen correcta.

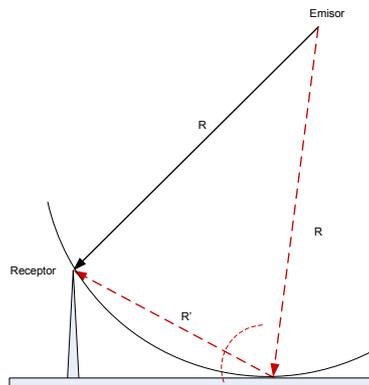


Figura 10: representación gráfica del fenómeno *multipath*.

Según [M. Irsigler et al. 2005] a día de hoy, el fenómeno *multipath* es el mayor inductor de error no evitable en los receptores GNSS. Las mejoras en el hardware o el uso de señales con un diseño más adecuado así como técnicas de modulación optimizadas reducen el resto de los errores en una cuantía muy significativa.

Ahora bien, ¿qué sucede con el *tracking* de un receptor GNSS en la presencia de un *multipath*? En palabras de [M. Irsigler et al, 2004] la forma más sencilla de hacer dicha evaluación se basa en considerar un único *multipath* atenuado de amplitud α y retrasado con respecto a la señal directa. La presencia de este genera un desplazamiento del cruce del discriminador por el eje temporal. Lo podemos apreciar en la Figura 11.

En ella tenemos tres gráficas. La azul representa la curva del discriminador de la señal directa. La verde corresponde a una señal *multipath* con una amplitud $\alpha=0.5$ desplazada dos unidades de tiempo respecto a la señal directa. La suma de ambas nos genera la curva del discriminador en presencia de un *multipath* (gráfica roja). Observamos que la curva del discriminador de la señal directa corta el eje temporal en un punto concreto. Por otra parte la curva del discriminador en presencia de *multipath* corta el eje en un tiempo distinto. La diferencia entre ambos tiempos (ΔR) es el valor del **error debido al *multipath***. ([P. Joosten et al. 2003]).

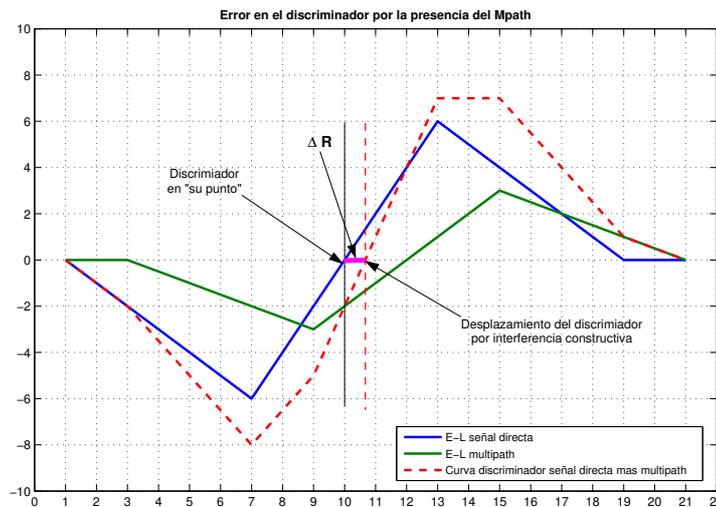


Figura 11: error de la curva del discriminador por la presencia de un *multipath*. Desfase 0° . Señal pulso rectangular.

Notemos que hemos considerado una interferencia con un desfase nulo entre la señal directa y el *multipath*, esto es, una interferencia constructiva. Podemos confeccionar una imagen como la Figura 12 similar para un desfase de 180° o interferencia destructiva.

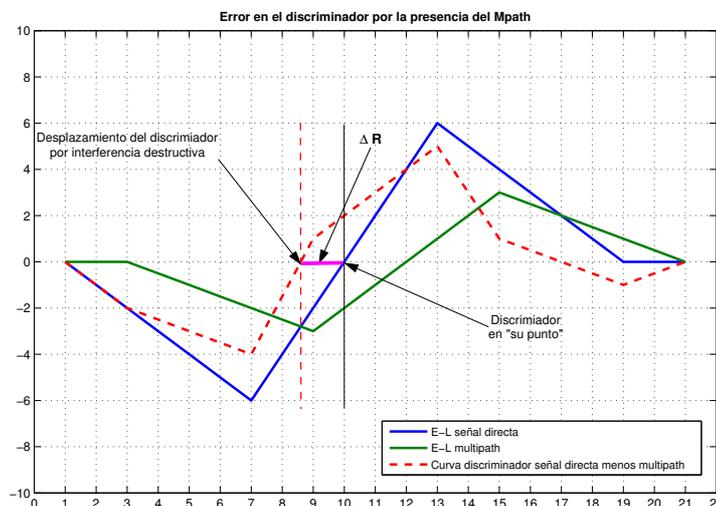


Figura 12: error de la curva del discriminador por la presencia de un *multipath*. Desfase 180° . Señal pulso rectangular.

Si realizamos valoración de ΔR para todos los tiempos posibles de la señal *multipath* en una interferencia constructiva obtenemos la envolvente de error para un desfase de 0° . Si repetimos el procedimiento para una interferencia destructiva obtenemos la envolvente de error para un desfase de 180° . Ambas situaciones son los dos casos peores posibles. [J. Blunt, 2007])

Es usual representar ambas gráficas en unidades de longitud expresadas en metros. La imagen obtenida constituye la **envolvente de error debido al *multipath***.

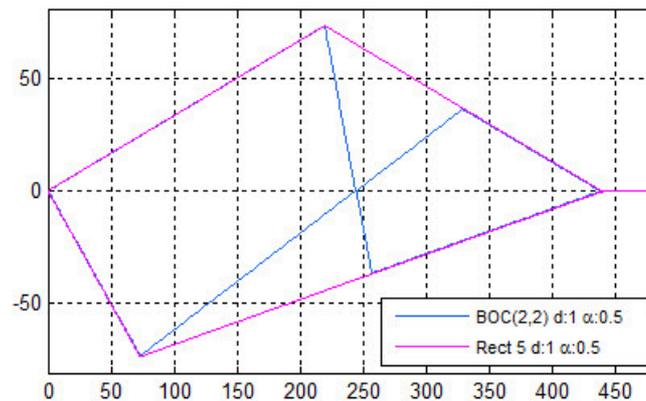


Figura 13: envolvente de error debido a *multipath* para las señales pulso rectangular y BOC(2,2).

Notemos que en el caso más sencillo, una señal consistente en un pulso cuadrado, el corte por el eje es único. En cambio, para señales empleadas en los nuevos GNSS como por ejemplo las BOC, sus curvas de discriminador cortan el eje por diferentes puntos generando una indeterminación. Esta situación la podemos apreciar en una representación para una señal BOC(1,1) con un *multipath* desplazado dos muestras en la Figura 14. Los círculos de color magenta marcan los varios pasos del discriminador por el eje temporal.

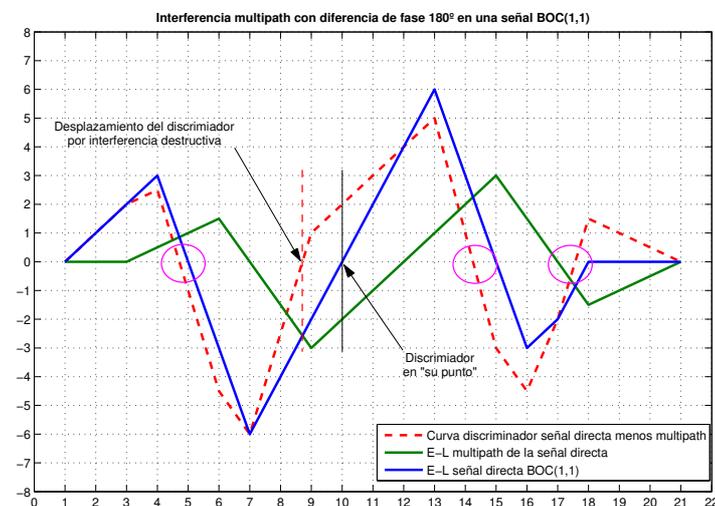


Figura 14: múltiples puntos de corte con el eje temporal para una señal BOC(1,1).

Las indeterminaciones pueden generar lo que se conoce como *false locks* o puntos de *tracking* incorrectos. Esto es, el discriminador se queda enganchado en el corte con el eje no apropiado. En la Figura 15 podemos observar como la autocorrelación de algunas señales presenta picos o arrugas que pueden inducir a estos *false locks*.

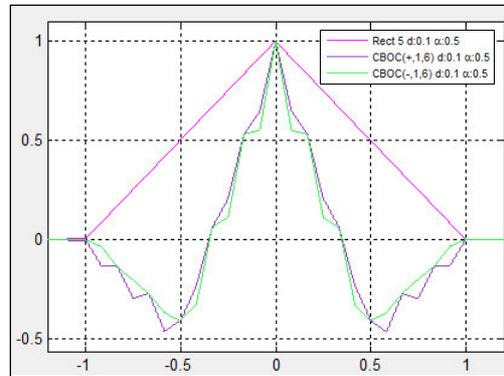


Figura 15: autocorrelación de pulso rectangular, CBOC(6,1,+) CBOC(6,1,-)

2.3 Señales

En esta sección haremos una revisión de las señales que hemos incorporado en las aplicaciones. Algunas de ellas no se usan en los GNSS o en radares pero, por algunas de sus características distintivas, serán útiles para poder usarlas como referencia para las empleadas en los sistemas objetivo de este trabajo.

2.3.1 Pulso rectangular

Trataremos el pulso rectangular como una sucesión de muestras de amplitud 1. Este pulso, en apariencia sencillo, constituye la base para las señales GPS L1. Los *chips* del código pseudoaleatorio que conforman sus señales se basan en un pulso rectangular de duración 1/1023 ms.

2.3.2 Square-Root Raised-Cosine

En el ámbito de las comunicaciones digitales se produce un fenómeno conocido como interferencia intersimbólica (ISI). Este consiste en que al muestrear la señal a través del filtro adaptado con una ratio de una muestra por símbolo no sólo obtenemos el símbolo si no de forma aditiva restos de símbolos precedentes y/o posteriores. Por tanto, es muy importante disponer de señales cuya ISI sea nula para poder realizar comunicaciones libres de errores.

Señales de este tipo son las que pertenecen a la familia de señales coseno alzado (RC) cuyas expresiones, según [M. Zoltowsky, 2010] frecuenciales y temporales son:

$$Z(f) = \begin{cases} T_s & 0 \leq |f| \leq \frac{1-\beta}{2T_s} \\ \frac{T_s}{2} \left(1 + \cos \left(\frac{\pi T_s}{\beta} \left(|f| - \frac{1-\beta}{2T_s} \right) \right) \right) & \frac{1-\beta}{2T_s} \leq |f| \leq \frac{1+\beta}{2T_s} \\ 0 & |f| \geq \frac{1+\beta}{2T_s} \end{cases} \quad (21)$$

$$z(t) = \frac{\cos \left(\pi \beta \frac{t}{T_s} \right) \sin \left(\pi \frac{t}{T_s} \right)}{1 - \left(2\beta \frac{t}{T_s} \right) \pi \frac{t}{T_s}} \quad (22)$$

La señal SQRRC está basada en la anterior y tiene por expresiones frecuenciales y temporales:

$$P(f) = |Z(f)|^2 \quad (23)$$

$$p(t) = \frac{2\beta}{\pi \sqrt{T_s}} \frac{\cos \left((1+\beta) \pi \frac{t}{T_s} \right) + \frac{\sin \left((1-\beta) \pi \frac{t}{T_s} \right)}{4\beta \frac{t}{T_s}}}{1 - \left(4\beta \frac{t}{T_s} \right)^2} \quad (24)$$

El parámetro β o factor de *roll-off* determina el ancho de banda del espectro de frecuencias como podemos apreciar en la figura siguiente:

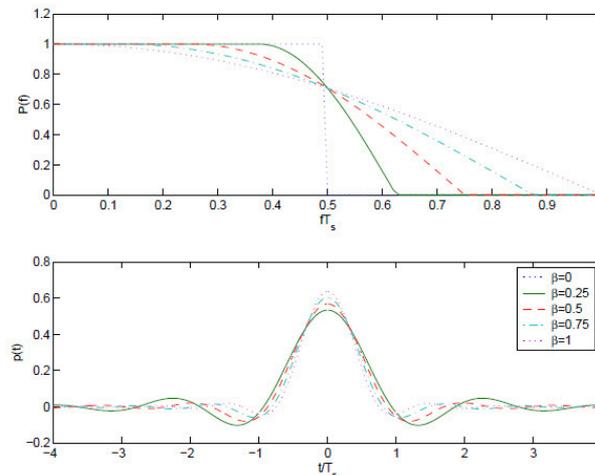


Figura 16: señal SQRRC con diferentes valores de β . Arriba dominio frecuencial, abajo temporal [M. Zoltowsky, 2010]

Notemos que la propia señal SQRRC no satisface la condición de ISI nula. No obstante, en recepción al pasar por esta por el filtro adaptado obtenemos la convolución de una SQRRC con otra que da lugar a una señal RC que si que tiene ISI nula.

