



Treball Fi de Grau

Grau d'Enginyeria en Sistemes de Telecomunicació

**Simulación de las etapas de adquisición
y tracking de un receptor GPS**

Raúl Jaramillo Valbuena

Director: Gonzalo Seco Granados

Departament de Telecomunicació i Enginyeria de Sistemes

Escola d'Enginyeria

Universitat Autònoma de Barcelona (UAB)

Juny 2014



El tribunal d'avaluació d'aquest Treball Fi de Grau, reunit el dia 4 de juliol de 2014 ha acordat concedir la següent qualificació:

A large, empty rectangular box with a thin black border, likely intended for a signature or stamp.

President: Gonzalo Seco Granados

Vocal: José A. López-Salcedo

Secretari: Pedro de Paco Sánchez



Els sotasignant, Gonzalo Seco Granados, Professor de l'Escola d'Enginyeria de la Universitat Autònoma de Barcelona (UAB),

CERTIFICA:

Que el projecte presentat en aquesta memòria de Treball Fi de Grau ha estat realitzat sota la seva direcció per l'alumne Raúl Jaramillo Valbuena.

I, perquè consti a tots els efectes, signa el present certificat.

Bellaterra, 4 de juliol de 2014.

Signatura:

Simulación de las etapas de adquisición y tracking de un receptor GPS

Raúl Jaramillo Valbuena

Junio de 2014, Bellaterra

Este proyecto es el final de una etapa. Es por ello que quiero agradecer a Gonzalo por su paciencia y dedicación durante estos meses, a mis padres por la educación que he recibido, a mis abuelos por su amor incondicional y a ti por creer en mí cuando perdido te encontré.

No es más fuerte quien menos veces cae, sino quien más veces se levanta

Índice

Índice	i
Lista de Figuras	iv
Lista de Tablas	viii
Lista de Siglas	ix
CAPÍTULO 1 INTRODUCCIÓN	1
1.1. Motivación del proyecto	1
1.2. Objetivos del proyecto	1
1.3. Descripción y estructura del documento	2
CAPÍTULO 2 SISTEMAS DE RADIONAVEGACIÓN	4
2.1. Sistemas de radionavegación terrestre	5
2.1.1. Radiofaros	6
2.1.2. Sistemas de navegación hiperbólicos	7
2.1.3. Sistemas de aproximación y aterrizaje	8
2.2. Sistemas de radionavegación por satélite	9
2.3. Clasificación de los sistemas de radionavegación	10
CAPÍTULO 3 FUNDAMENTOS DEL SISTEMA GPS	11
3.1. Mecánica orbital	12
3.1.1. Elipsoide de referencia	12
3.1.2. Sistemas de coordenadas	13
3.1.2.1. Coordenadas geográficas	13
3.1.2.2. Coordenadas cartesianas	14
3.1.3. Leyes de Kepler	15
3.1.4. Parámetros de orbitales	16
3.2. Arquitectura del Sistema GPS	18
3.2.1. Segmento espacial	18
3.2.2. Segmento de control	20
3.2.3. Segmento de usuario	21
3.3. Cálculo de la posición	22

CAPÍTULO 4 LA SEÑAL GPS	26
4.1. La señal GPS	26
4.2. Frecuencia portadora.....	26
4.3. Mensaje de navegación	27
4.4. Código C/A	31
4.4.1. Generación del código C/A.....	33
4.4.2. Propiedades de correlación del código C/A	34
4.5. Estructura de la señal GPS	36
4.6. Transmisión de la señal GPS	38
4.6.1. Modulación DSSS.....	38
4.6.2. Técnica de acceso al medio CDMA.....	40
CAPÍTULO 5 IMPLEMENTACIÓN DEL RECEPTOR GPS	42
5.1. Bloques de un receptor GPS	42
5.2. Front-End	43
5.3. Módulo de adquisición	44
5.3.1. Remuestreo de la señal réplica	44
5.3.2. Búsqueda bidimensional en tiempo y frecuencia	46
5.3.2.1. Dos periodos de la señal GPS y uno de la réplica.....	48
5.3.2.1.1. Señal GPS simulada	52
5.3.2.1.2. Señal GPS real	53
5.3.2.2. Un periodo de la señal GPS y dos de la réplica	55
5.3.2.2.1. Señal GPS simulada	56
5.3.2.2.2. Señal GPS real	57
5.3.4. Módulo de tracking.....	59
5.4.1. Delay Lock Loop. DLL.....	60
5.4.1.1. Generación de x_{DLL}	61
5.4.1.2. Módulo Correlador	61
5.4.1.3. Discriminador DLL	62
5.4.1.4. Filtro de lazo DLL.....	62
5.4.1.4.1. Filtro de lazo DLL de orden 0.....	63
5.4.1.4.2. Filtro de lazo DLL de orden 1.....	64
5.4.1.5. NCO _{DLL}	65
5.4.1.6. Generador de secuencia	65
5.4.1.7. DLL. Ejemplo de funcionamiento	68
5.4.2. Phase Lock Loop. PLL.....	74
5.4.2.1. Discriminador PLL	75
5.4.2.2. Filtro de lazo PLL	77
5.4.2.3. NCO _{PLL}	77
5.4.2.4. Generador de exponencial.....	77
5.4.2.5. PLL. Ejemplo de funcionamiento	77
5.5. Módulo de navegación	82

CAPÍTULO 6 ANÁLISIS DEL FUNCIONAMIENTO DEL RECEPTOR BAJO EFECTOS REALES	84
6.1. Efecto Doppler	84
6.1.1. Frecuencia Doppler	86
6.1.2. Estudio del sistema	88
6.1.2.1. Implementación de la señal GPS.....	88
6.1.2.2. Implementación del receptor	88
6.1.3. Análisis de la fase de la señal GPS	89
6.1.4. Cálculo del pseudorrango con la arquitectura DLL	91
6.1.4.1. Dinámica de un tramo	92
6.1.4.2. Dinámica de dos tramos	93
6.1.4.3. Dinámica de tres tramos.....	95
6.1.5. Cálculo del observable de fase con la arquitectura PLL	98
6.1.5.1. Dinámica de un tramo	99
6.1.5.2. Dinámica de dos tramos	101
6.1.5.3. Dinámica de tres tramos.....	105
6.1.6. Comparativa entre DLL y PLL.....	108
6.2. Filtrado	110
6.2.1. Filtro utilizado	111
6.2.2. Simulación de filtrado	114
6.3. Multipath	120
6.3.1. Simulación de retardo	121
6.3.2. Simulación de retardo, fase y amplitud	123
6.3.3. Simulación de retardo y amplitud	125
6.4. Ruido	127
6.4.1. Modelo de ruido.....	127
6.4.2. Simulación de matriz de correlación	129
6.4.3. Simulación de envolventes.....	130
6.4.4. Simulación de fase estimada	132
6.5. Efecto combinado	133
CAPÍTULO 7 CONCLUSIONES	139
ANEXO Código MATLAB®	140
Bibliografía	165

Lista de Figuras

Fig. 2.1. Esquema de funcionamiento a alto nivel de los sistemas de radionavegación terrestre.....	5
Fig. 3.1. Elipsoide de referencia.....	12
Fig. 3.2. Sistema de coordenadas geográficas.....	14
Fig. 3.3. Sistema de coordenadas cartesianas	14
Fig. 3.4. Parámetros orbitales que permiten posicionar un satélite en el espacio	17
Fig. 3.5. Arquitectura del Sistema GPS	18
Fig. 3.6. Constelación base del Sistema GPS.....	19
Fig. 3.7. Distribución geográfica del segmento de control del Sistema GPS	21
Fig. 3.8. Cálculo de la posición en 2D únicamente	22
Fig. 3.9. Cálculo de la posición en 2D con ambigüedad.....	23
Fig. 3.10. Principio de funcionamiento del Sistema GPS para el cálculo de la posición	24
Fig. 4.1. Forma temporal del mensaje de navegación	28
Fig. 4.2. Espectro del mensaje de navegación	28
Fig. 4.3. Estructura general del mensaje de navegación	29
Fig. 4.4. Forma temporal de los 10 primeros chips del código PRN 12	32
Fig. 4.5. Espectro del código C/A del satélite asociado al PRN 12	32
Fig. 4.6. Esquema de generación de un código C/A	33
Fig. 4.7. Correlación cruzada del código C/A entre los satélites 12 y 29	35
Fig. 4.8. Autocorrelación del código C/A del satélite 12.....	35
Fig. 4.9. Esquema de transmisión de la señal GPS.....	36
Fig. 4.10. Forma temporal comparativa de las señales que forman la señal GPS	37
Fig. 4.11. Proceso de modulación de la señal GPS	39
Fig. 4.12. Esquema de transmisión a alto nivel de la señal GPS	41
Fig. 4.13. Esquema de recepción a alto nivel de la señal GPS	41
Fig. 5.1. Las señales GPS transmitida y recibida no son exactamente iguales	42
Fig. 5.2. Arquitectura de un receptor GPS.....	42
Fig. 5.3. Esquema funcional de la salida del <i>front-end</i> de un receptor software	43
Fig. 5.4. Diagrama de bloques de la etapa de <i>front-end</i>	43
Fig. 5.5. Operaciones del módulo de adquisición	47
Fig. 5.6. Esquema en muestras de la señal réplica perfectamente alineada.....	48
Fig. 5.7. Esquema en muestras de la señal GPS desalineada	48
Fig. 5.8. Esquema del primer estado de la convolución	49