Si observamos la Figura 16 la representación temporal de la señal SQRRC se parece mucho a la función *sinc*. En efecto, *sinc* es una SQRRC con *roll-off* nulo.

En teoría las señales SQRRC son de duración infinita. En nuestro desarrollo las truncaremos definiendo una cantidad de símbolos por pulso. Este recorte llevará a unos fenómenos en el espectro que podremos apreciar en la aplicación.

Finalmente, observemos que ni Galileo ni GPS usan señales SQRRC. No obstante estas se incorporan a este trabajo por ser de un ancho de banda limitado en contraposición al pulso cuadrado que es, idealmente, de ancho de banda ilimitado. Esto nos permite confrontar señales con estas características distintivas.

2.3.3 Binary Offset Carrier

La señal Binary Offset Carrier (BOC) es el elemento clave para el nuevo sistema GNSS Galileo promovido por la Unión Europea. La base de la misma, mas allá de la expresión analítica que podemos encontrar en [J. A. Ávila-Rodríguez, 2008] consiste en tomar una señal moduladora sinusoidal con frecuencia f_c múltiplo entero de 1023 y extraer de la misma únicamente el signo que toma. Por otra parte definimos un *chip* con frecuencia f_s que a su vez también es un múltiplo entero de 1023. El pulso BOC cumple la relación:

$$\phi = 2 \frac{f_s}{f_c} = 2 \frac{1023m}{1023n} = 2 \frac{m}{n} \quad (25)$$

Esto es, un pulso BOC dura m/n semiperiodos de la moduladora. Por ejemplo, si $m = 5$ y $n = 2$, el pulso BOC ocupará 5 semiperiodos y nos referiremos a éste como $\text{BOC}_{\sin}(5,2)$. Lo podemos observar en la Figura 17. El sufijo '*sin*' proviene de la modulación senoide. Igualmente se pueden definir señales BOC usando una moduladora cosenoidal y se mostrarán con el sufijo '*cos*'. No obstante, cuando nos referimos a una BOC sinusoidal, es habitual obviar el sufijo.

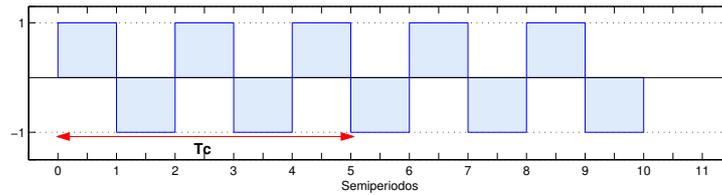


Figura 17: la sinusoide moduladora tomada sólo en su signo. T_c es el período de un *chip*, en este caso, BOC(5,2).

Para modular un código PRN con el *chip* o subportadora definido basta con multiplicar a aquel por el *chip* para obtener la señal a transmitir. Por ejemplo si el código a enviar fuera la secuencia $\{1, -1, -1\}$ obtendríamos la señal temporal de la Figura 18:

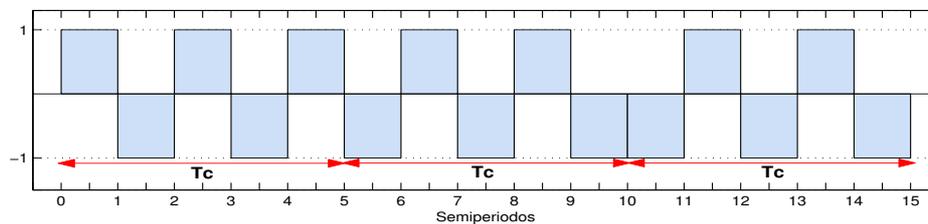


Figura 18: código modulado por un *chip* BOC(5,2)

2.3.4 Composite Binary Offset Carrier

Las señales BOC compuestas o CBOCs consisten en hacer sumas o restas de señales BOC donde cada una de las partes aporta una cantidad promediada de la potencia. Cuando se emplean sumas designaremos la señal como CBOC en fase o CBOC(+). Si se usa la diferencia para la composición y estaremos en el caso de una señal CBOC en antifase o CBOC(-). Finalmente podemos usar la suma para los *chips* pares y la diferencia para los impares obteniéndose una CBOC(+/-).

En la Figura 19 hemos compuesto en fase las señales BOC(1,1) y BOC(6,1) con potencias $1/11$ y $10/11$ respectivamente. En la Figura 20, la situación es la misma con la diferencia que las señales están en antifase.

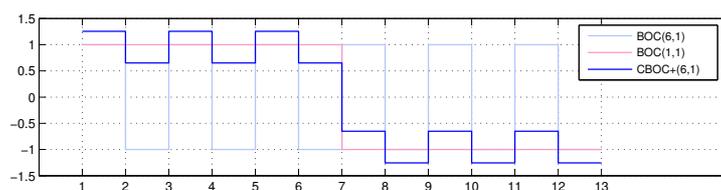


Figura 19: señal CBOC en fase o CBOC(+).

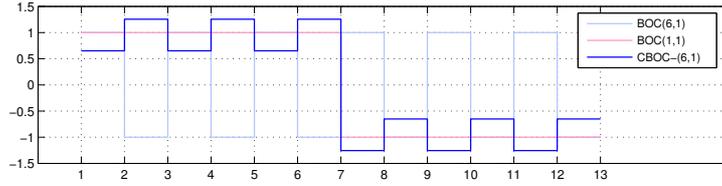


Figura 20: señal CBOC en antifase o CBOC(-).

La señal E1 ([OS IS ICD, 2010]) que se usa en el GNSS Galileo es precisamente el primer pulso BOC compuesto descrito.

2.3.5 Señales basadas en códigos binarios

Hemos visto que dos de las utilidades de los sistemas de radar son la determinación de la velocidad y la distancia de un objeto. En la actualidad el uso de códigos binarios está llevando a unas mejoras sensibles en este segundo apartado. Así mismo, estas señales son menos sensibles a los bloqueos por interferencia (*jamming*).

Una señal basada en un código binario se puede describir de forma general como ([B. R. Mahafza, 2009]):

$$x(t) = e^{j\omega_0 t} \sum_{n=1}^N P_n(t) e^{j(\omega_n t + \theta_n)} \quad (26)$$

Donde ω_0 es la frecuencia de la portadora, (ω_n, θ_n) unas constantes, N la longitud del código y $P_n(t)$ un pulso rectangular

$$P_n(t) = a_n \text{Rect} \left(\frac{t}{\tau_0} \right) \quad (27)$$

Con a_n o bien 1 o bien 0.

Las diferentes variaciones de los parámetros nos llevan los siguientes tipos de códigos:

- Tren de pulsos: $\omega_n, \theta_n = 0$ y $a_n = 1$ o 0
- Códigos binarios codificados en fase: $\omega_n = 0$ y $a_n = 1$
- Códigos binarios codificados en frecuencia: $\theta_n = 0$ y $a_n = 1$ o 0

2.3.5.1 Códigos Barker

Son un ejemplo característico de los códigos binarios codificados en fase. Se construyen a partir de un pulso de longitud N' el cual se divide en N subpulsos de ancho $\Delta\tau = N'/N$. A la fase θ_n de cada uno de estos subpulsos se le asignan de forma aleatoria los valores 0 y π . Cuando esta es 0 , se denota el subpulso como $+$ y si es π como 0 o $-$. Si la longitud del código es n , denominaremos al código Barker resultante B_n .

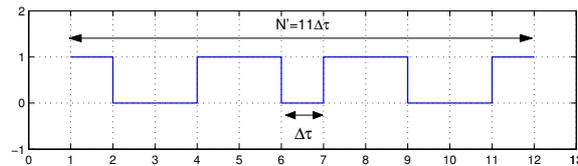


Figura 21: un código Barker cualquiera de longitud 11

Ahora bien, si los valores de las fases se seleccionan con cuidado obtenemos unos códigos cuya propiedad fundamental es que su autocorrelación produce un pico de altura N' y ancho $2\Delta\tau$ en el cero y los $(N'-1)/2$ lóbulos laterales son de valor constante 1. En la Figura 22 tenemos la representación de la FA de un código B_{13} donde la autocorrelación de la señal está dibujada en magenta. Podemos comprobar las características citadas. El valor de los picos de los lóbulos laterales es 0.07692 . Habida cuenta que la función está normalizada $1/0.07692 = 13.00052 \approx 13$. Así mismo comprobamos que tenemos 6 picos en cada uno de los lóbulos laterales.

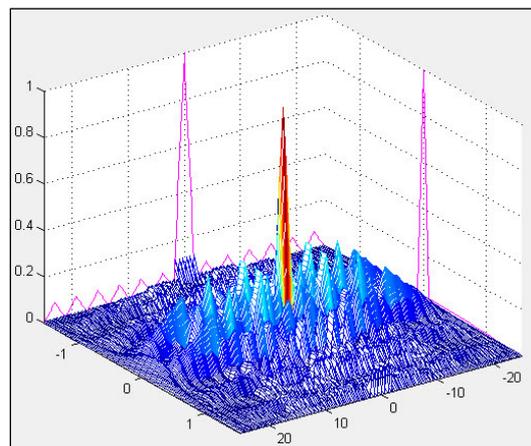


Figura 22: FA de un código Barker 13

Sólo existen 7 códigos Barker con esta propiedad. En la Tabla 1 los tenemos resumidos. Nótese que los de longitudes 2 y 4 se pueden expresar de dos formas diferentes.

Código	Longitud	Fases
B ₂	2	+ - + +
B ₃	3	+ + -
B ₄	4	+ + - + + + + -
B ₅	5	+ + + - +
B ₇	7	+ + + - - + -
B ₁₁	11	+ + + - - - + - - + -
B ₁₃	13	+ + + + + - - + + - + - +

Tabla 1: Códigos Barker

Los códigos Barker se pueden combinar conformando códigos compuestos como el de la Figura 23:

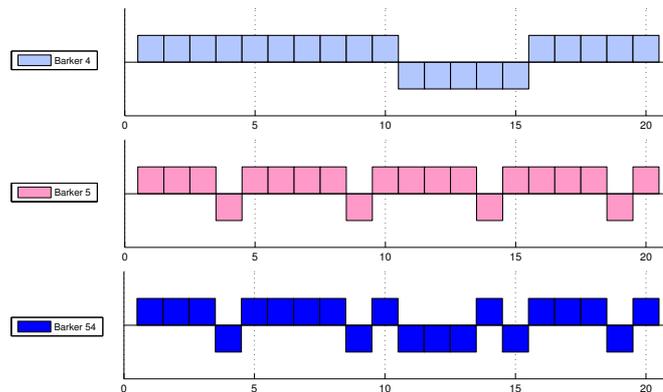


Figura 23: construcción del código Barker compuesto B₅₄.

3. MATLAB y las Interfaces Gráficas de Usuario

El entorno de trabajo MATLAB de la empresa MathWorks es un producto de gran solera en el ambiente de investigación y educativo. Sus primeras versiones datan de 1994 y está en constante evolución.

De entrada no es un producto al que se pueda acceder por técnicas de “*point and clic*” para obtener resultados inmediatos. Al neófito se le presenta una ventana con una línea de comandos desde la cual es muy probable que no sea capaz, como fue nuestro caso al comenzar el presente trabajo, representar siquiera la función seno. Su base de trabajo son los vectores y matrices de ahí su nombre (MATrix LABoratory). Así, como ejemplo, para resolver el problema de la representación de la función seno deberemos definir un vector de valores sobre el que se calculará, para cada punto, el seno. De esta manera obtendremos un nuevo vector que podremos representar con el comando apropiado. Una vez comprendida esta técnica, MATLAB se convierte en una poderosa herramienta de trabajo con infinitas combinaciones.

Antes de continuar, notemos que esta sección no pretende, ni puede ser en manera alguna, un manual de usuario para MATLAB. Este dispone de un manual de una extensión intimidatoria y de un servicio de asistencia a la comunidad por medio de foros excelente. La intención es suministrar unos conceptos muy genéricos acerca de algunas de las posibilidades de MATLAB. No obstante veremos con algo más de profundidad como se desarrolla una GUI.