Fig. 5.9. Esquema del segundo estado de la convolución	49
Fig. 5.10. Esquema del tercer estado de la convolución	50
Fig. 5.11. Ejemplo de retraso de código igual a 5.....	51
Fig. 5.12. Adquisición positiva del satélite asociado al código PRN 12	52
Fig. 5.13. Adquisición fallida del satélite asociado al código PRN 12	53
Fig. 5.14. Adquisición positiva del satélite asociado al código PRN 12	54
Fig. 5.15. Adquisición fallida del satélite asociado al código PRN 12	54
Fig. 5.16. Esquema en muestras de la señal réplica perfectamente alineada.....	55
Fig. 5.17. Esquema en muestras de la señal GPS desalineada	55
Fig. 5.18. Adquisición positiva del satélite asociado al código PRN 12	56
Fig. 5.19. Adquisición fallida del satélite asociado al código PRN 12	56
Fig. 5.20. Adquisición positiva del satélite asociado al código PRN 12	57
Fig. 5.21. Adquisición fallida del satélite asociado al código PRN 12	57
Fig. 5.22. Diagrama de bloques de la arquitectura DLL.....	60
Fig. 5.23. Módulo correlador del DLL	62
Fig. 5.24. Esquema de un filtro de lazo de orden 1	63
Fig. 5.25. Esquema de un filtro de lazo de orden 0	63
Fig. 5.26. Efecto en las envolventes del remuestreo correcto del código	67
Fig. 5.27. Magnitud de las envolventes con un seguimiento perfecto de retraso de código	70
Fig. 5.28. Magnitud de las envolventes en función del retraso en chips	71
Fig. 5.29. Triángulo de correlaciones de las envolventes	72
Fig. 5.30. Entrada y salida del filtro de lazo DLL	73
Fig. 5.31. Arquitectura completa del módulo de tracking	74
Fig. 5.32. Discriminador PLL de fase tipo Costas	76
Fig. 5.33. Entrada al discriminador PLL.....	78
Fig. 5.34. Entrada y salida del filtro de lazo PLL de orden 0	79
Fig. 5.35. Salida del NCO_{PLL} con la entrada de un filtro de lazo PLL de orden 0.....	79
Fig. 5.36. Entrada y salida del filtro de lazo PLL de orden 1	80
Fig. 5.37. Salida del NCO_{PLL} con la entrada de un filtro de lazo PLL de orden 1.....	80
Fig. 5.38. Ilustración de la solución de navegación con error del receptor en 2D	82
Fig. 6.1. Movimiento del satélite respecto al receptor	84
Fig. 6.2. Variación de la frecuencia percibida por el receptor.....	85
Fig. 6.3. Variación real de distancia (izquierda), variación con el DLL (derecha).....	92
Fig. 6.4. Variación real de distancia (izquierda), variación con el DLL (derecha).....	94
Fig. 6.5. Variación real de distancia (izquierda), variación con el DLL (derecha).....	96

Fig. 6.6. Fase estimada por el NCO _{PLL}	99
Fig. 6.7. Variación real de distancia (izquierda), variación con el PLL (derecha)	100
Fig. 6.8. Fase estimada por el NCO _{PLL}	102
Fig. 6.9. Variación real de distancia (izquierda), variación con el PLL (derecha)	104
Fig. 6.10. Fase estimada por el NCO _{PLL}	105
Fig. 6.11. Variación real de distancia (izquierda), variación con el PLL (derecha)	107
Fig. 6.12. Espectro de la señal GPS	110
Fig. 6.13. Rango de frecuencias tras el muestreo.....	111
Fig. 6.14. Espectro de la señal GPS en banda base muestreada	111
Fig. 6.15. Respuesta frecuencial del filtro paso bajo	112
Fig. 6.16. Espectro de la señal GPS antes y después de aplicar el filtrado	113
Fig. 6.17. Magnitud de las envolventes sin filtro (izquierda), y con filtro (derecha)	114
Fig. 6.18. Triángulo de correlaciones sin filtro (izquierda), y con filtro (derecha)	115
Fig. 6.19. Entrada y salida del filtro de lazo DLL sin filtro (superior) y con filtro (inferior).....	116
Fig. 6.20. Entrada al discriminador PLL sin filtro (superior), y con filtro (inferior)	117
Fig. 6.21. Entrada y salida del filtro de lazo PLL de orden 1 sin filtro (superior) y con filtro (inferior).....	118
Fig. 6.22. Fase estimada sin desviación Doppler sin filtro (izquierda) y con filtro (derecha)	118
Fig. 6.23. Fase estimada con desviación Doppler sin filtro (izquierda) y con filtro (derecha)	119
Fig. 6.24. Variación real de distancia (izquierda), variación filtrada con el PLL (derecha)	119
Fig. 6.25. Esquema de la propagación multicamino.....	120
Fig. 6.26. Triángulo de correlaciones de la señal directa y de la señal réplica	122
Fig. 6.27. Triángulo de correlaciones de la suma de señales.....	122
Fig. 6.28. Triángulo de correlaciones de la señal directa y de la señal réplica	124
Fig. 6.29. Triángulo de correlaciones de la suma de señales.....	124
Fig. 6.30. Triángulo de correlaciones de la señal directa y de la señal réplica	125
Fig. 6.31. Triángulo de correlaciones de la suma de señales.....	126
Fig. 6.32. Matriz de adquisición. De izquierda a derecha y de arriba a abajo, con valores de <i>SNR</i> de $\infty, -10, -20$, y -27 dB	129
Fig. 6.33. Magnitud de las envolventes sin ruido (izquierda), y para una <i>SNR</i> = -10 dB (derecha)	130
Fig. 6.34. Triángulo de correlaciones sin ruido (izquierda), y para una <i>SNR</i> = -10 dB (derecha)	130
Fig. 6.35. Triángulo de correlaciones para una <i>SNR</i> = -20 dB (izquierda), y para una <i>SNR</i> = -27 dB (derecha)	131
Fig. 6.36. Estimación de fase del PLL. De izquierda a derecha y de arriba a abajo, con valores de <i>SNR</i> de $\infty, -10, -20$, y -27 dB	132

Fig. 6.37. Matriz de correlación para simular el efecto multicamino.....	133
Fig. 6.38. Magnitud de las envolventes de los correladores	134
Fig. 6.39. Triángulo de correlación de las envolventes.....	135
Fig. 6.40. Salida del discriminador y salida del filtro de lazo DLL	135
Fig. 6.41. Entrada al discriminador PLL.....	136
Fig. 6.42. Fase estimada por el NCO _{PLL}	136
Fig. 6.43. Variación real de distancia (izquierda), variación con el PLL (derecha).....	137

Lista de Tablas

Tabla 2.1. Clasificación de los sistemas de radionavegación.....	10
Tabla 4.1. Estructura de las tramas del mensaje de navegación	29
Tabla 4.2. Identificación de pares de posiciones para generar G2	34
Tabla 5.1. Rango de muestras de la señal x_{DLL} , señal GPS y exponencial	61
Tabla 6.1 Comparación de f_{dop} : real y estimada	93
Tabla 6.2 Comparación de f_{dop} : real y estimada	95
Tabla 6.3 Comparación de f_{dop} : real y estimada	97
Tabla 6.4 Comparación de f_{dop} : real y estimada	101
Tabla 6.5 Comparación de f_{dop} : real y estimada	104
Tabla 6.6 Comparación de f_{dop} : real y estimada	108
Tabla 6.7. Comparativa de simulaciones.....	108

Lista de Siglas

ADC	Analog to Digital Converter	MEO	Medium Earth Orbit
ADF	Automatic Direction Finder	MLS	Microwave Landing System
AWGN	Additive White Gaussian Noise	MS	Monitor Stations
C/A	Coarse Acquisition Code	NCO	Numerically Controlled Oscillator
CDMA	Code Division Multiple Access	NDB	Non-Directional Beacon
DLL	Delay Lock Loop	P	Encrypted Precision Code
DME	Distance Measurement Equipment	PLL	Phase Lock Loop
DSSS	Direct Sequence Spread Spectrum	PRN	Pseudo-random Noise.
DTS	Data Transmission Stations	QZSS	Quasi Zenith Satellite System
GEO	Geosynchronous Earth Orbit	RNSS	Regional Navigation Satellite System
GLONASS	Global Navigation Satellite System	SNR	Signal to Noise Ratio
GNSS	Global Navigation Satellite System	TACAN	Tactical Air Navigation
GPS	Global Positioning System	TLM	TeLeMetry
HOW	HandOver Word	TOW	Time Of Week
ILS	Instrument Landing System	TRSB	Time Reference Scanning-Beam
IRNSS	Indian Regional Navigation Sat System	UHF	Ultra High Frequency
LEO	Low Earth Orbit	UTC	Universal Coordinated Time
LORAN	Long Range Navigation	VOR	Very High Frequency Omnidirectional Range
MCS	Master Control Station		

1. Introducción

1.1. Motivación del proyecto

La idea para la realización de este proyecto nace de mi profundo interés por las comunicaciones por satélite: ¿cómo funcionan?, ¿cómo es posible que se pueda recibir la señal de interés a tantos miles de kilómetros de distancia?, ¿cómo se determina la posición de un usuario mediante la señal que recibe del satélite?, ¿cuáles son los parámetros clave usados en este tipo de comunicación? El mundo de las comunicaciones por satélite es amplio y complejo, y este proyecto me ha dado la oportunidad de iniciarme en él, respondiendo parcialmente a esas y otras preguntas que han ido surgiendo a medida que avanzaba el estudio.

La radionavegación por satélite está en auge y diversos países están apostando por lanzar en órbita sus propios satélites para disponer de su propio sistema de posicionamiento. Pero la localización de usuarios es solo una de las múltiples aplicaciones que tiene esta tecnología, su potencial en la navegación, sincronización, rastreo y toma de datos topográficos y geodésicos, entre otras aplicaciones, abre un gran abanico de posibilidades para desarrollar una carrera profesional en el sector.

De esta manera, se ha unido mi interés tecnológico junto con las oportunidades de negocio que he podido descubrir a medida que avanzaba en el estudio. Todo ello siendo muy consciente de la dificultad que entraña labrarse una carrera en el mundo de las comunicaciones por satélite, y teniendo claro que este proyecto solo es el punto de partida para profundizar en los GNSS (*Global Navigation Satellite Systems*) durante mi formación académica venidera.

1.2. Objetivos del proyecto

El principal objetivo del proyecto es aprender cómo funciona un receptor GPS, implementarlo, y hacer experimentos con efectos reales para entender su funcionamiento. Para conseguirlo, es necesario la consecución de diversos objetivos secundarios: generar la señal GPS, estimar y refinar sus parámetros retraso de código y desviación Doppler una vez la señal llegue al receptor, simular diversos efectos reales que experimenta la señal GPS, y finalmente calcular la distancia entre satélite y receptor. El cálculo de esta distancia se entregará a la última etapa del receptor, la cual no se implementará, para que resuelva la posición del usuario mediante un sistema de ecuaciones.

1.3. Descripción y estructura del documento

Este proyecto es el resultado final de las asignaturas *Treball de Fi de Grau y Projecte Avançat d'Enginyeria*, de 12 ECTS cada una, por lo que engloba una carga de 24 ECTS. La fase previa de lectura de bibliografía empezó en octubre de 2013 y el proyecto en sí se ha desarrollado desde noviembre de ese mismo año hasta mayo de 2014.

El presente documento se divide en 7 capítulos. Cada uno de ellos tiene sus correspondientes secciones formadas por apartados. La nomenclatura que se utiliza para definir la estructura del documento a partir del cuarto nivel es la de subapartado.

El primer capítulo, en el que están escritas estas líneas, tiene como objetivo dar una visión global del proyecto, responder al porqué de su realización, mostrar los objetivos que pretende conseguir y proporcionar las herramientas básicas para obtener una lectura amena y provechosa del documento. Después del capítulo introductorio, un segundo capítulo ofrece un breve análisis de la historia y clasificación de los sistemas de radionavegación, contextualizando el Sistema GPS.

El tercer capítulo introduce las técnicas para situar a los satélites en el espacio y a los usuarios en la superficie terrestre, presenta la arquitectura del Sistema GPS, y ofrece, a alto nivel, la manera de posicionar a un usuario mediante el cálculo de la distancia que le separa de los satélites. Una vez finalizada la lectura del tercer capítulo se habrán sentado las bases para adentrarnos en el estudio del Sistema GPS.

El cuarto capítulo presenta las componentes y la estructura de la señal GPS que envía el satélite. Es aquí donde se introducen señales y conceptos tan importantes como el código C/A, el mensaje de navegación, la frecuencia portadora, la modulación DSSS o la técnica de acceso al medio CDMA.

En el quinto capítulo se lleva a cabo el procesado de la señal GPS analizada en el capítulo anterior. El dispositivo que realiza esta tarea es el receptor, del cual se presentarán sus distintos módulos, haciendo especial hincapié en la adquisición y el tracking, palabras que dan nombre al proyecto. En este capítulo se presentarán diferentes simulaciones y gráficas explicativas para entender el funcionamiento del receptor.

El sexto capítulo analiza el comportamiento del receptor bajo efectos reales, tales como la frecuencia Doppler, *multipath*, filtrado y ruido. Mostraremos y explicaremos las diferencias existentes con las simulaciones observadas en el capítulo anterior, y calcularemos la distancia entre el satélite y el receptor, parámetro que usará el módulo de navegación para resolver la posición del usuario.

El capítulo final presenta las conclusiones a las que hemos llegado tras el análisis realizado en el receptor. Analizaremos si se han cumplido los objetivos planteados al inicio del proyecto, mostraremos posibles áreas de mejora dentro del propio diseño del receptor y trazaremos las posibles líneas de investigación que se pueden seguir tomando como punto de partida la realización de este proyecto. Para finalizar el documento se presentará la bibliografía, el código MATLAB® utilizado durante el proyecto y un breve resumen del mismo.

Cabe destacar que hemos simulado la señal GPS implementándola a la salida del primer módulo del receptor, el *front-end*. De esta manera, generamos un flujo constante de muestras que recogen la estructura de la señal GPS y los efectos que se habrían producido en ella durante su viaje por el espacio hasta llegar al receptor. A partir de esta señal obtenemos todas las simulaciones posteriores, si bien es cierto que también trabajaremos con muestras de señales reales captadas por un receptor GPS concreto. Todas las simulaciones son originales y se han generado mediante el software MATLAB®.

Las funciones que se han implementado para realizar la simulación del receptor GPS se muestran en el capítulo Anexo. El punto de partida para la simulación son las funciones *main_ideal.m*, si el efecto Doppler se tiene en cuenta solo en la portadora, y *main_fchip.m*, si el efecto Doppler se simula en toda la señal GPS. De las dos funciones principales anteriormente mencionadas cuelgan las funciones *cacode.m*, para generar el código C/A, y las funciones *adq_1Inc2CA.m* y *adq_2Inc1CA.m*, que simulan el módulo de adquisición. El tracking las realizan las funciones *track_ideal2.m* a partir de *main_ideal.m*, y *track_fchip2.m* y *bucle_tracking2.m* a partir de *main_fchip.m*. Finalmente, la función *graficas.m* se utiliza para generar las gráficas del cuarto capítulo.

Aunque al final del documento mostraremos toda la bibliografía relacionada con el proyecto, destacamos que parte de los capítulos descriptivos del proyecto, segundo y tercero, se han sintetizado a partir del extenso material docente de la asignatura de la UOC, Sistemas de radionavegación, impartida por José A. López-Salcedo, José López Vicario y Gonzalo Seco Granados, así como explicaciones detalladas de las etapas de adquisición y tracking. Otros libros de referencia utilizados para entender el funcionamiento del receptor han sido, como se les conoce en la jerga GPS, Tsui y Borre.

La realización del proyecto engloba conocimientos estudiados en varias asignaturas de la carrera, tales como Fundamentos de Comunicaciones, Fundamentos de Señales y Sistemas, Señales y Sistemas Discretos, Tratamiento Digital de la Señal, Teoría de la Comunicación, Radiación y Ondas Guiadas, y Transmisores y Receptores de Telecomunicaciones.

Así acabamos el capítulo introductorio, esperamos que pueda disfrutar de una lectura amena y ágil, pudiendo ir a los capítulos que más le interesen siguiendo la estructura presentada en esta sección.

2. Sistemas de radionavegación

Los sistemas de radionavegación modernos han sido consecuencia de un proceso milenario de investigación. Durante siglos se utilizó el sol y las estrellas como referencia para medir la posición relativa de personas u objetos, y las técnicas de navegación basadas en la observación del cielo y en la brújula avanzaron significativamente durante la exploración de los océanos en los siglos XV y XVI.

Este fue un periodo de tiempo fértil para la historia de los sistemas de posicionamiento, ya que se desarrollaron sistemas como el astrolabio, que permitieron a los navegantes leer la posición de las estrellas y, a partir de ahí, calcular su propia posición a través de los mapas de los que disponían en la época. Durante el siglo XVI también aparecieron técnicas predictivas para calcular posiciones futuras a partir de movimientos pasados, muy útiles en la navegación cuando no había visibilidad de las estrellas porque las nubes lo impedían.

Este tipo de técnicas fueron mejorando hasta el siglo XVIII, en el que se produjo el primero de los dos grandes puntos de inflexión en los sistemas de radionavegación: la posibilidad de medir el tiempo. Se inventó el cronómetro, que se utilizó ampliamente en la navegación marítima.

No fue hasta finales del siglo XIX cuando se produjo el segundo punto de inflexión, que daría un giro de tuerca a los sistemas de radionavegación tal y como se conocían en la época: se descubrieron las ondas electromagnéticas, que se han convertido en la base de los actuales sistemas de radionavegación basados en comunicaciones inalámbricas.

El descubrimiento de las ondas electromagnéticas y la posibilidad de medir el tiempo abrió un mundo de posibilidades para el estudio de nuevas técnicas de posicionamiento durante el siglo pasado, y los inventos en la materia se fueron sucediendo uno tras otro.

A mediados del siglo XX se inventaron los radares y con ellos los sistemas de navegación terrestre. En esa época se produjeron grandes avances en la tecnología y técnicas de posicionamiento, y la radionavegación terrestre eclosionó con el sistema LORAN, activo hasta 2010, que proporcionaba una excelente precisión gracias a la transmisión de señales por parte de varias estaciones de manera síncrona, lo que permitía al receptor medir la diferencia de distancias a partir de pares de estaciones.

A partir de la década de los sesenta empezaron a desarrollarse los primeros sistemas de radionavegación por satélite, conocidos como GNSS (*Global Navigation Satellite Systems*). Estos últimos son ampliamente utilizados ya que únicamente requieren visibilidad del cielo y un receptor para situar su posición. Uno de los puntales de los GNSS ha sido el Sistema GPS (*Global Positioning System*), declarado operacional a finales de los noventa. El Sistema GPS revolucionó la radionavegación, ya que a partir de señales electromagnéticas transmitidas a largas distancias es capaz de obtener en cualquier momento la posición de cualquier lugar terrestre con visibilidad del cielo utilizando un receptor económico. A partir de la información de localización que generalizó el Sistema GPS entre la población, surgieron diversas aplicaciones que nos resultan muy familiares hoy en día, como la navegación terrestre o el mapeado preciso de la Tierra.

2.1 Sistemas de radionavegación terrestre

En este apartado introduciremos brevemente los sistemas de radionavegación terrestre con la voluntad de contextualizar el camino que ha seguido la radionavegación, y ser capaces de entender el gran avance que posteriormente se ha logrado con los sistemas basados en satélites.

Como hemos comentado anteriormente, los primeros sistemas de radionavegación fueron los terrestres, precursores de los modernos sistemas actuales. Este tipo de sistemas utilizan señales de radiofrecuencia transmitidas desde equipos terrestres, es decir, el emisor se encuentra ubicado sobre la superficie de la Tierra. La ubicación de las estaciones radio emisoras es conocida y a partir de ella se calculan los diferentes parámetros de posicionamiento del blanco.