3.1 El entorno de trabajo

Cuando inicializamos por primera vez MATLAB nos presentan una ventana con una línea de comandos para comenzar a trabajar. Esta ventana se puede configurar de diferentes maneras según nuestras necesidades. La experiencia nos dictaminará que elementos queremos tener a la vista. Por ejemplo en la Figura 24 tenemos el editor de programas, la línea de comandos, el historial de comandos, el árbol de ficheros y un área destinada a la representación gráfica.

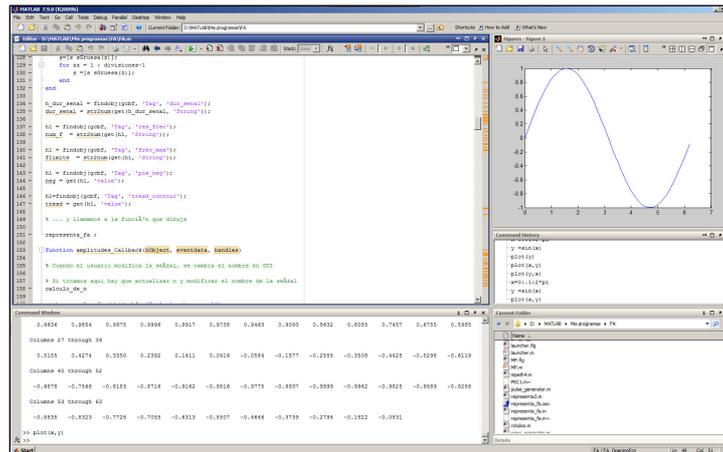


Figura 24: ventana de trabajo típica en el entorno MATLAB.

MATLAB incorpora, además de un extraordinario elenco de funciones matemáticas y rutinas, una serie de cajas de herramientas o programas interactivos para diferentes áreas de conocimiento: señales, comunicaciones, finanzas, bases de datos, filtros...

Otro elemento de gran importancia es el software paralelo denominado *Simulink* que permite modelar un determinado sistema y luego manipularlo para realizar simulaciones y análisis del mismo. Finalmente debemos reseñar las capacidades gráficas de MATLAB. Desde gráficas en dos dimensiones hasta complejos gráficos en 3D las combinaciones son infinitas.

3.2 El lenguaje de programación MATLAB

El lenguaje de programación de MATLAB es del tipo interpretado, esto es, no hay compiladores. A primera vista puede parecer un atraso y carente de prestaciones. Nada más lejos de la realidad. No obstante esta aproximación tiene una gran ventaja y es que podemos modificar una rutina o una función sin interrumpir la ejecución del programa. Cuando este llame a la rutina la ejecutará sin que haya que hacer compilaciones ni enlazados. Adicionalmente no tendremos que preocuparnos de reservar memoria ni tareas similares características de otros lenguajes.

Para crear un programa MATLAB basta con que lo redactemos en el editor que incorpora el entorno. Lo guardamos con la extensión `.m` y ya está listo para correrlo pulsando el icono correspondiente o llamándolo desde la línea de comandos.

Para la depuración podemos marcar puntos de pausa o *breakpoints* y comprobar el estado de las variables en el área *workspace* del entorno de trabajo.

Otra de las características del lenguaje MATLAB es que es orientado a objetos. Podremos crear clases gestionadas por valor o por *handles*. Esta última opción será la que más usemos en el desarrollo de los GUI.

Finalmente en MATLAB se pueden ejecutar programas escritos en otros lenguajes como C o Fortran.

3.3 Interfaces Gráficas de Usuario

Para que un lenguaje de programación revele todo su potencial debe permitir que los usuarios puedan interactuar con los programas de una forma cómoda. Para esto se desarrolló el concepto de GUI.

En MATLAB podemos desarrollar las GUI de dos maneras. O bien desde cero, lo cual implica tener que definir por texto cada uno de los elementos que configuran la GUI. O bien haciendo uso de una utilidad que incorpora MATLAB llamada GUIDE.

Para construir una GUI con GUIDE desde esta generaremos una distribución de los elementos que la componen (botones, textos, paneles, desplegables...), GUIDE compilará estos elementos en un fichero con extensión *.fig* y creará un fichero *.m* que deberemos programar para obtener las funcionalidades deseadas.

3.3.1 Uso de GUIDE

Para inicializar GUIDE deberemos ejecutar el comando `guide` desde la línea de comandos y obtendremos una ventana como la de la Figura 25.

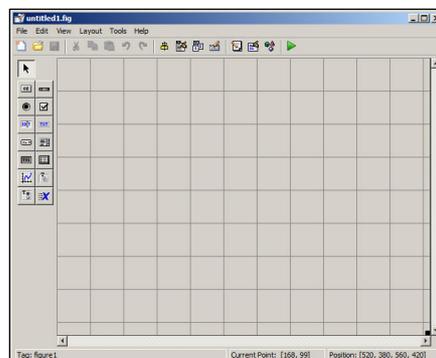


Figura 25: aplicación GUIDE.

En la Figura 26 podemos observar como hemos creado un panel y un botón. En esencia esta es la GUI mas sencilla que podemos crear.

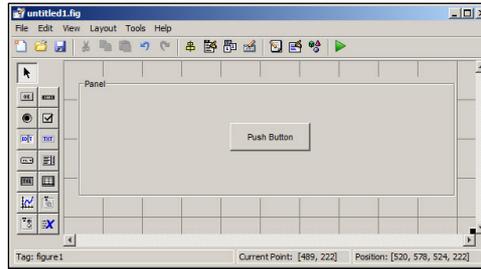


Figura 26: mini GUI con GUIDE.

Para modificar las características de los objetos deberemos hacer uso del inspector de atributos al que se accede con el menú contextual del botón derecho. Para el botón este tiene el aspecto de la Figura 27. Como atributos más obvios, resaltado en rojo, tenemos el 'String' que es el texto del botón (Push Button) o el 'BackgroundColor' (gris medio). No obstante los atributos más interesantes de la cara a la programación son 'Tag' y 'UserData'. El primero nos permite darle un nombre concreto y nos ayudará a encontrar el objeto suministrándonos su *handle*. El segundo es un campo que nos permite almacenar datos del tipo que queramos (números, textos...).

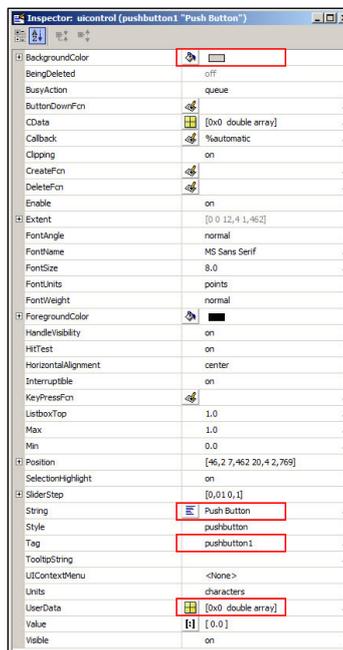


Figura 27: atributos del botón *pushbutton1*

Cuando guardemos la GUI, GUIDE generará el fichero `.m` que deberemos programar. Notemos que todos los elementos de la GUI son objetos del entorno de programación. Una vez que ejecutemos la GUI tendremos unas instancias de estos que podremos manejar y editar como tales.

3.3.2 Fichero .m de GUIDE

El aspecto del fichero .m generado por GUIDE es aproximadamente como el de la Figura 28. En él se ha eliminado alguna información superflua, en forma de comentarios, que genera de forma automática GUIDE.

```
function varargout = miniGUI(varargin)
% MINIGUI M-file for miniGUI.fig
%   MINIGUI, by itself, creates a new MINIGUI or raises the existing
%   singleton*.
%
% See also: GUIDE, GUIDATA, GUIHANDLES
%
% Last Modified by GUIDE v2.5 26-May-2011 14:25:58
%
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @miniGUI_OpeningFcn, ...
                  'gui_OutputFcn',  @miniGUI_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before miniGUI is made visible.
function miniGUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to miniGUI (see VARARGIN)

% Choose default command line output for miniGUI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = miniGUI_OutputFcn(hObject, eventdata, handles)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
```

Figura 28: fichero .m generado por GUIDE

Hemos resaltado la función `pushbutton1_Callback()`. Esta la es la que se ejecutará cuando pulsemos el botón. En este momento está vacía pero podemos programarla para que haga cosa de nuestro interés: representar un gráfica, llamar a otra función...

Para terminar queremos mostrar como se accede a un objeto concreto y como se modifican sus características. Esto se hace por medio de la función `findobj()`. Por ejemplo:

```
h_boton = findobj('Tag', 'pushbutton1');
```

Esta función nos devolverá la *handle* que apunta a la entidad cuya etiqueta ('Tag') es 'pushbutton1'. Una vez que hemos encontrado el objeto podemos leer los valores de sus atributos con funciones `get()` y modificarlos con `set()`. Por ejemplo, con las instrucciones:

```
color = get(h_boton, 'BackgroundColor');  
set(h_boton, 'BackgroundColor', rand(1,3))
```

obtendremos, por si tenemos interés en ello, el color del botón (`get()`) y lo modificaremos con el color que queramos, en este caso con un color aleatorio suministrando un vector con 3 valores elegidos al azar (`set()`).

Vemos pues, que en esencia, la interacción con la GUI a nivel de programación se realiza en base a encontrar la *handle* del objeto, consultar sus propiedades, modificarlas y recuperarlas. Esto aplica a cualquier elemento presente en una GUI: cajetines, botones de radio, gráficas... Pero no sólo a estos, si no a otros objetos que conforman otros más complejos. En una gráfica tenemos los ejes de coordenadas, los datos, la línea que representa los datos, las leyendas, etc.

4. Implementaciones y Algoritmos

La increíble potencia de MATLAB y de los ordenadores personales actuales a veces hace que trivialicemos ciertos cálculos que tres décadas atrás hubieran necesitado del acceso a ordenadores alejados de cualquier usuario corriente. Por este motivo en esta sección queremos resaltar algunos detalles y funciones empleadas en el desarrollo de las aplicaciones.

4.1 Implementación Función de Ambigüedad

Para implementar la FA en un programa informático debemos ser conscientes que debemos hacer una transición del mundo del tiempo continuo al del tiempo discreto. En esta transición las señales, de carácter complejo, no tendrán una duración infinita y vendrán representadas por una serie finita de N pulsos o muestras caracterizados por su amplitud y fase.

Así, la FA, la podemos reescribir como:

$$\chi[\tau, f_d] = \sum_{n=1}^N u[n]u^*[n-\tau]e^{j2\pi f_d n} \quad (28)$$

Donde $u[n]$ es la señal expresada de forma compleja y $u^*[n+\tau]$ es el complejo de conjugado $u[n]$ desplazado en el tiempo τ . De ahí que nos refiramos a τ como retardo. Notemos así mismo que $\chi[\tau, f_d]$ sólo tiene valores distintos de cero mientras las señales se solapan, esto es, para tiempos entre $-N+1$ y $N-1$. No obstante nosotros representaremos la FA para tiempos entre $-N$ y N .

En el programa, al ser nuestra primera aplicación realizada en MATLAB, hemos hecho el cálculo de la FA de forma directa usando dos bucles `for`. En [M. Budge, 2005] podemos encontrar un paso a paso de un algoritmo muy popular para el cálculo de la FA. Este algoritmo se basa en la relación siguiente:

$$F\{\chi[\tau, f_d]\} = F\{u(\tau)e^{j2\pi f_d \tau}\}F^*\{v(\tau)\} \quad (29)$$

Donde $F\{\}$ es la TF. Por tanto:

$$\chi[\tau, f_d] = F^{-1}\{F_u F_v^*\} \quad (30)$$

Para ellos se debe hacer uso de dos potentes funciones implementadas en MATLAB como son la Transformada Discreta de Fourier (DFT) y su inversa (IDFT).

Como hemos comentado, para representar una señal la descompondremos en muestras y cada una de estas muestras vendrá descrita por su amplitud, su fase y la fase de la moduladora. La aplicación trabaja sobre un conjunto de tres vectores que recogen estos valores. Por ejemplo, para describir un pulso rectangular no modulado deberemos introducir los valores de la amplitud de las muestras e ignorar las fases y modulaciones.

El programa permite aumentar la resolución de la señal aumentando las muestras. Para ello interpolará valores en función de estas. Hemos creado el parámetro resolución que indicará cuántas interpolaciones incluiremos entre dos muestras consecutivas. Por ejemplo no tiene mucho sentido tener que definir un pulso rectangular como una sucesión de 100 muestras de amplitud 1. Es más cómodo definir dos muestras y aumentar la resolución.