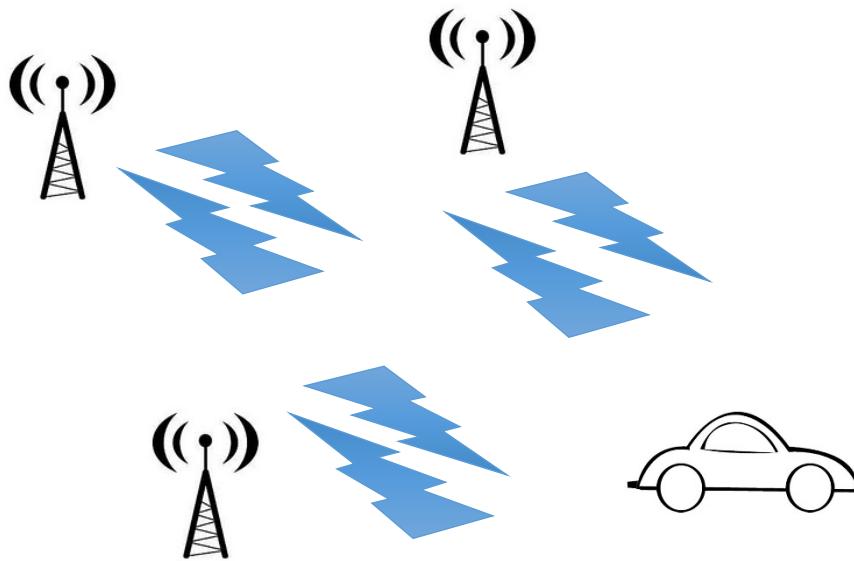


Fig. 2.1. Esquema de funcionamiento a alto nivel de los sistemas de radionavegación terrestre

En el esquema anterior (Fig. 2.1) podemos observar un esquema básico para entender el principio elemental de funcionamiento de los sistemas de navegación terrestre. Sin perder de vista el objetivo final, que no es otro que el de determinar la posición del móvil, las estaciones radio actúan como transmisoras y/o receptoras, procesando en cada caso las señales radio que se intercambian con el móvil para hallar los parámetros de interés.

A modo de ejemplo, podemos clasificar los sistemas de radionavegación terrestre en tres grandes grupos: los radiofaros, los sistemas de radionavegación hiperbólicos y los sistemas de aproximación y aterrizaje.

2.1.1 Radiofaros

Los radiofaros son estaciones terrestres con una ubicación fija que transmiten señales radio de forma automática con el objetivo de ayudar a la navegación aérea y marítima. A continuación ofreceremos un breve repaso a los tipos de radiofaros existentes:

NDB (*Non-Directional Beacon*), Baliza no direccional. Es el tipo de radiofaro más antiguo. El radiofaro emite una portadora omnidireccional, es decir, transmite una frecuencia fija con una potencia igual en todas las direcciones, para que pueda ser captada en recepción por cualquier equipo situado en su radio de alcance, unos 50 kilómetros. Dado que la emisión es omnidireccional, el cálculo de la posición del móvil la lleva a cabo el propio receptor mediante el sistema ADF¹.

VOR (*Very High Frequency Omnidirectional Range*). Es la evolución de la baliza no direccional y es el sistema más utilizado alrededor del mundo. En este caso la señal enviada por el radiofaro es direccional. Para ello, se utilizan dos antenas: una no direccional (como en el caso del NDB), y otra direccional formada por dos aristas desfasadas 90° entre sí.

DME (*Distance Measurement Equipment*). El equipo de medida de la distancia se basa en la tecnología radar para medir la distancia del blanco a partir del tiempo de propagación de un pulso en retornar al propio blanco. Es decir, se mide el lapso de tiempo en el que el pulso hace el recorrido equipo – radiofaro – equipo, y a partir de este tiempo se calcula la distancia al radiofaro.

TACAN (*Tactical Air Navigation*). La navegación aérea táctica se utiliza para dar servicio a equipos militares, tales como barcos o aviones. TACAN tiene dos funcionalidades básicas: la obtención de la distancia, utilizando DME; y la obtención de la orientación del equipo, a partir de VOR.

CONSOL. Se basa en la obtención de la orientación del equipo utilizando una agrupación de tres antenas. Primero, la antena central emite el código de identificación del radiofaro en morse y, posteriormente, las tres antenas envían la señal de radionavegación. Luego vuelve a repetirse el ciclo.

¹ ADF son las siglas de *Automatic Direction Finder*, sistema que permite estimar la posición del blanco determinando la dirección del radiofaro. Esto es posible gracias a la obtención de la dirección de propagación de la onda electromagnética midiendo la diferencia de potencial en los extremos de la antena de cuadro incorporada al sistema.

2.1.2 Sistemas de radionavegación hiperbólicos

Los sistemas de navegación hiperbólicos son aquellos que toman como medida la diferencia de tiempos de llegada de dos señales recibidas, enviadas desde dos transmisores diferentes. En el espacio, todos los puntos que presentan un mismo retardo temporal forman una hipérbola, figura geométrica que da nombre al sistema de radionavegación. Algunos de los más conocidos son los siguientes:

DECCA. Este sistema utiliza una estación maestra y tres esclavas, definiendo tres hipérboles. El punto de intersección entre ellas nos determinará la posición del blanco. Cada grupo de cuatro estaciones se denomina cadena, y cada cadena tiene una frecuencia base asignada. Cada una de las estaciones transmite a una frecuencia diferente, siempre múltiplo de la frecuencia base. Se determinan pares maestra-esclava con el objetivo de calcular la diferencia de fase entre señales, parámetro que se utilizará en la carta de navegación para identificar la hipérbola que describe la posición del blanco.

LORAN. Es un tipo de radionavegación de largo alcance más robusta que el sistema DECCA. Se han ido implementando diferentes versiones: LORAN-A, LORAN-B, LORAN-C y LORAN-D. Las más utilizadas fueron LORAN-A y LORAN-C. Por ejemplo, el sistema LORAN-A está formado por cadenas de una estación maestra y dos esclavas, y las diferencias de tiempos se toman entre una maestra y una esclava. Este sistema consiste en modular una portadora con una serie de pulsos. Estos pulsos son utilizados para medir la diferencia de tiempos de propagación entre dos señales recibidas, cada una de ellas transmitida desde una estación base diferente.

GEE. Sistema de radionavegación similar a LORAN y diseñado en paralelo a este. En GEE las cadenas están formadas por una estación maestra y tres esclavas. El receptor se encarga de medir las diferencias de tiempos entre los pares maestra-esclava, y a partir de ahí se puede determinar la posición del blanco mediante la identificación de las hipérboles correspondientes.

Omega. Es un sistema de navegación similar a DECCA, con la particularidad de que en Omega las diferencias de tiempos se pueden medir entre pares cualesquiera de las estaciones que forman la cadena: ya no existe la nomenclatura de maestra y esclava. Constaba de 8 estaciones distribuidas por toda la superficie terrestre, con la voluntad de ofrecer un sistema global de navegación, operativo a cualquier hora en cualquier lugar del planeta.

2.1.3 Sistemas de aproximación y aterrizaje

Este tipo de sistemas de radionavegación terrestre no se analizan en perspectiva de obtener la distancia u orientación de una aeronave, como los radiofaros o los sistemas de radionavegación hiperbólicos, sino que proporcionan herramientas a los pilotos y al personal del aeropuerto para facilitar y controlar las maniobras de aproximación a tierra y el posterior aterrizaje. A continuación presentaremos estos sistemas con sus características básicas más importantes:

Sistemas de control aéreo. Los aeropuertos están equipados con un sistema de radares para monitorizar los blancos, aeronaves, que sobrevuelan el espacio cercano. Las aeronaves son detectadas por el sistema y aparecen en la pantalla del controlador aéreo indicando su posición. Según el principio de funcionamiento que rige los radares, podemos diferenciar entre radares primarios y secundarios.

En los radares primarios el blanco es utilizado a modo de espejo, con la función de hacer rebotar la señal a tierra para así calcular su distancia, y de ahí su posición en el espacio y en la pantalla del controlador. En los secundarios el blanco ya no es un actor pasivo, sino que procesa la señal emitida desde el sistema radar emisor ubicado en tierra y responde con una señal de vuelta, que originariamente procedía de la necesidad militar de identificar a las aeronaves como aliadas o enemigas.

Sistemas de ayuda al aterrizaje. Existen dos sistemas de ayuda al aterrizaje, el sistema ILS y el sistema MLS.

El sistema ILS está concebido para guiar a los pilotos de aeronaves hacia la pista de aterrizaje. Está formado por tres partes: un localizador, situado al final de la pista de aterrizaje que proporciona la azimut² a la aeronave; el *glide slope*, situado a pie de pista que proporciona información sobre la elevación de la aeronave; y diferentes marcadores, que se encargan de ir ofreciendo a la aeronave información sobre la distancia hasta la pista. Estos tres elementos facilitan la maniobra de aterrizaje y permiten al piloto saber en todo momento dónde se encuentra su aeronave teniendo como referencia la pista de aterrizaje.

El sistema MLS opera en la banda de las microondas y el diseño corresponde al proceso *time reference scanning-beam* (TRSB). La TRSB propone el uso de dos antenas: una azimutal, que proporciona la azimut al piloto de la aeronave; y una de elevación que, como en el caso del *glide slope* de ILS, se sitúa a pie de pista y proporciona información sobre la elevación de la aeronave.

Actualmente, tanto ILS como MLS coexisten en la mayoría de los aeropuertos. Si bien es cierto que MLS proporciona resultados más precisos, el alto coste de equipamiento requerido hace que todavía no se haya abandonado ILS.

² Azimut es el ángulo, longitud de arco, formado entre la dirección de referencia Norte y la proyección vertical de la aeronave sobre el plano que definen la dirección de referencia y la proyección.

2.2 Sistemas de radionavegación por satélite

Los sistemas de radionavegación por satélite (GNSS) utilizan una constelación de satélites a modo de estaciones emisoras para enviar señales de radionavegación de forma continua. A partir de estas señales y de la posición del satélite, el receptor ubicado en la superficie terrestre puede estimar la distancia que le separa del satélite para más tarde calcular su posición.

A diferencia de gran parte de los sistemas de radionavegación terrestre, la mayoría de los basados en satélites proporcionan una cobertura global a lo largo de todo el planeta. Dentro de los GNSS se está produciendo una carrera por copar un mercado todavía al alza y actualmente existen diversos sistemas en liza. El más conocido de ellos, GPS, es el sistema americano y será objeto de estudio en el presente proyecto. Existen otros como el ruso GLONASS, el chino COMPASS o el estándar europeo Galileo.

De todos ellos, únicamente GPS y GLONASS tienen a día de hoy cobertura global sobre el planeta, mientras que COMPASS y Galileo ofrecen todavía cobertura local, con la intención de cubrir toda la Tierra lanzando en órbita más satélites en los próximos años. Es por ello que, aunque no tengan cobertura global, se les engloba dentro de los GNSS.

De forma paralela a los GNSS, existen los RNSS (*Regional Navigation Satellite Systems*), con cobertura local sobre determinadas zonas de la Tierra. Así, existe el estándar japonés QZSS y el indio IRNSS.

Todos los sistemas y estándares anteriormente mencionados tienen una arquitectura similar basada en tres segmentos: segmento espacial, de control, y de usuario. El segmento espacial está constituido por los satélites que orbitan alrededor de la Tierra, encargados de transmitir las señales de radionavegación. El segmento de control se encarga de la monitorización y el seguimiento de los satélites del segmento espacial. El segmento de usuario está formado por el conjunto de receptores ubicados sobre la superficie terrestre. La arquitectura de los GNSS será presentada de forma detallada tomando como ejemplo el estándar americano en el siguiente capítulo.

Los sistemas de radionavegación por satélite tienen múltiples aplicaciones:

Localización. Para determinar la posición de un receptor.

Navegación. Permite llegar de una ubicación a otra.

Seguimiento. Monitorizar el movimiento de personas u objetos.

Mapeado. Crear mapas de las diferentes zonas de la Tierra.

Sincronización. Ofrecer sincronización entre dispositivos.

2.3 Clasificación de los sistemas de radionavegación

A modo de resumen, presentamos una tabla (Tabla 2.1) para poner sobre la mesa todos los sistemas anteriores. Como podemos observar, dada la propia arquitectura de los sistemas, la radionavegación terrestre es más proclive a tener una cobertura local, mientras que la radionavegación por satélite tiende a tener una cobertura global sobre toda la superficie de la Tierra.

Radionavegación	Tipo	Ejemplo	Cobertura
Terrestre	Radiofaros	NDB	Local
		VOR	Local
		DME	Local
		TACAN	Local
		CONSOL	Local
	Hiperbólicos	DECCA	Local
		LORAN	Potencialmente Global
		GEE	Potencialmente Global
		Omega	Potencialmente Global
	Aproximación y aterrizaje	Control Aéreo	Local
		Ayuda al aterrizaje	Local
Por satélite	GNSS	GPS	Global
		GLONASS	Global
		COMPASS	Potencialmente Global
		Galileo	Potencialmente Global
	RNSS	QZSS	Local
		IRNSS	Local

Tabla 2.1. Clasificación de los sistemas de radionavegación

La tabla anterior (Tabla 2.1) contextualiza el Sistema GPS dentro del total de los sistemas de radionavegación para dar por concluido el segundo capítulo. El tercer capítulo se adentra por completo en el Sistema GPS, analizando su arquitectura, y la manera de posicionar a un usuario mediante el cálculo de la distancia que le separa de los satélites. Además, introduce las técnicas para situar a los satélites en el espacio y a los usuarios en la superficie terrestre.

3. Fundamentos del Sistema GPS

El Sistema GPS fue inicialmente concebido como una herramienta de estrategia bélica por el Departamento de Defensa de Estados Unidos en la década de los 70. Años más tarde, y analizando el gran impacto que podría tener sobre la vida cotidiana, los altos dirigentes norteamericanos divulgaron el invento entre el público civil.

Pero vayamos por partes. Todo comenzó con el lanzamiento del primer satélite artificial, Sputnik I, por parte de la Unión Soviética en 1957. Aprovechándose de este hecho, investigadores de la universidad norteamericana Johns Hopkins intentaron determinar su posición en el espacio mediante el uso de varios receptores fijos sobre la superficie terrestre de ubicación conocida. Después de meses de estudio, consiguieron dar con las coordenadas del satélite. Entonces, se plantearon la problemática inversa, ¿sería posible determinar la posición de un objeto en la Tierra mediante satélites de ubicación conocida utilizando las señales que emitían los propios satélites?

Obviamente ahora conocemos la respuesta, pero en la década de los 60 las actualizaciones de posición estaban disponibles cada 40 minutos, y el receptor debía permanecer estático para poder obtener la información adecuada de su posición. Aunque había muchísimo margen de mejora, los investigadores norteamericanos dieron con la solución al posicionamiento terrestre mediante señales satelitales, y de esta manera nació el sistema TRANSIT, que quedó operativo en 1964 y tres años más tarde estuvo disponible para uso comercial.

TRANSIT se convirtió en el precursor del Sistema GPS, y tenía la función de ayudar a la flota estadounidense en tareas de posicionamiento: obstáculos, enemigos, ubicación de zonas aliadas, etc. El sistema constaba de 6 satélites que recorrían órbitas polares³ muy bajas, a una altura de 1074 kilómetros. Su funcionamiento se basaba en la emisión de dos señales y eran los receptores los que determinaban su propia ubicación midiendo la desviación de la frecuencia Doppler de las señales recibidas respecto de unas tablas almacenadas en su memoria.

En plena Guerra Fría, la Unión Soviética tardó en reaccionar y no fue hasta 1974 cuando lanzó el primer satélite de su nuevo sistema de radionavegación, bautizado como TSICADA. La carrera del posicionamiento global estaba en marcha pero los norteamericanos llevaban una década de ventaja. Así, estos últimos dieron un golpe de efecto porque cuando la Unión Soviética recién empezaba a dar sus primeros pasos con TSICADA, la Armada y la Fuerza Aérea de los Estados Unidos combinaron sus programas para modificar las señales satelitales mediante un código pseudo-aleatorio PRN. Este fue el punto de partida del programa NAVSTAR GPS.

Entre 1978 y 1985 se lanzaron 11 satélites experimentales NAVSTAR, a los que siguieron otras generaciones de satélites hasta completar la constelación actual. El Sistema GPS se declaró con capacidad operacional inicial en diciembre de 1993, y con capacidad operacional total en abril de 1995.

³ La órbita polar es un tipo de órbita que pasa por encima de los polos de un planeta o muy cerca de ellos, es decir, la inclinación de la órbita es cercana a los 90°.

3.1 Mecánica orbital

Antes de meternos de lleno en aspectos específicos del Sistema GPS necesitamos entender el funcionamiento de las órbitas de los satélites y las leyes físicas que las regulan, analizaremos los parámetros de referencia de cualquier sistema de radionavegación por satélite, y definiremos un sistema de coordenadas que nos permita referenciar las medidas obtenidas a partir de las señales radio.

Por tanto, durante los dos primeros apartados de esta sección estudiaremos qué sistemas se utilizan para situar un móvil en un punto de la Tierra, y durante los dos últimos presentaremos las técnicas que nos permitirán saber en qué punto del espacio se encuentra un satélite determinado.

3.1.1 Elipsoide de referencia

La Tierra no dispone de una forma geométrica estándar: tiene un perfil irregular. Teniendo claro que no conseguiremos una representación perfecta, necesitamos construir un modelo para representarla ya que la posición hace referencia a un punto sobre la superficie del planeta. Para los sistemas de coordenadas que comentaremos más tarde, el modelo que se utiliza es el elipsoide de referencia.

El elipsoide de referencia consiste en un elipsoide de revolución que se obtiene a partir de la rotación de una elipse sobre su eje menor, siendo este eje paralelo al eje de rotación de la Tierra (Fig. 3.1).

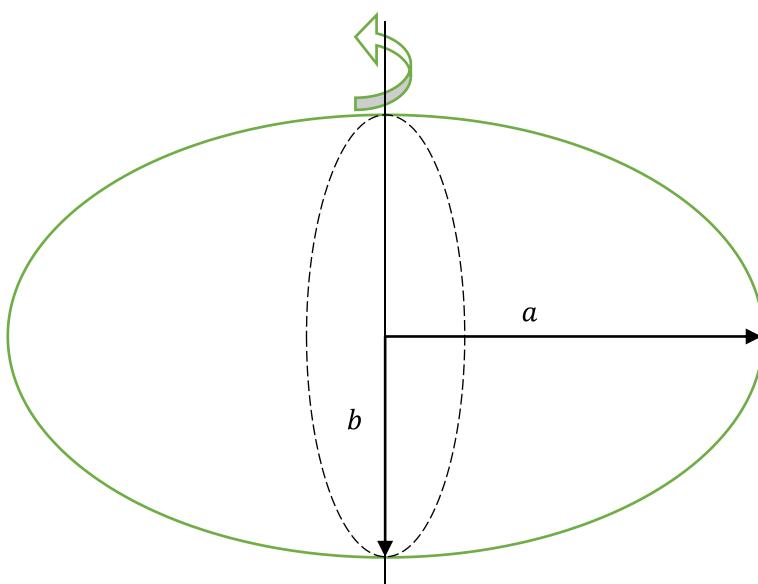


Fig. 3.1. Elipsoide de referencia

El elipsoide de referencia está definido por el semieje mayor a y el semieje menor b . Ambos ejes definen el factor de aplanamiento del elipsoide f de acuerdo con la expresión:

$$f = \frac{a - b}{a}$$

3.1.2 Sistemas de coordenadas

Una vez definido el elipsoide de referencia, y por tanto modelada la Tierra, debemos representar la posición obtenida a partir de las señales radio mediante un sistema de coordenadas. Aunque el presente proyecto no determinará la posición de objetos en la superficie terrestre, comentaremos brevemente los sistemas de coordenadas más habituales.