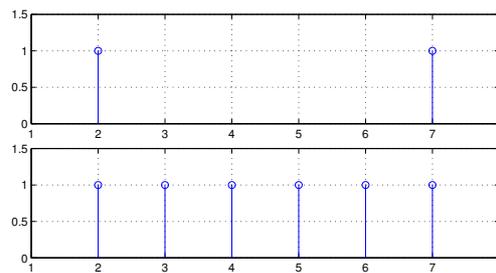


Figura 29: interpolación de las dos muestras que configuran un pulso rectangular para aumentar la resolución a 6 muestras

En la interfaz aparece el número de puntos temporales usados dependientes del número de muestras y de la resolución aplicada. Así, si tenemos 2 muestras y aplicamos una resolución de 30, el programa usará $30 + 2$ elementos temporales.

Veremos que muchas veces el número de puntos empleados no coincide con lo que debería teóricamente. Esto se debe a la forma de expresar las señales. Las muestras [1 1] son fáciles de interpolar como hemos visto. Pero las muestras [1 0 0 1] no lo son tanto. ¿Que colocamos entre el 1 y el 0? ¿Y entre el 0 y el 1? ¿La media de los valores? El programa calculará cuantos unos y cuantos ceros para mantener el balance de la señal. Podemos observar un ejemplo de esta situación en la Figura 30.

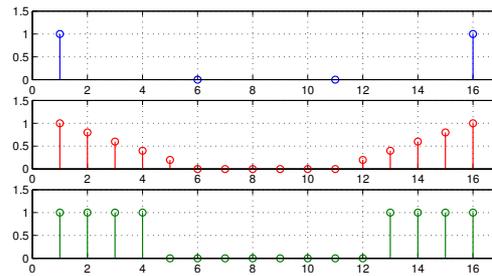


Figura 30: interpolaciones. Arriba muestras originales. En el medio interpolación errónea. Abajo interpolación correcta.

Hay establecido un parámetro que es la frecuencia máxima que nos permite centrar nuestro interés en el origen de coordenados. Podemos elegir cuantos cortes vamos a usar así como si queremos usar o no las frecuencias negativas. Recordemos que la FA es simétrica. Cuando usamos las frecuencias negativas se representa el corte por *doppler* cero, esto es, $\chi(\tau, 0)$.

Podremos optar por representar la FA en un gráfico tridimensional usualmente complicado de interpretar o usar la opción de contornos o curvas de nivel muchos más explicativas aunque menos espectaculares. Para ello se usa la representación `contour()`.

Para comodidad de uso y resultados instantáneos hemos incluido algunas señales predefinidas: pulso rectangular, varios pulsos rectangulares consecutivos, pulso modulado en frecuencia, pulso variable en fase (Barker 13) y el pulso BOC. Se podrá acceder a ellas por medio de un menú desplegable.

Finalmente debemos reseñar que los resultados se expresan en unidades de microsegundos y megahercios. Se podría haber optado por dejarlos referenciados a la duración de la señal, pero creemos que es más intuitivo de esta manera.

4.2 Implementación Cota Cramér-Rao

El objetivo final es la obtención de un programa que se capaz de mostrar gráficamente las densidades espectrales de diferentes señales así como la CRB. También disponemos de una opción de representar simultáneamente más de una señal. El desarrollo se ha centrado en la formulación de la interfaz gráfica ya que los cálculos necesarios los realiza MATLAB por medio de sus propias funciones. A continuación

hacemos una descripción de los elementos presentes, así como una mención de las herramientas y funciones empleadas en el desarrollo del programa.

4.2.1 Elementos presentes

La GUI dispone de un menú desplegable con señales predefinidas. Según el tipo de señal elegida dispondremos de unos campos u otros con valores por defecto pero editables. Siempre tendremos a la vista una representación gráfica en el dominio temporal de la señal.

Por otra parte disponemos de las opciones para el cálculo de la CRB en función de relación de la potencia de la señal con la del ruido y el ancho de banda del receptor. Así mismo hay una opción para aumentar la resolución de la gráfica.

La parte que consideramos mas interesante es el almacén de señales. Una vez que hemos encontrado una señal de nuestro interés la podemos almacenar y recuperar cuando queramos.

Finalmente tenemos un área de representación de la densidad espectral con las frecuencias normalizadas y el valor en decibelios. El área inferior muestra la acotación inferior del retardo aplicando la CRB dentro del margen de valores de la relación señal/ruido seleccionados. Los colores de la gráfica son variables.

4.2.2 Funciones empleadas

Para el cálculo de la densidad espectral hemos recurrido a la obtención del periodograma. Como hemos descrito en el apartado teórico basta con calcular la TF (`fft()`) de la propia señal y elevar al cuadrado el módulo. Para centrar los datos en la frecuencia 0 deberemos rotar el resultado (`fftshift()`). Para aumentar la resolución de la señal debemos recurrir a la técnica del *zero padding* suministrando a la función de cálculo de la TF cuantos puntos queremos usar en el mismo.

Para el cálculo de la CRB debemos obtener el ancho de banda de Gabor que a su vez se obtiene del cálculo de dos integrales. Para ello hemos usado la función de sumas acumulativas trapezoidales (`trapz()`). Esta da una aproximación de la integral de un grupo de valores agrupados en un vector con un paso o dx que será la unidad. Notemos que para hacer los cálculos correctamente deberemos aplicar el paso correspondiente a

nuestra función multiplicando por este. Como ejemplo, para calcular $\int_0^\pi \sin(x)dx$ cuyo valor es 2 deberemos hacer:

```
x = 0:pi/100:pi;
y = sin(x);
Z = pi/100 x trapz(y)
```

donde $\pi/100$ sería el paso referido.

Como hemos comentado en el caso de CRB debemos calcular unas integrales. Estas se hacen sobre la DEP y sus límites de integración (frecuencias) vendrán impuestos por el inverso de la duración de la señal. El paso lo determinará el número de valores que tengamos de la DEP.

Finalmente, notemos que si bien los valores C/N0 se expresan en dB-Hz estos se han de evaluar en el programa en magnitudes lineales.

$$dB = 10 \log(x) \Rightarrow x = 10^{\frac{dB}{10}} \quad (31)$$

4.3 Implementación *multipath*

El objetivo es la obtención de un programa que se capaz de mostrar gráficamente las correlaciones de diferentes señales así como la envolvente de error debido a la presencia de un *multipath*.

4.3.1 Elementos presentes

Encontramos un menú desplegable con las señales predefinidas. Según la selección de la señal dispondremos de unos campos u otros con valores por defecto pero editables. Siempre tendremos a la vista una representación gráfica en el dominio temporal de la señal. Para calcular la envolvente del error consideramos un único *multipath* presente. Las opciones para el cálculo permiten variar la amplitud relativa del *multipath* y el ancho d del correlador (distancia *early-late*).

Como opción gráfica podemos aumentar la resolución de la representación de la correlación de las señales que a su vez afecta a la representación de la envolvente.

4.3.2 Funciones empleadas

Para el cálculo de la correlación hemos empleado la función de correlación (`xcorr()`) de MATLAB. Al resultado le hemos aplicado una función que nos permite interpolar (`interpoll()`) una serie de datos para que el aspecto gráfico sea mejor.

El algoritmo general para calcular la envolvente de error es el siguiente:

- Tomamos la señal y calculamos su autocorrelación.
- Desplazamos la autocorrelación a los puntos *early* y *late* y calculamos su diferencia obteniendo la curva del discriminador.
- Hacemos lo mismo para un *multipath* atenuado y desplazado.
- Componemos ambos resultados por medio de una suma, Esto nos da una curva que corta el eje temporal por un punto diferente a la curva del discriminador.
- Evaluamos la diferencia en tiempo entre ambos cortes. Recordemos la Figura 11 y la Figura 12.
- Lo repetimos para todos los desplazamientos posibles del *multipath*. Esto nos da la envolvente para un desfase de 0° .
- Repetimos todos los cálculos haciendo la composición por medio de una diferencia. Esto nos da la envolvente para un desfase de 180° .

Observamos pues, que la clave para el cálculo de la envolvente de error es encontrar el corte por el eje temporal de la curva compuesta. Para ello hemos usado una función del laboratorio de soporte de MathWorks. Se trata de la función `crossing()` de Stephen Bruckner.

Una vez que hemos averiguado la situación de los puntos de corte debemos saber cuál de ellos es el que tenemos que evaluar. Recordemos que cualquier señal que en origen presenta algún corte por el eje temporal también presentará múltiples puntos de corte en su curva del discriminador. Para limitar el número de cortes que se encuentran hemos tomado la porción central de la curva compuesta y buscado el cero en un entorno del centro de $\pm d/2$, esto es, $\pm(E-L)/2$. No obstante se nos presenta un problema debido a la naturaleza discreta de los cálculos. Si no disponemos de suficientes puntos y esto lo combinamos con una apertura del discriminador alta obtendremos múltiples puntos de corte. La solución es buscar los puntos de corte en un entorno más pequeño, pero si hacemos el entorno demasiado pequeño es posible que la

curva compuesta no corte el eje temporal tal y como observamos en la Figura 31. En esta apreciamos que un entorno demasiado ancho nos produciría dos cortes por el eje temporal y uno demasiado estrecho ninguno.

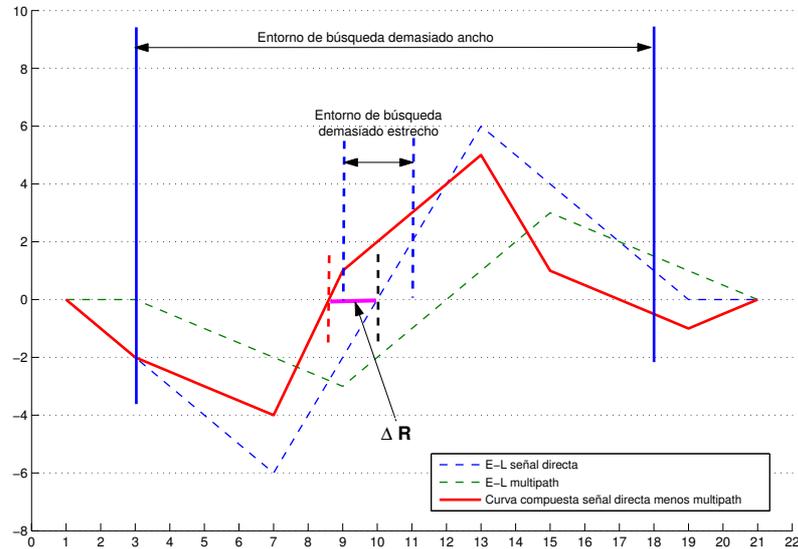


Figura 31: señal BOC(1,1) afectada de *multipath* con múltiples cortes en el eje temporal

Para solucionar el dilema hemos usado el referido entorno $\pm d/2$ y cuando encontramos múltiples puntos de corte empleamos aquel que está más cerca del punto de corte de la curva del discriminador

5. Conclusiones

Después de leer múltiples veces la memoria y probar y reprobado las aplicaciones pensamos que hemos alcanzado los objetivos propuestos: obtener unos programas y una documentación destinada a terceros que permitan analizar visualmente las propiedades de algunas señales partiendo de una descripción paramétrica de las mismas. Adicionalmente, se ha trabajado mucho en cotejar los datos obtenidos con los que figuran en la documentación disponible. Pensamos que estos refrendan nuestros resultados.

La lectura y pruebas de las aplicaciones, no obstante, también revelan como se ha desarrollado el trabajo. Se ha partido de nulas nociones de MATLAB y se han ido implementado las aplicaciones secuencialmente según requerimientos y como unidades individuales. Si tuviéramos que volver a desarrollar el proyecto lo enfocaríamos pensando en confeccionar una aplicación integrada bajo una definición uniforme de las señales a lo largo de todo el proyecto. Esto es, primero especificación de las señales y a continuación desarrollo de los cálculos, funciones, gráficas, etc.

Examinando el código de las aplicaciones se hace patente la evolución en las habilidades con MATLAB. Esto es, hemos ampliado el espectro de habilidades que debe tener un ingeniero cumpliendo de esta manera con una de las premisas de partida: la multidisciplinaria.

Por otra parte, para cada aplicación ha habido que tomar un enunciado, buscar bibliografía, tomar los elementos necesarios de esta y darle aplicación práctica. Hasta aquí nada reseñable. Es lo habitual en un proyecto. Sin embargo, las referencias bibliográficas que hemos encontrado, virtualmente sin excepción, están dirigidas a un público maduro en la temática, son de una extensión apabullante y, en ocasiones, de una densidad asombrosa. Esto ha dificultado de sobre manera el trabajo. Pero, recordemos que unos de los objetivos propuestos era trabajar sobre material de plena actualidad y este es un precio que hay que asumir por ello.