3.1.2.1 Coordenadas geográficas

Las coordenadas geográficas se basan en el uso del elipsoide de referencia, dividido en paralelos y meridianos, para modelar la superficie terrestre. Los paralelos son las elipses resultantes de ir cortando el elipsoide con planos paralelos al plano del ecuador. En cambio, los meridianos son las elipses resultantes de realizar los cortes de forma paralela al plano ortogonal del plano del ecuador que pasa por los polos.

Una vez dividido el elipsoide de referencia en paralelos y meridianos se debe escoger un meridiano de referencia y, normalmente, se usa el meridiano de Greenwich. En este momento ya podemos situar el blanco usando tres coordenadas: latitud, longitud y altura elipsoidal.

La **longitud λ** es el ángulo que se forma entre un meridiano de referencia y el meridiano donde se encuentra el punto.

La **latitud ϕ** se puede calcular de dos maneras atendiendo a dos referencias diferentes. Si la referenciamos al centro de la Tierra, la latitud ϕ^C es el ángulo existente entre la dirección del radio en el punto a posicionar p y el plano del ecuador. En cambio, si la referenciamos a la normal del plano tangente a la Tierra definido por p , la latitud ϕ^D se mide como el ángulo existente entre la normal al plano tangente al elipsoide en el punto a posicionar p y el plano del ecuador.

La **altura elipsoidal h** es la distancia entre la superficie del elipsoide y el punto a posicionar. Si el blanco está posicionado sobre la superficie del elipsoide de referencia, $h = 0$.

Podemos observar el sistema de coordenadas geográficas en la siguiente figura (Fig. 3.2):

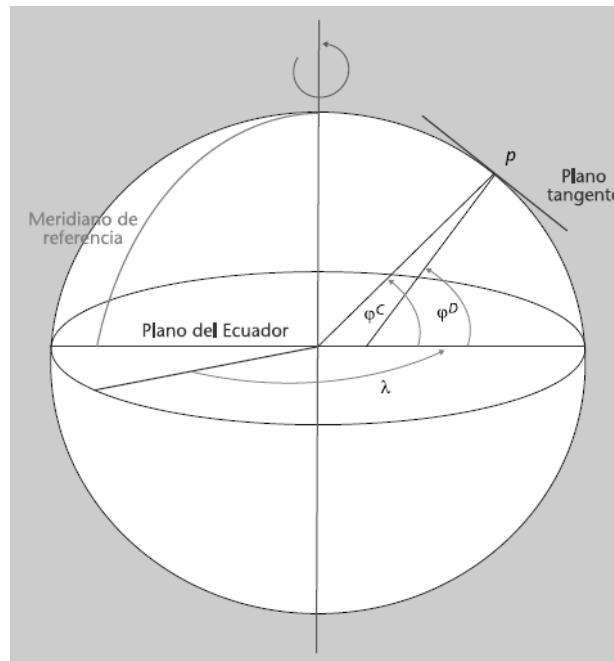


Fig. 3.2. Sistema de coordenadas geográficas

3.1.2.2 Coordenadas cartesianas

El sistema de coordenadas cartesianas (Fig. 3.3) utiliza tres coordenadas (x, y, z) para posicionar el blanco. Normalmente z se hace coincidir con el eje de rotación de la Tierra, x sigue el sentido del meridiano de referencia e y se forma de manera perpendicular a los dos ejes anteriores siguiendo el producto vectorial $x \wedge y = z$.

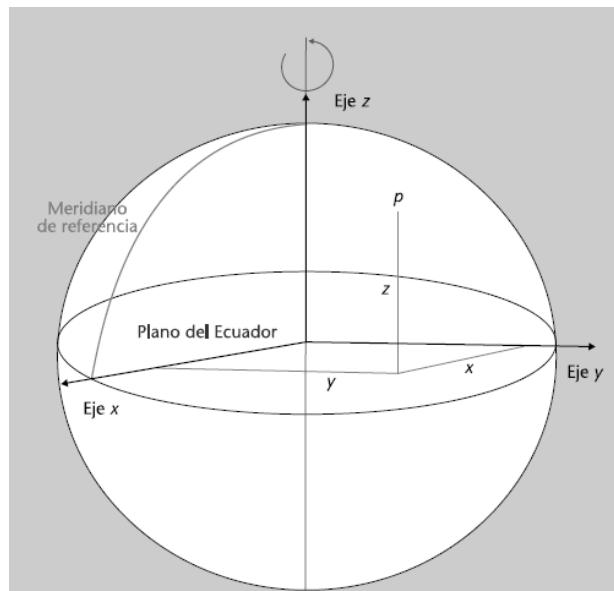


Fig. 3.3. Sistema de coordenadas cartesianas

3.1.3 Leyes de Kepler

Una vez introducida la manera de posicionar blancos terrestres, vamos a estudiar cómo saber en qué punto de su órbita se encuentra un satélite. Las leyes que rigen el movimiento de los satélites alrededor de la Tierra son las Leyes de Kepler, enunciadas por Johannes Kepler para describir matemáticamente las órbitas de los planetas alrededor del Sol. Si bien es cierto que estas leyes fueron formuladas para describir el movimiento de los planetas alrededor del Sol, son igualmente válidas para describir el movimiento de los satélites artificiales alrededor de la Tierra, ya que se considera que los satélites están únicamente afectados por la fuerza gravitacional de la Tierra, y su masa es considerada como masa puntual.

Primera Ley. La órbita de cada planeta es una elipse y el Sol se encuentra en uno de los focos de la elipse. Trasladando la ley a nuestro sistema, podemos decir que la órbita de cada satélite es una elipse con el centro de gravedad de la Tierra como uno de los focos.

Segunda Ley. El vector que une un planeta y el Sol barre áreas iguales en tiempo iguales. Esta ley tiene una implicación práctica importante: a medida que el satélite se encuentra más próximo a la Tierra, se mueve más rápido.

Tercera Ley. Para cualquier planeta o satélite, el cuadrado de su periodo orbital⁴ es directamente proporcional al cubo del semieje mayor de su órbita elíptica.

Además, para determinar la órbita que presenta cada uno de los satélites y su posición dentro de su órbita, debemos hacer uso de la Ley de Gravitación Universal, en la que se establece que dos cuerpos de masa M y m separados por una distancia r se atraen con fuerzas iguales y opuestas, el módulo de las cuales es:

$$F = G \frac{M \cdot m}{r^2}$$

Siendo $G = 6.67 \cdot 10^{-11} \frac{\text{m}^3}{\text{kg} \cdot \text{s}^2}$ la constante de gravitación universal.

⁴ Tiempo que tarda un planeta en dar una vuelta alrededor del Sol, o un satélite alrededor de la Tierra.

3.1.4 Parámetros orbitales

Cuando hablábamos del elipsoide de referencia lo utilizábamos para representar la posición de un blanco terrestre, obtenida a partir de las señales radio mediante un sistema de coordenadas. Es decir, a partir del elipsoide de referencia posicionábamos un blanco terrestre con un sistema de coordenadas.

Con las Leyes de Kepler y la Ley de Gravitación Universal somos capaces de determinar la órbita de un satélite y su posición dentro de ella. Ahora lo que necesitamos representar es la posición del satélite en el espacio en un instante t dado, obtenida a partir de las leyes anteriormente comentadas. Para ello utilizamos unas coordenadas especiales, llamadas parámetros orbitales:

Semieje mayor de la órbita a . Determina el tamaño de la órbita del satélite de la misma manera que el elipsoide de referencia modelaba la Tierra.

Semieje menor de la órbita b . Determina el tamaño de la órbita del satélite de la misma manera que el elipsoide de referencia modelaba la Tierra.

Excentricidad de la órbita e . Determina la forma de la órbita y viene dada por la expresión:

$$e = \frac{\sqrt{a^2 - b^2}}{a}$$

Notar que si la órbita del satélite es circular la excentricidad es nula.

Inclinación del plano de órbita i . Ángulo entre el plano de la órbita y el del ecuador.

Longitud del nodo ascendente Ω . Ángulo, medido en el plano del ecuador, existente entre el meridiano de Greenwich y el punto donde la órbita corta con el plano del ecuador.

Argumento del perigeo⁵ ω . Ángulo, medido en el plano de la órbita del satélite, entre el vector calculado al pasar por el perigeo y la dirección del nodo ascendente.

Instante de paso por el perigeo t_o . Instante de tiempo en que el satélite pasa por el perigeo.

Conociendo el valor de los parámetros $(a, e, i, \Omega, \omega, t_o)$ seremos capaces de determinar la posición en la que se encuentra el satélite dentro de su órbita en cada instante t .

⁵ El perigeo es el punto de la órbita elíptica del satélite más cercano al centro de la Tierra. Por el contrario, el apogeo es el punto más alejado.

La siguiente figura (Fig. 3.4) pretende clarificar los parámetros explicados anteriormente.

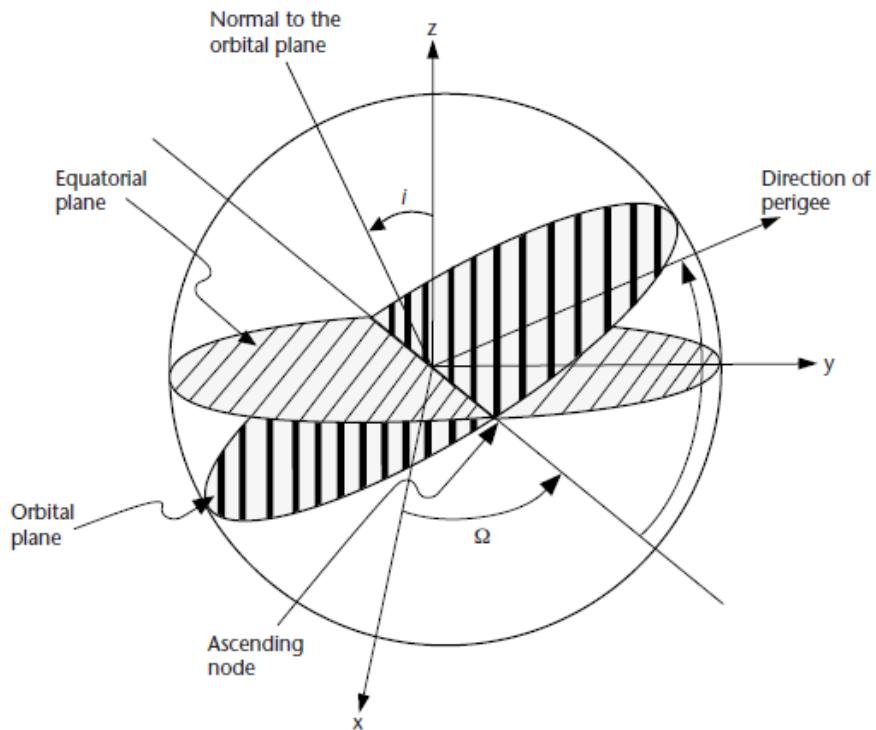


Fig. 3.4. Parámetros orbitales que permiten posicionar un satélite en el espacio

3.2 Arquitectura del Sistema GPS

El sistema de radionavegación por satélite GPS presenta una arquitectura dividida en tres segmentos claramente diferenciados (Fig. 3.5): el segmento espacial, formado por la constelación de satélites; el segmento de control, formado por las estaciones terrestres de seguimiento; y el segmento de usuario, formado por el conjunto de receptores.

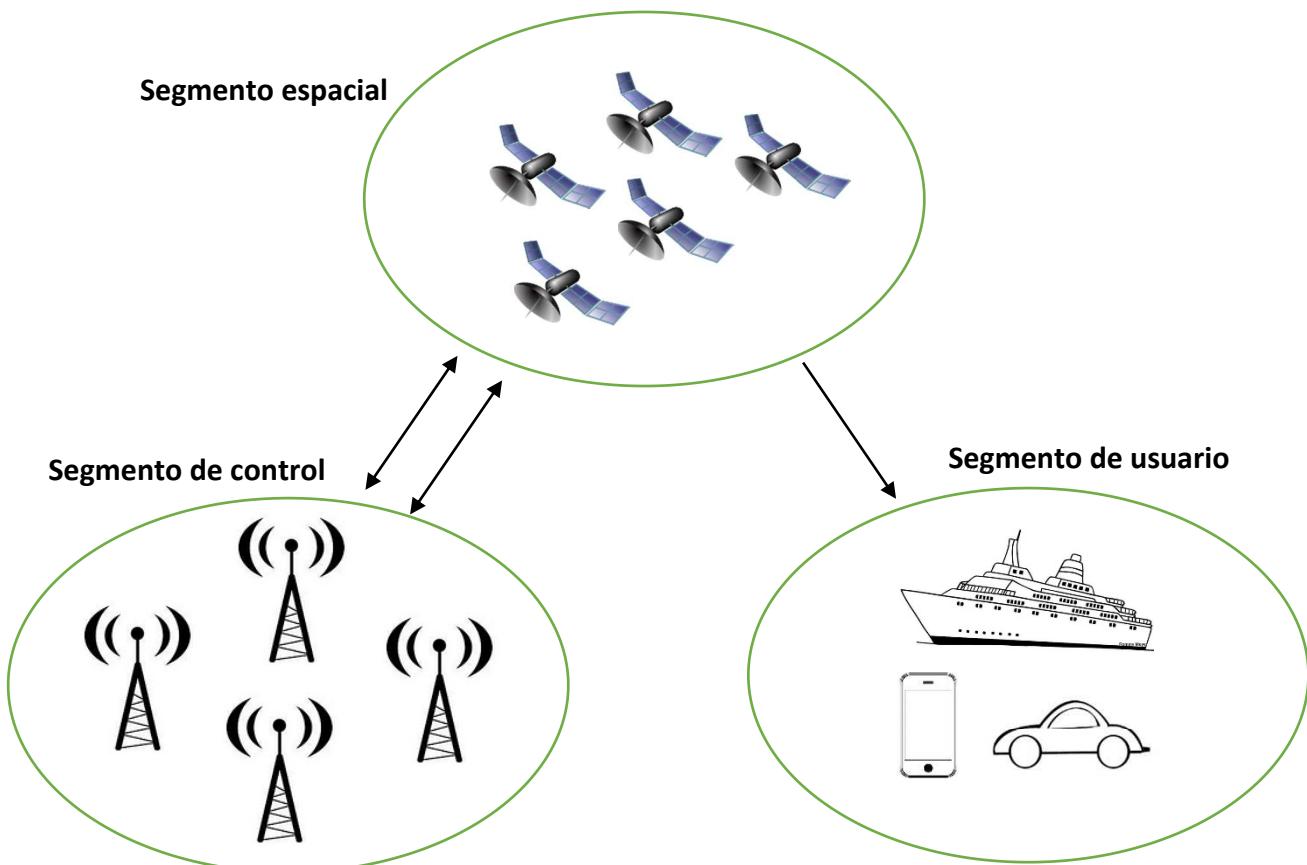


Fig. 3.5. Arquitectura del Sistema GPS

3.2.1 Segmento espacial

El segmento espacial está formado por la constelación de satélites GPS. Los satélites son simples repetidores, encargados de transmitir de forma continua el mensaje de navegación, que incluye sus efemérides⁶ y señales de tiempo, para que el receptor pueda estimar la distancia que le separa del satélite y posteriormente calcular su propia

⁶ Las efemérides son los parámetros orbitales del satélite, que permiten al receptor conocer la ubicación del satélite en el espacio.

posición. Como estudiaremos en la siguiente sección, son necesarios un mínimo de satélites visibles para que el receptor pueda calcular su posición.

Originariamente el Sistema GPS estaba compuesto por 24 satélites divididos en 6 planos orbitales (Fig. 3.6), 4 satélites en cada plano, aunque actualmente dispone de una constelación de 32. Siguen una órbita MEO⁷ prácticamente circular a una distancia aproximada de 26000 kilómetros sobre la superficie terrestre, lo cual permite que como mínimo 4 satélites sean visibles desde cualquier punto de la Tierra durante las 24 horas del día. Los satélites tienen un periodo orbital de 11 horas y 58 minutos.

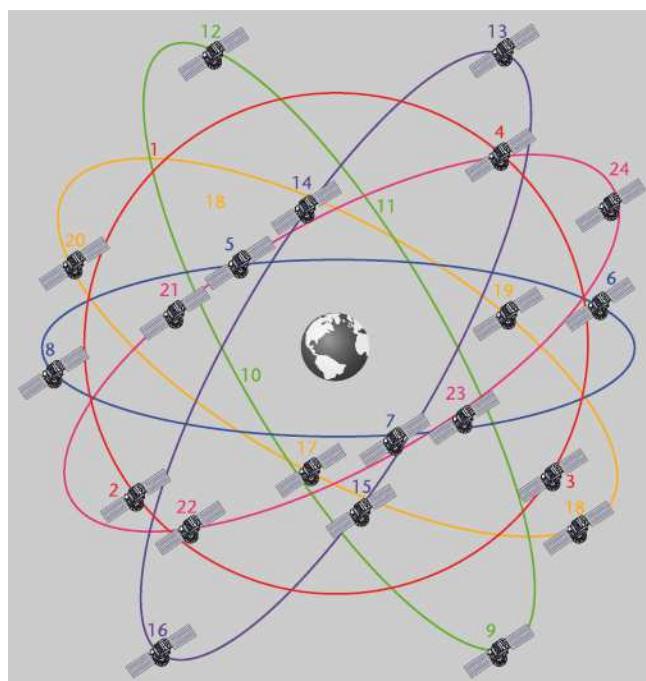


Fig. 3.6. Constelación base del Sistema GPS

Los parámetros orbitales de los satélites GPS son los siguientes:

$$a = 26000 \text{ km}$$

$$e = 0.02$$

$$i = 55^\circ$$

$$\Omega = \pm 2^\circ$$

$$\omega = \pm 180^\circ$$

⁷ MEO son las siglas en inglés de *Medium Earth Orbit*, y comprende distancias a la Tierra de entre 6000 km y 32000 km. También existen las órbitas LEO, *Low Earth Orbit*, y GEO, *Geosynchronous Earth Orbit*, entre otras.

Los satélites tienen diferentes funciones:

- Reciben y almacenan los datos del segmento de control.
- Mantienen una gestión del tiempo muy precisa, ya que disponen de relojes atómicos para evitar el problema de sincronismo con los demás satélites, y que este error de sincronismo solo tenga lugar con el reloj del receptor.
- Transmiten datos al segmento de usuario a través de diferentes frecuencias.
- Controlan su posición y altitud.
- Puede habilitarse un enlace inalámbrico entre satélites.

3.2.2 Segmento de control

El segmento de control es el encargado de monitorizar y realizar el seguimiento de los diferentes satélites del segmento espacial, actualizar sus efemérides y construir el mensaje de navegación que envía cada uno de ellos. Además, también actualiza los estados de los relojes atómicos de los satélites para sincronizar el sistema y resuelve las anomalías que presenten los satélites. El segmento de control está formado por estaciones de monitorización, una estación de control maestra y estaciones de transmisión de datos.

Las **estaciones de monitorización⁸ (MS)** están repartidas por toda la superficie terrestre y son las encargadas de seguir las señales de cada satélite calculando las distancias entre la estación y los diferentes satélites. Una estación de monitorización está formada por un receptor en banda L, dos relojes atómicos de cesio, sensores meteorológicos, servidores locales y un equipo de comunicaciones para establecer comunicación con la estación de control maestra. Las estaciones de monitorización se encuentran en emplazamientos cercanos al ecuador para maximizar su cobertura, tal y como podemos observar en la Fig. 3.7.

La **estación de control maestra⁹ (MCS)** es la encargada de centralizar toda la información que han recogido las estaciones de monitorización anteriores. Además, lleva a cabo todo el procesado de información para mantener a los satélites en sus órbitas, actualiza sus relojes y construye el mensaje de navegación que debe enviar cada uno de ellos. La estación de control maestra se encuentra bajo el control de las Fuerzas Aéreas de Estados Unidos y está localizada en la Base Aérea Schriever, en Colorado Springs (Estados Unidos).