Queremos pensar que este trabajo puede servir como puente para que otros estudiantes salven la distancia que existe entre los conocimientos que se adquieren en asignaturas como Sistemas Lineales o Transmisión Digital y una realidad física compleja como son los sistemas de radar y GNSS.

Finalmente somos conscientes que no hemos hecho más que rascar la superficie de una materia extensa y, en cierto grado, bastante compleja. La puerta queda abierta para una reevaluación, un nuevo planteamiento y a continuación, el desarrollo sistemático de una gran aplicación destinada al análisis de señales.

5.1 Trabajo futuro

Esta memoria está redactada de una manera muy esquemática y se han quedado en el tintero muchos enfoques y temas para desarrollar. Como hemos comentado, pensamos que puede servir como una base de trabajo para el perfeccionamiento o nuevo desarrollo de las aplicaciones propuestas o como guión para un trabajo más amplio.

De cara a trabajos concretos a desarrollar podemos citar:

- Calcular la Función de Ambigüedad por el método sugerido por [M. Budge, 2005]. Si el tiempo lo hubiera permitido y después de las habilidades adquiridas nos hubiera gustado implementarlo.
- Mejora del cálculo de la envolvente de error: la técnica que hemos aplicado es correcta, pero se debe optimizar la búsqueda de los cortes por el eje temporal. O bien buscar otra alternativa como puede ser el análisis de la pendiente de las curvas.
- Un cajón de señales: sistematizar la descripción de las señales. Una aplicación que genere las señales en un formato concreto y que luego se puedan reusar. Como hemos visto en la evolución de las aplicaciones hemos tenido que reescribir elementos que ya estaban definidos pero con matices diferentes. Hubiera sido muy práctico disponer desde el inicio de un formato descriptivo de las señales.
- Cálculo de la envolvente de error *multipath* considerando un doble discriminador en el receptor u otros tipos de discriminadores.
- Creación de un receptor GNSS por software. Quizás en conjunción con otro grupo de trabajo que quiera desarrollar el aspecto del hardware dedicado a la adquisición física de las señales.

6. ANEXO: Manuales de Usuario

Esta sección recoge los manuales de usuario para las 3 aplicaciones construidas. Notemos que estas son completamente independientes entre sí aunque hemos construido un pequeño programa lanzador desde el que se puede ejecutar cualquiera de ellas.

Asumiremos que el usuario tiene conocimientos elementales de MATLAB y sabe donde tiene que volcar los programas y que tiene que tener el *path* actualizado. También asumiremos que tiene los conocimientos teóricos adecuados para saber que realizan las aplicaciones. Si no es así es recomendable leer previamente la sección 2 de esta memoria: “Fundamentos Teóricos”

Las imágenes que hemos empleado en estos manuales provienen de correr las aplicaciones en el sistema operativo Windows 7 64 bits y MATLAB 7.9.0 (R2009b), no obstante las aplicaciones definitivas se han codificado y probado en toda la extensión posible en Linux Ubuntu 32 bits y MATLAB 7.10 (R2010a). Por tanto el aspecto de las ventanas puede diferir pero los contenidos y la operatividad serán los mismos.

6.1 Inicio rápido. Aplicación ‘lanzador’

Volcar todos los ficheros referenciados en el apartado 5.5 en el mismo directorio, anotar dicho directorio en el *path* de MATLAB, ejecutar el programa ‘lanzador’ (`launcher.m`) y seleccionar la aplicación deseada. En caso de duda volver a este punto del manual y leer detenidamente.



Figura 32 : lanzador de las aplicaciones.

6.2 Aplicación Función de Ambigüedad

Podremos ejecutar la aplicación desde el lanzador (ver apartado 5.1) o bien ejecutando programa `FA.m`

6.2.1 Descripción de la pantalla

Una vez que hayamos inicializado el programa obtendremos una ventana con la siguiente imagen con 5 zonas definidas:

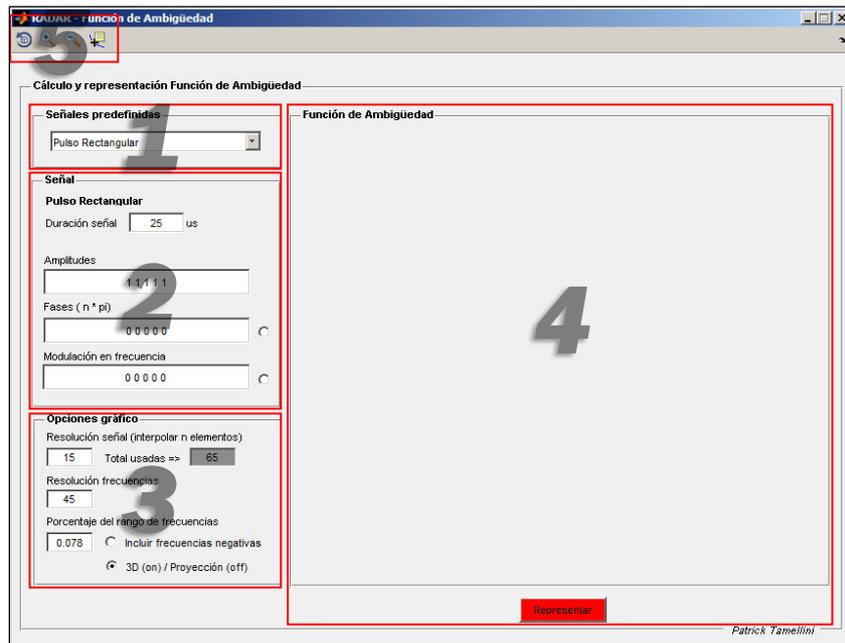


Figura 33: pantalla de inicio de la aplicación Función de Ambigüedad.

- **Zona 1:** menú desplegable con una serie de señales predefinidas.
- **Zona 2:** es donde definimos las características de la señal. Cuando ejecutamos el programa por primera vez esta zona se precarga con los datos para representar un pulso rectangular.
- **Zona 3:** en este apartado podemos seleccionar las diferentes opciones para la gráfica propiamente dicha.
- **Zona 4:** es el área donde se representará la Función de Ambigüedad cuando hagamos uso del botón '*Representar*'
- **Zona 5:** herramientas para la manipulación de la gráfica.

6.2.2 Descripción de las zonas

Podremos editar cualquier elemento de las zonas 2 y 3 salvo que esté sombreado. Para ello basta con marcar el cajetín e introducir los datos.

NOTA: dejando el puntero sobre los cajetines o botones por medio de un mensaje de texto obtendremos alguna ayuda o consejo adicional

6.2.2.1 Selector de señales predefinidas

Por medio del menú desplegable podemos seleccionar una de las señales predefinidas. Entre ellas se incluyen el pulso rectangular sin modular, el tren de pulsos rectangulares, el pulso modulado en frecuencia, un ejemplo de pulso modulado en fase (código Barker 13) y el pulso BOC en su versión sinusoidal.

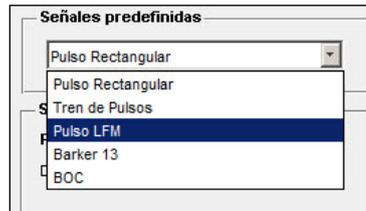


Figura 34: menú desplegable con las señales predefinidas.

Cuando seleccionamos una señal se cargan las características de la misma y así como las opciones gráficas más acordes a la señal. Se recomienda comenzar con estos y luego probar con otros valores. A veces pequeñas variaciones pueden llevar a grandes mejoras en el aspecto gráfico de la FA.

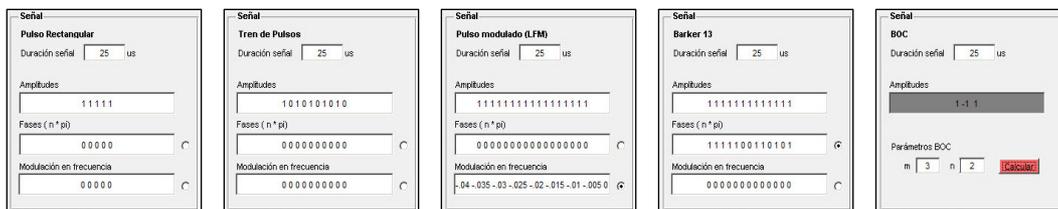


Figura 35: valores de las señales predefinidas. De izq. a der.: pulso rectangular, tren de pulsos, pulso modulado en frecuencia, código Barker 13, BOC(3,2).

6.2.2.2 Características de la señal

En la zona 2 se cargarán las características de la señal. Estas son editables por el usuario.

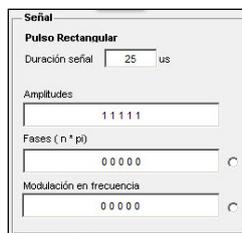


Figura 36: opciones de la señal pulso rectangular.

- **Duración de la señal:** es lo que dura toda la señal en μs . Se pueden usar expresiones del tipo 10^6 para indicar la duración en segundos.
- **Amplitudes:** son las amplitudes de cada una de las muestras de la señal. Puede tomar valores negativos y decimales.

- **Fases:** es el valor de la fase de cada muestra en partes de π . Si queremos indicar una fase de 90° deberemos introducir 0.5 o un $1/2$.
- **Modulación en frecuencia:** representa la modulación en frecuencia de cada una de las muestras.

NOTA: en estos cajetines debe haber al menos un espacio entre valor y valor pudiendo haber más de uno. Es muy importante que haya la misma cantidad de valores en los 3 cajetines. Si no, se producirá un error en el programa.

- **Radiobuttons:** junto a los cajetines de modulación en fase y en frecuencia se encuentra un botón para usar o no estas características. Si el botón está encendido se usarán los datos del cajetín.

NOTA: si estamos manejando señales donde sólo nos interesa la amplitud, por ejemplo señales BOC, será más práctico dejar sin marcar los cajetines de fase y de frecuencia. La aplicación ignorará por completo sus contenidos.

- **El pulso BOC:** en el caso del tipo señal BOC deberemos tener cuenta que viene caracterizada por ser una sucesión de pulsos de amplitudes positivas y negativas fruto de los parámetros m y n . Cuando seleccionamos la señal BOC de entre las predefinidas se nos presenta por defecto $m = 3$ y $n = 2$. Para generar el pulso BOC deberemos calcularlo pulsando el botón '*Calcular*'. El cajetín de amplitudes se actualizará de acuerdo con los parámetros. Este cajetín no se puede editar manualmente.

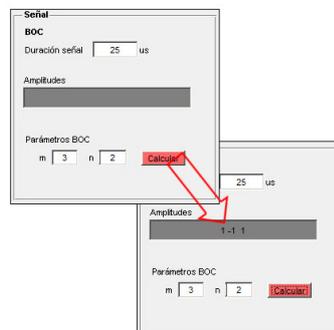


Figura 37: cálculo del pulso BOC(3,2).

NOTA: si pulsamos '*Representar*' antes de calcular el pulso BOC se producirá un error.

6.2.2.3 Opciones gráficas

En la zona 3 tenemos las opciones que nos permiten personalizar la representación gráfica de la FA.

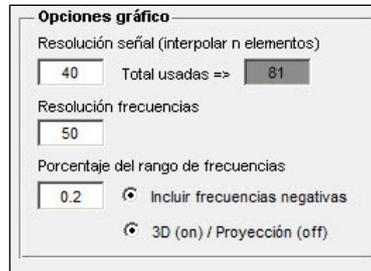


Figura 38: opciones gráficas.

- **Resolución de la señal:** para mejorar la gráfica, sobre una señal dada podremos interpolar tantas muestras como hayamos especificado, esto es, aumentar el número de muestras a evaluar de la señal. Dependiendo de las características de la señal no siempre es posible interpolar tantas muestras como las especificadas, no obstante, el programa nos indicará el total de muestras que se usarán.

NOTA: la resolución de la señal afectará directamente al número de divisiones del eje temporal (τ).

- **Resolución frecuencias:** indicaremos cuantas frecuencias *doppler* se han calcular.
- **Frecuencia máxima:** cantidad porcentual de frecuencias a calcular. En la FA la zona de interés se suele concentrar en un rango de frecuencias próximo al cero. Con esta opción lo que hacemos a es determinar este rango. Un 100% indicaría calcular todas las frecuencias posibles.

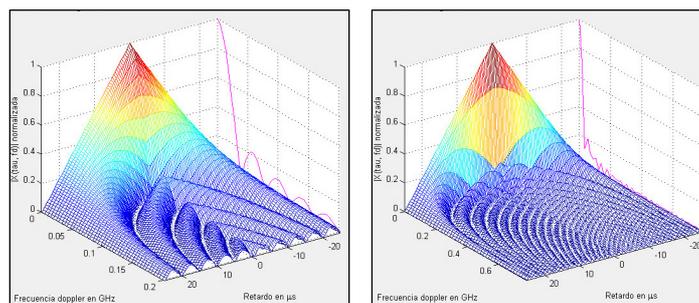


Figura 39: representación de la FA de un pulso rectangular restringido al 8% y al 30% (izq.) de las frecuencias posibles.

- **Incluir frecuencias negativas:** si lo marcamos el programa representara también las frecuencias negativas. Aunque la FA es simétrica con $|\chi(-\tau, -f)| = |\chi(\tau, f)|$ no es cierto que $|\chi(\tau, f)| = |\chi(\tau, -f)|$. Cuando incluimos frecuencias negativas se representará el corte por la frecuencia cero de la FA.
- **3D/Proyección:** activado obtendremos una representación tridimensional. Desactivado una proyección de curvas de nivel.

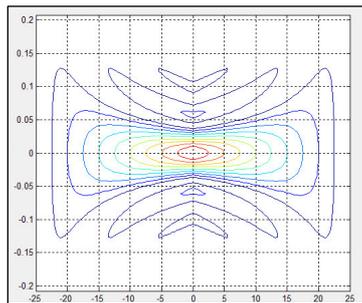


Figura 40: proyección de la FA de un pulso rectangular usando el 8% de las frecuencias e incluyendo las negativas.

6.2.2.4 Área de la gráfica y botón ‘Representar’

Pulsando ‘Representar’ obtendremos una imagen de la FA en el área gráfica.

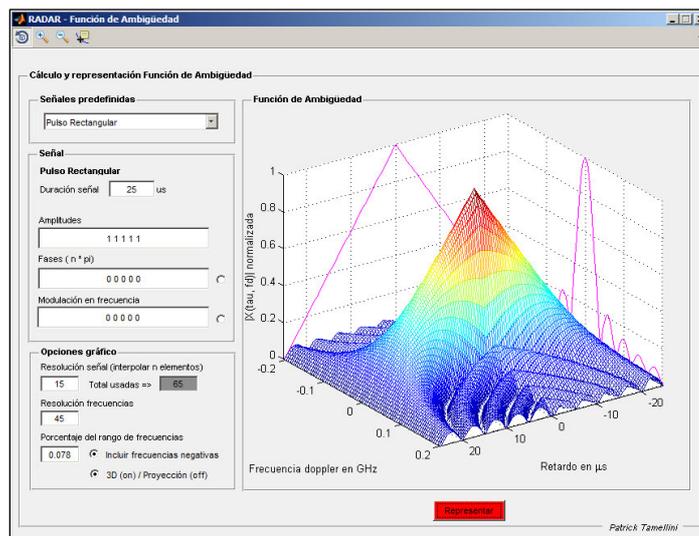


Figura 41: FA de un pulso rectangular.

6.2.2.5 Herramientas para manejar la gráfica

Una vez que hemos representado la gráfica podremos manipularla por medio de los iconos que encontramos en la zona 5:



Figura 42: manipulación de la gráfica: rotar y leer el valor del punto.

Una vez que hemos seleccionado uno de los iconos podemos desplegar un menú contextual con el botón derecho. Por ejemplo, podemos seleccionar vistas laterales.

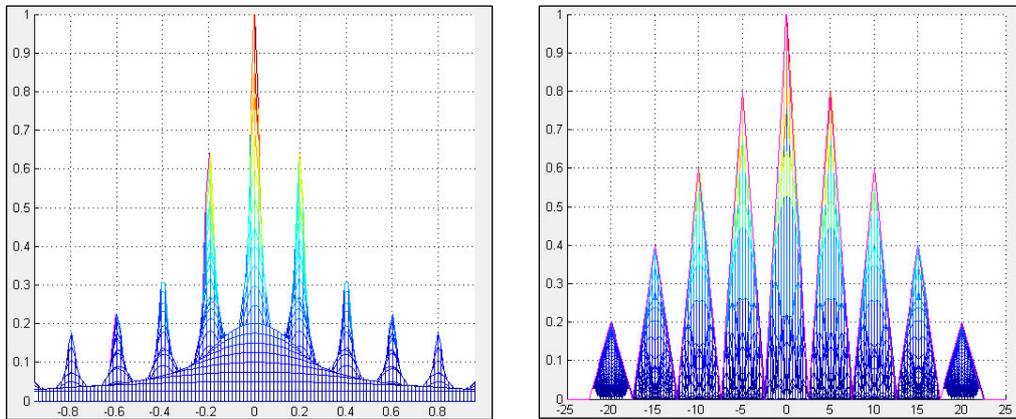


Figura 43: vista lateral (frecuencias) y frontal (retardos) de la FA de un tren de pulsos rectangulares.

6.3 Aplicación Cota Cramér-Rao

Podremos ejecutar la aplicación desde el lanzador (ver apartado 5.1) o bien ejecutando programa CCR.m. En el apartado 5.5 hay una relación de los ficheros asociados a esta aplicación.

6.3.1 Descripción de la pantalla

Después de inicializar la aplicación obtendremos una ventana con la siguiente imagen con 7 zonas definidas:

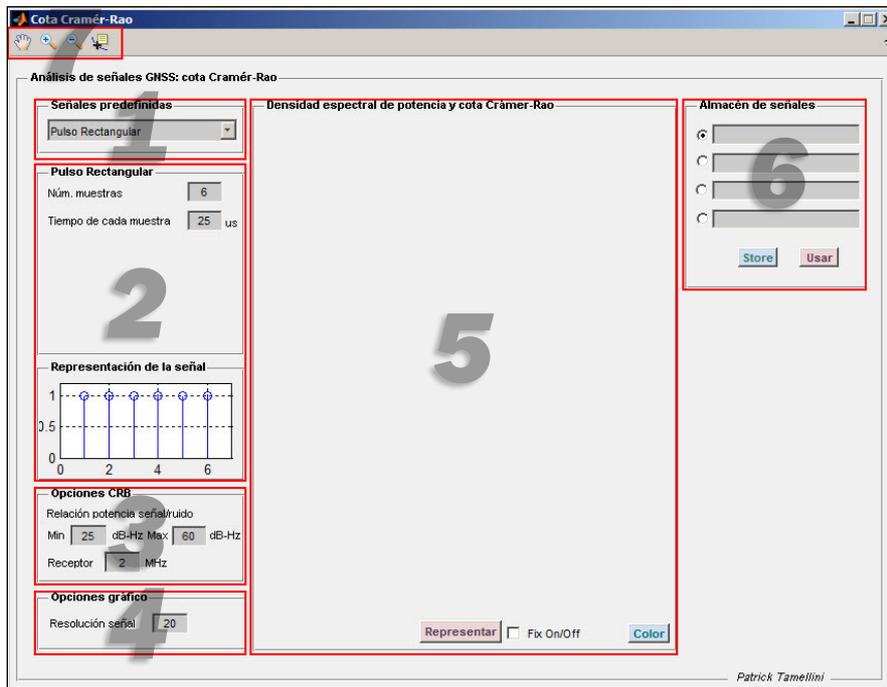


Figura 44: pantalla de inicio del programa Cota Cramér-Rao.

- **Zona 1:** selector del tipo de señal.
- **Zona 2:** área descriptiva de la señal incluida una representación gráfica de sus valores temporales.
- **Zona 3:** opciones para el cálculo de la CRB.
- **Zona 4:** opciones para la representación gráfica.
- **Zona 5:** área para la representación de la densidad espectral y de la CRB.
- **Zona 6:** zona para almacenar las señales de nuestro interés.
- **Zona 7:** herramientas para la manipulación de las gráficas.

6.3.2 Uso del programa

A continuación describimos el procedimiento general para la utilización del programa.

6.3.2.1 Selección de la señal

Lo primero que deberemos hacer es seleccionar una señal del desplegable de la zona 1 o bien hacer uso de la señal por defecto “Pulso Rectangular”.



Figura 45: señales disponibles en la aplicación Cota Cramér-Rao.

Podemos elegir de un elenco de 5 señales: pulso rectangular, raíz cuadrada coseno alzado (SQRRC), Binary Offset Carrier (BOC), GPS-L1 y la BOC compuesta empleada en el sistema GNSS Galileo.

Dependiendo de la señal elegida el panel de la zona 2 que contiene los datos de la señal se modificará acordeamente.



Figura 46: las diferentes señales con sus opciones. De izq. a der. pulso, SQRRC, BOC, E1 y GPS L1.

Creemos que los elementos son autoexplicativos. La modificación de cualquiera de ellos lleva consigo la actualización de la representación temporal

NOTA: Nótese que en la señal SQRRC, símbolos se refiere a la cantidad de símbolos que conforman un pulso o *shape* y no al número de pulsos. En esta aplicación siempre usaremos un único pulso. Esto es, no usaremos secuencias de pulsos.

NOTA: si en una señal BOC creamos una combinación no posible para m y n , la GUI mostrará ** en el número de muestras y la representación temporal de la señal será nula.

6.3.2.2 Opciones CRB

Disponemos de dos opciones para la CRB. Por un lado la relación entre la potencia espectral de la señal y la del ruido. Debemos indicar el valor máximo y el mínimo y este será el rango por el que se calculará y representara la CRB. Por otra parte tenemos el ancho de banda del receptor.

Opciones CRB
Relación potencia señal/ruido
Min 25 dB-Hz Max 60 dB-Hz
Receptor 2 MHz

Figura 47: opciones CRB.

NOTA: Si introducimos valores máximos menores que el mínimo la aplicación tomará el mínimo y viceversa.

6.3.2.3 Opciones gráficas

La única opción disponible es aumentar la resolución de la representación de la señal. Cuanto más grande mas resolución.

Opciones gráfico
Resolución señal 20

Figura 48: opciones gráficas.

6.3.2.4 Representación gráfica

Una vez que han sido seleccionadas y/o configuradas las señales podemos pasar a representar la función.

Representar Fix On/Off Color

Figura 49: los botones para generar las representaciones.

En la zona 5 o área gráfica disponemos de 3 controles:

- **Representar:** genera las gráficas.
- **Fix On/Off:** cuando está marcado, al representar una nueva señal, ésta se superpone a la anterior.
- **Color:** sirve para cambiar el color de la última señal representada. El color por defecto de la señal es magenta. El color que se usará al pulsar 'Color' es uno aleatorio. Si el color seleccionado no es de nuestro agrado basta con volver a pulsar el botón 'Color'.

NOTA: los botones 'Representar' y 'Color' son idénticos en su funcionalidad con la salvedad que el primer representa en color magenta y el segundo en un color aleatorio.

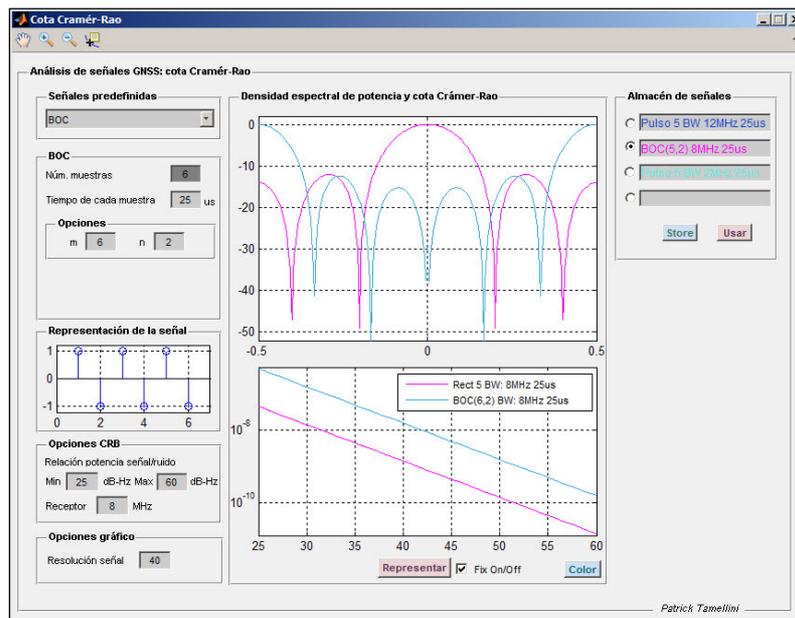


Figura 50: dos señales superpuestas usando las opciones 'Fix On/Off' en on y 'Color'.

6.3.2.5 El almacén de señales

Si la **última** señal representada es de nuestro interés podremos reservarla en el almacén de señales que encontramos en la zona 6. En éste disponemos de dos controles:

- **Store:** almacena la señal en el cajetín seleccionado. Podemos hacer esto aún cuando no hayamos representado la señal.

- **Usar:** vuelca los datos de la señal del cajetín seleccionado en la GUI y representa la señal.

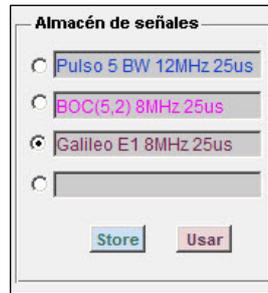


Figura 51: el almacén de señales con dos señales guardadas.

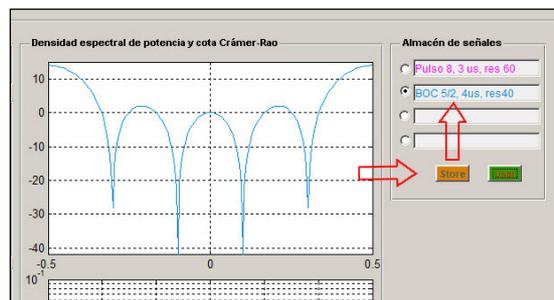


Figura 52: reservando una señal en el cajetín num. 2.

NOTA: al pulsar 'Usar' se representa la señal. Dependiendo el estado del control 'Fix On/Off' esta se superpondrá o no. En este último caso las señales no guardadas se perderán.

6.3.2.6 Herramientas para manejar la gráfica

Una vez que hemos representado la gráfica podremos manipularla por medio de los iconos dispuestos en la zona 7:



Figura 53: manipulación de la gráfica. De izq. a der.: desplazar, acercar, alejar y leer el valor del punto.

6.3.3 Interpretación de los resultados

La aplicación nos permitirá comprobar que señales minimizan la varianza del retardo en función de su distribución espectral en potencia. También podremos hacer uso de la aplicación para encontrar el ancho de banda del receptor que hace que dos señales diferentes tengan la misma CRB.

6.4 Aplicación *multipath*

Podremos ejecutar la aplicación desde el lanzador (ver apartado 5.1) o bien ejecutando programa MP.m. En el apartado 5.5 hay una relación de los ficheros asociados a esta aplicación.

6.4.1 Descripción de la pantalla

Después de ejecutar el programa obtendremos la ventana con la siguiente imagen con 8 zonas definidas:

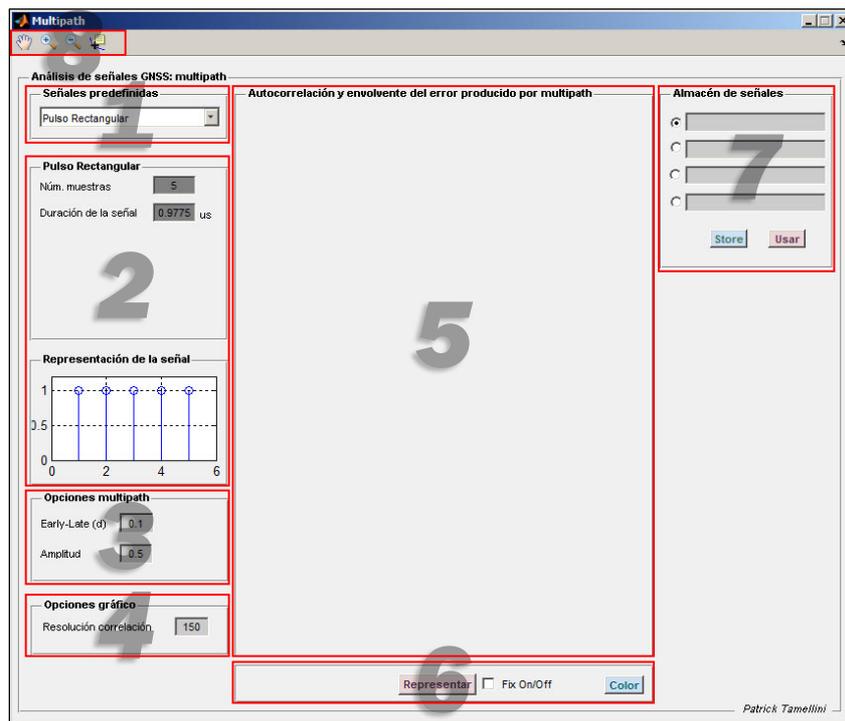


Figura 54: pantalla de inicio de la aplicación *multipath*.

- **Zona 1:** selector del tipo de señal.
- **Zona 2:** área descriptiva de la señal incluida una representación gráfica de sus valores temporales.
- **Zona 3:** opciones para la característica del *multipath*.
- **Zona 4:** opciones para la representación gráfica.
- **Zona 5:** área para la representación de la densidad correlación y la envolvente de error debido al *multipath*.
- **Zona 6:** controles para la representación

- **Zona 7:** zona para almacenar las señales de nuestro interés.
- **Zona 8:** herramientas para la manipulación de las gráficas.

6.4.2 Uso del programa

A continuación describimos el procedimiento general para la utilización del programa.

6.4.2.1 Selección de la señal

Lo primero que deberemos hacer es seleccionar una señal del desplegable de la zona 1 o bien hacer uso de la señal por defecto “Pulso Rectangular”.



Figura 55: señales disponibles

Podemos elegir de un elenco de 6 señales: pulso rectangular, raíz cuadrada coseno alzado (SQRRC), Binary Offset Carrier (BOC), Composite BOC (CBOC) y los casos particulares GPS-L1 y Galileo E1.

Dependiendo de la señal elegida el panel de la zona 2 que contiene los datos de la señal se modificará acordeamente.

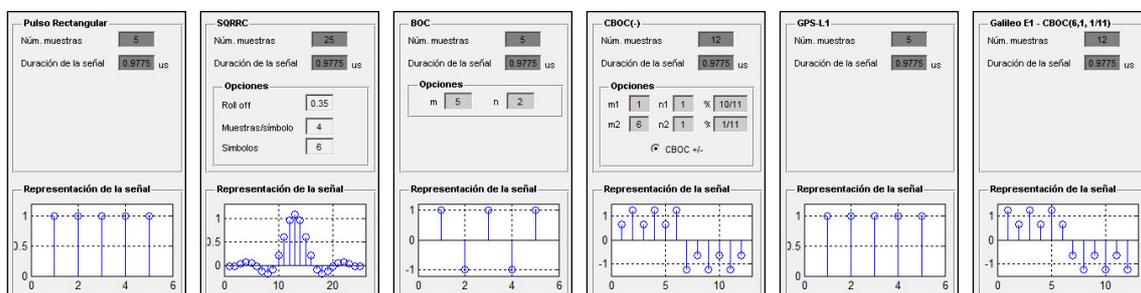


Figura 56: las diferentes señales con sus opciones. De izq. a der. pulso, SQRRC, BOC, CBOC, GPS L1 y Galileo E1.

Los elementos son autoexplicativos. La modificación de cualquiera de ellos lleva consigo la actualización de la representación temporal.

NOTA: Nótese que en la señal SQRRRC, símbolos se refiere a la cantidad de símbolos que conforman un pulso o *shape* y no al número de pulsos. En esta aplicación siempre usaremos un único pulso. Esto es, no usaremos secuencias de pulsos.

NOTA: si en una señal BOC o CBOC creamos una combinación no posible para m y n la GUI mostrará ** en el número de muestras y la representación temporal de la señal será nula.

NOTA: las señales GPS-L1 y Galileo E1 son casos concretos del pulso rectangular y CBOC respectivamente, por tanto no se pueden editar.

6.4.2.2 Opciones *multipath* y correlador

En esta aplicación se considera la presencia de un único *multipath* cuya variable es la amplitud. La otra opción es la apertura del discriminador (d). El valor máximo tanto para la apertura del discriminador como el *multipath* es 1. Es de consenso usar $d=0.1$ para discriminadores estrechos, $d=1$ para discriminadores anchos o estándar y una amplitud de 0.5 para el *multipath*.

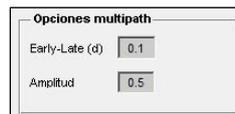


Figura 57: opciones *multipath* y apertura del correlador.

NOTA: según el tipo de señal, el uso de aperturas demasiado anchas y o demasiado estrechas, puede llevar a resultados gráficos inesperados. Ver discusión teórica.

6.4.2.3 Opciones gráficas

La única opción disponible es aumentar la resolución del cálculo de correlación que a su vez influye directamente con el cálculo de la envolvente de error *multipath*. Se

recomiendan valores iguales o superiores a los que establecidos por defecto. Un valor excesivamente bajo producirá unas gráficas muy escalonadas.

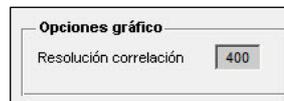


Figura 58: opciones gráficas.

6.4.2.4 Representación gráfica

Una vez que han sido seleccionadas y/o configuradas las señales podemos pasar a representar la función.



Figura 59: los botones para generar las representaciones.

Disponemos de 3 controles:

- **Representar:** genera las gráficas de la correlación de la señal y de la envolvente de error debido a la presencia del *multipath*. El color de la representación al usar este control es magenta.
- **Fix On/Off:** cuando está marcado, al representar una nueva señal, ésta se superpone a la anterior. Si no lo está, se hará una representación nueva.
- **Color:** sirve para cambiar el color de la última señal representada. El color que se usará será uno aleatorio. Si el color seleccionado no es de nuestro agrado basta con volver a pulsar el botón 'Color'.

NOTA: los botones 'Representar' y 'Color' son idénticos en su funcionalidad con la salvedad que el primer representa en color magenta y el segundo en un color aleatorio.

6.4.2.5 El almacén de señales

Si la **última** señal representada es de nuestro interés podremos reservarla en el almacén de señales. En éste podemos encontrar dos controles:

- **Store:** almacena la señal en el cajetín seleccionado. Podemos hacer esto aún cuando no hayamos representado la señal.

- **Usar:** vuelca los datos de la señal del cajetín seleccionado en la GUI y representa la señal.



Figura 60: el almacén con tres señales guardadas.

NOTA: al pulsar 'Usar' se representa la señal. Dependiendo el estado del control 'Fix On/Off' esta se superpondrá o no. En este último caso las señales no guardadas se perderán.

6.4.2.6 Herramienta de manipulación

Disponemos de cuatro herramientas para manipular las gráficas. Están orientadas a obtener datos específicos. La mano sirve para desplazar la gráfica, las lupas para acercar y alejar y la cruz para obtener valores concretos sobre la gráfica.



Figura 61: herramientas gráficas.

6.4.3 Interpretación de los resultados

Después de hacer varias representaciones podemos obtener una gráfica como la de la Figura 62. En la parte superior observamos las correlaciones de tres señales: un pulso BOC y dos pulsos rectangulares. En la inferior tenemos la envolvente de error en función del retardo con que llega el *multipath* al receptor.

Constatamos, que para una misma señal (pulso rectangular), el uso de discriminadores estrechos (azul) es una mejora sustancial para el error frente al discriminador ancho (magenta).

Así mismo constatamos que la señal BOC (celeste) tiene una leve mejora, especialmente en los retardos entorno al medio *chip*, frente al pulso rectangular (magenta) para una misma apertura del discriminador ($d=1$).

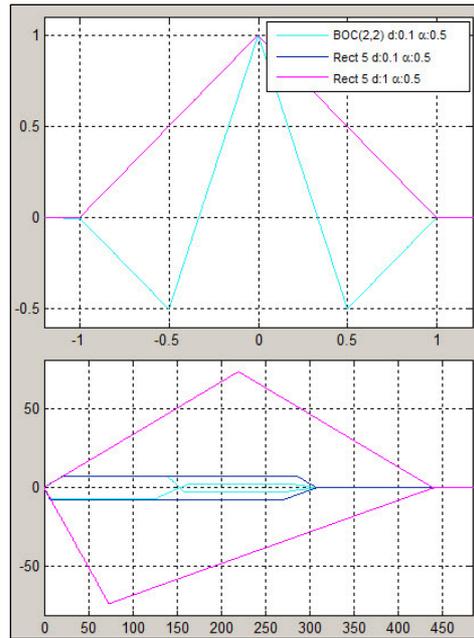


Figura 62: envolvente de error para varias señales y aperturas.

NOTA: tanto el retardo del *multipath* como el valor del error vienen expresados en metros. Se ha usado para la conversión la duración del pulso básico de una señal GPS-L1. Este dura 1/1023 ms, esto es, 293 metros.

6.5 Lista de ficheros necesarios para correr las aplicaciones

Para que las aplicaciones funcionen adecuadamente deberemos tener todos los ficheros que las componen. A continuación los listamos agrupados por aplicaciones.

Aplicación lanzador

launcher.fig launcher.m

Aplicación Función de Ambigüedad

FA.fig FA.m
representa_fa.m

Aplicación Cota Cramér-Rao

CCR.fig CCR.m
periodograma.m pulse_generator_ccr.m
crb2.m sqrrc_generator_ccr.m
representa.m boc_generator_ccr.m
store_ccr.m gps_L1_generator_ccr.m
usar_ccr.m El_generator_ccr.m
rotulos_ccr.m check_signal.ccr

Aplicación *multipath*

MP.fig MP.m
autocorrelador.m pulse_generator.m
mpath4.m sqrrc_generator.m
representa3.m boc_generator.m
store.m CBOC_generator.m
usar.m gps_L1_generator.m
check_signal.m El_generator.m
rotulos.m crossing.m

6.6 FAQs: Preguntas frecuentes

En este apartado respondemos a algunas de las preguntas que nos han hecho nuestro grupo de probadores de las aplicaciones.

6.6.1 FAQs Función de Ambigüedad

- *¿De verdad que tengo que introducir los datos por triplicado en los cajetines?*

No, solo debes introducir los datos de las características relevantes de la señal. Por ejemplo, si es un pulso rectangular sin modular deja sin marcar los botones de las fases y la modulación en frecuencia y sólo introduce los datos de las amplitudes.

- *Me da un error acerca de que las longitudes no coinciden.*

Te tienes que asegurar que metes la misma cantidad de datos en todos los cajetines.

6.6.2 FAQs Cota Cramér-Rao

- *Se me han acabado los cajetines y quiero almacenar una señal. ¿Qué hago?*

Basta con seleccionar un cajetín ocupado y guardar. La señal nueva sustituirá a la antigua.

- *Tengo demasiadas gráficas representadas. No encuentro el botón de borrar. ¿Cómo las borro?*

Deja el cajetín 'fix' sin marcar y representa la nueva función.

- *La leyenda en el área de representación de la CCR no me deja ver la gráfica.*

Pincha sobre la leyenda y colócala donde mejor te venga.

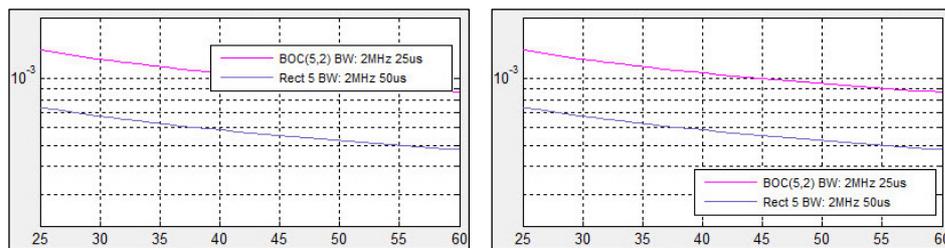


Figura 63: leyendas en el lugar más apropiado

- *No tengo leyenda en la representación de la densidad espectral, pero la necesito.*

Pincha sobre la leyenda de la CCR y arrástrala a donde más te convenga en el área de la DEP.

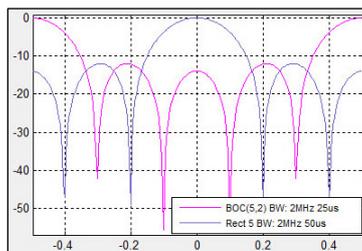


Figura 64: DEP con leyenda.

6.6.3 FAQs *multipath*

- ¿Por qué no puedo editar el tiempo de duración de la señal?

Para poder hacer una comparativa objetiva se debe hacer sobre la misma duración de chip. La hemos establecido en 1000 /1023 ms que es lo que dura un chip de una secuencia GPS-L1.

- ¿Por qué no puedo editar el número de muestras de cada señal?

Con excepción del pulso rectangular el número de muestras depende de los parámetros que configuran la señal. En cualquier caso no tiene ningún sentido aumentar las muestras ya que el valor de la envolvente no depende del número de estas.

- ¿Por qué me salen estas gráficas con tantos escalones?

Por que te falta resolución en la correlación. Basta con aumentarla. En la Figura 65 tenemos una envolvente de error con son resoluciones de 10 (gráfica morada) y 300 (gráfica magenta).

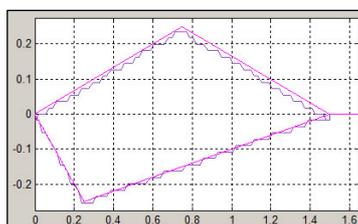


Figura 65: pulso rectangular con $d = 1$ y resolución 10 (morado) y resolución 300 (magenta)

- ¿Por qué obtengo estas gráficas con “manchurrones”?

Se debe a que la apertura del discriminador es excesiva para una señal con demasiados picos. Ver discusión teórica y algoritmos. La solución es cerrar el discriminador.

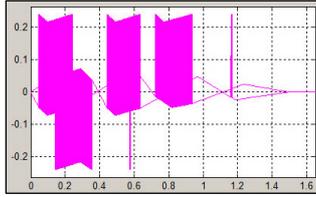


Figura 66: BOC(3,2) con apertura $d=1$.

- *Se me han acabado los cajetines y quiero almacenar una señal. ¿Qué hago?*

Basta con seleccionar un cajetín ocupado y guardar. La señal nueva sustituirá a la antigua.

- *Tengo demasiadas gráficas representadas. No encuentro el botón de borrar. ¿Cómo las borro?*

Deja el cajetín 'fix' sin marcar y representa la nueva función.

Bibliografía

[F. Antreich et al. 2008] F. Antreich, J. A. Nossek, J.L. Issler, Conference: *GNSS Signal Design Approach Considering Receiver Performance*. NAVITEC 2008, ESA-ESTEC Noordwijk, Países bajos.

[J. A. Ávila-Rodríguez, 2008] J. A. Ávila Rodríguez, Doctoral Thesis: *On Generalized Signal Waveforms for Satellite Navigation*, Luft- und Raumfahrttechnik der Universität der Bundeswehr, München, Alemania.

[J. A. Ávila-Rodríguez et al. 2006] J. A. Ávila-Rodríguez, S. Wallner, G. W. Hein, E. Rebeyrol, O. Julien, C. Macabiau, L. Ries, A. De Latour, L. Lestarquit, J.-L. Issler, *CBOC – An implementation of MBOC*. 2006, University FAF, Múnich, Alemania.

[J. A. Ávila-Rodríguez et al. 2006] J. A. Ávila Rodríguez, J. W. Betz, C. J. Hegarty, G. W. Hein, J. L. Issler, Capt. S. Lenahan, A. L. Kraay, J. R. Owen, A. R. Pratt, J. J. Rushanan, T. A. Stansell, S. Wallner, Conference: *MBOC: the new optimized spreading modulation recommended for Galileo E1 and GPS L1C*, NAVITEC 2006, ESA-ESTEC Noordwijk, Países bajos.

[J. Blunt, 2007] J. Blunt, Doctoral Thesis: *Advanced Global Navigation Satellite System Receiver Design*, 2007, University of Surrey, Reino Unido.

[K. Borre, 2009] K. Borre, Lecture at SU: *The E1 Galileo Signal*, 2009, Aalborg University, Aalborg, Dinamarca.

[M. S. Braasch, 1992] M. Braasch, Disertation: *On the Characterization of Multipath Errors in Satellite-Based Precision Approach and Landing Systems*, 1992, College of Engineering and Technology, Ohio University, Estados Unidos de América.

[R. G Brown et al, 1997] R. G. Brown, P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering, Third Edition*, 1997, John Wiley & Sons, Nueva York, Estados Unidos de América.

[S. Brueckener, 2007] S. Brueckner, *Zero Crossing function for MATLAB*, [recurso online] accedido, 10 de abril 2011, <http://www.mathworks.com/matlabcentral/fileexchange/2432>

[M. Budge, 2005] M. Budge, *Lecture notes for the course "Into to Radar Systems"*, 2005, University of Alabama. <http://www.ece.uah.edu/courses/material/EE619/>

[R. Canalda Pedrós, 2009] R. Canalda Pedrós, *Galileo Signal Generation – Simulation Analysis*, 2009, University of Limerick, Munster, República de Irlanda.

[F. Dovis et al. 2008] F. Dovis, L. Lo Presti, M. Fantino, P. Mulassano, J. Godet, Article: *Comparison between Galileo CBOC Candidates and BOC(1,1) in Terms of Detection Performance*. 2008, International Journal of Navigation and Observations, article ID 793868, Hindawi Publishing.

[M. Estévez Báez et al. 2008] M. Estévez Báez, A. Machado García, J. M. Estévez Carrera, *El periodograma como método para la estimación de la densidad espectral de potencia*, [recurso online] accedido 8 de abril 2011, http://fbio.uh.cu/vfc_docs/EIPeriodogramaComoMetodoParaEstimacionDensidadEspectralDePotencia.pdf

[D. A. García Álvarez, 2008] D. A. García Álvarez, Proyecto Fin de Carrera: *Sistema GNSS*, 2008, Universidad Autónoma de Madrid, Madrid, España.

[M. Irsigler et al. 2005] M. Irsigler, J. A. Ávila-Rodríguez, G. W. Hein, *Criteria for GNSS Multipath Performance Assessment*, 2005, Institute of Geodesy and Navigation, University FAF, Múnich, Alemania.

[M. Irsigler et al. 2003] M. Irsigler, B. Eissfeller, *Comparison of Multipath Mitigation Techniques with Consideration of Future Signal Structures*, 2003, Institute of Geodesy and Navigation, University FAF, Múnich, Alemania.

[M. Irsigler et al, 2004] M. Irsigler, B. Eissfeller, G. W. Hein, *Multipath Performance Analysis for Future GNSS Signals*, 2004, Institute of Geodesy and Navigation, University FAF, Múnich, Alemania.

[P. Joosten et al. 2003] P. Joosten, M. Irsigler, *GNSS ambiguity resolution in the presence of multipath*, 2003, Delft University of Technology, Delft, Países Bajos.

[E. D. Kaplan et al. 2006] E. D. Kaplan, C. Hegarty, *Understanding GPS – Principles and Applications, 2nd edition*, 2006, Artech House, INC., Noweood, MA, Estados Unidos de América.

[S. M. Kay, 1993] S. M. Kay, *Fundamentals of Statistical Signal Processing. Estimation Theory*, 1993, Prentice Hall, New Jersey, Estados Unidos de América.

[N. Levanon et al. 2002] N. Levanon, E. Mozeson, *MATLAB Code for Plotting Ambiguity Functions*, IEEE Transactions on Aerospace and Electronic Systems, vol. 38, no. 3, July 2002.

[B. R. Mahafza, 2009] B. R. Mahafza, *Radar Signal Analysis and Processing Using MATLAB*, 2009, CRC Press, Taylor & Francis Group LLC, Boca Ratón, Florida, USA.

[A. V. Oppenheim et al. 1998] A. V. Oppenheim, A. S. Willsky, S. H. Nawab, *Señales y Sistemas, Segunda Edición*, 1998, Prentice Hall Hispanoamericana S.A., Naucalpan de Juárez, México.

[OS IS ICD, 2010] *European GNSS (Galileo) Open Service, Signal In Space, Interface Control Document*, issue 1.1, 2010, European Commission.

[D. Plausinaitis, 2009] D. Plausinaitis, *Course Lectures: Code Tracking; Multipath*, 2009, Aalborg University, Aalborg, Dinamarca.

[D. Plausinaitis, 2009] D. Plausinaitis, *Course Lectures: GPS Signal Acquisition*, 2009, Aalborg University, Aalborg, Dinamarca.

[B. A. Siddiqui, 2009] B. A. Siddiqui, *Simulink-Based Acquisition Unit for Galileo E1 CBOC Modulated Signals*, 2009, Tampere University of Technology, Tampere, Finlandia.

[M. I. Skolnik, 1980] M. I. Skolnik, *Introduction to Radar Systems, Second Edition*, 1980, McGraw-Hill, Inc. Estados Unidos de América.

[J. Zhang et al., 2010] J. Zhang, E. S. Lohan, *Multi-correlator structures for tracking Galileo signals with CBOC and SinBOC(1,1) reference receivers and limited front-end bandwidths*, Positioning Navigation and Communication (WPNC), 2010 7th Workshop. Páginas 179-186, Department of Communication Engineering, Tampere University of Technology, Tampere, Finlandia.

[M. Zoltowsky, 2010] M. Zoltowsky, *Notes on Square-Root Raised-Cosine spectrum*, 2010, University of Purdue, Indiana, Estados Unidos de América.