Las **estaciones de transmisión de datos¹⁰ (DTS)** son las encargadas de transmitir a los satélites las órdenes de control y los mensajes de navegación creados por la MCS. Existen 4 estaciones localizadas en el mismo emplazamiento que el de algunas de las estaciones de monitorización: Isla de Ascensión, Diego García, Kwajalein y Cabo

⁸ Monitor Stations (MS) en inglés.

⁹ Master Control Station (MCS) en inglés.

¹⁰ Data Transmission Stations (DTS) en inglés.

Cañaveral (Fig. 3.7). Estas estaciones operan en la banda S, reservada para el enlace de control entre el segmento terrestre y los satélites.

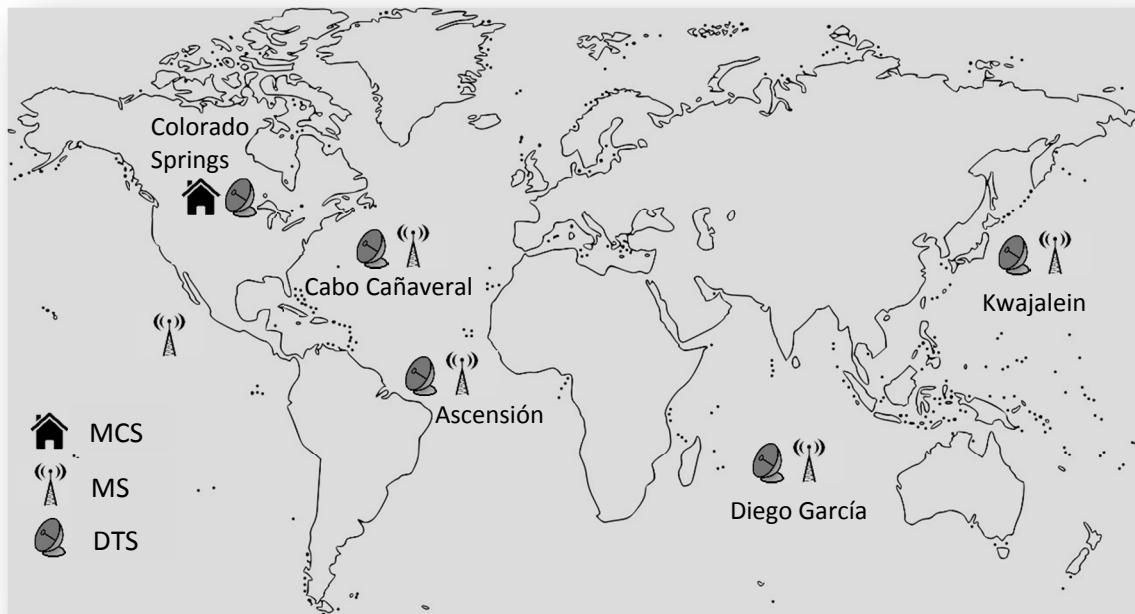


Fig. 3.7. Distribución geográfica del segmento de control del Sistema GPS

3.2.3 Segmento de usuario

El segmento de usuario está formado por el conjunto de receptores GPS. Pueden ser civiles o militares, y tienen tres bandas frecuenciales de trabajo: L1, L2 y L5, que estudiaremos en el siguiente capítulo. Como hemos comentado anteriormente, son los encargados de estimar la distancia que les separa de cada satélite para así calcular su propia posición. En el capítulo 5, Implementación del receptor GPS, estudiaremos en detalle las partes, funciones y características de un receptor GPS.

3.3 Cálculo de la posición

El concepto fundamental en que se basa el Sistema GPS para calcular la posición de un móvil es que se puede encontrar la posición de cualquier punto del espacio a partir de la distancia existente entre el punto y otros puntos de referencia de coordenadas conocidas. El punto del que se quiere encontrar su posición es el móvil, el receptor terrestre, y los puntos de referencia que se utilizan como anclaje para calcular las distancias son los satélites.

La siguiente figura (Fig. 3.8) representa el caso de la búsqueda bidimensional de la posición del móvil m a partir de la medida de la de las distancias d_1, d_2 y d_3 entre los satélites s_1, s_2 y s_3 y el móvil m .

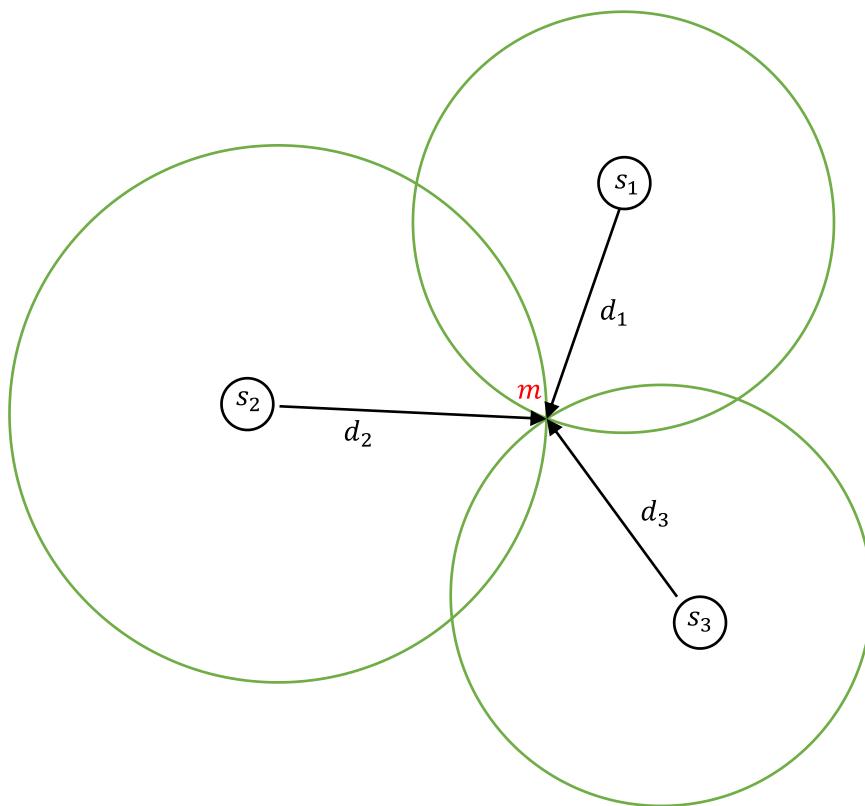


Fig. 3.8. Cálculo de la posición en 2D únicamente

Una vez conocidos los valores de distancia d_1, d_2 y d_3 , para un satélite dado de los tres anteriores el móvil puede estar situado en cualquiera de los puntos de la circunferencia que describe, ya que todos los puntos de una circunferencia distan la misma distancia del centro. De esta manera, la posición objetivo que buscamos m quedará determinada únicamente por la intersección de las tres circunferencias que describen la distancia al objetivo de cada satélite, tal y como podemos observar en la figura anterior.

En caso de disponer únicamente de información sobre la distancia de dos satélites, tendríamos una ambigüedad de posición, ya que no sabríamos en cuál de las dos posiciones de la Fig. 3.9 estaría el móvil m .

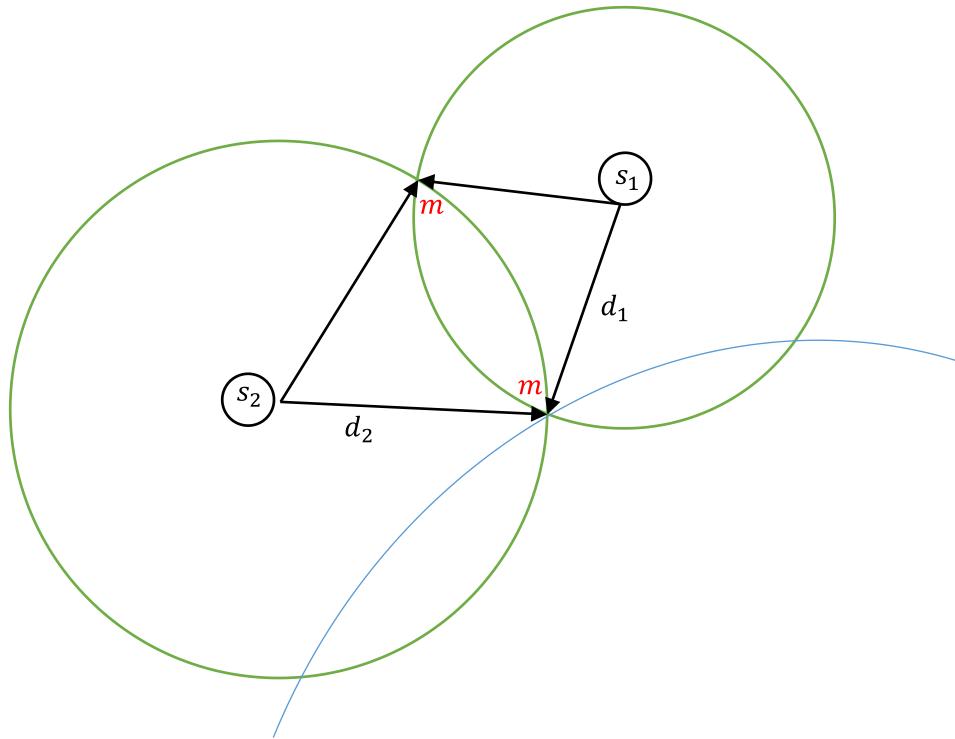


Fig. 3.9. Cálculo de la posición en 2D con ambigüedad

No obstante, uno de los objetivos del Sistema GPS es el cálculo de la posición de un móvil en un espacio tridimensional, en el que la forma geométrica que forman todos los puntos que se encuentran a una misma distancia del satélite ya no es una circunferencia como en el caso bidimensional, sino una esfera. Por lo tanto, si dispusiéramos de tres satélites habría dos puntos en el espacio que cumplirían con la restricción de intersección de las tres esferas (de igual manera que en la Fig. 3.9 ocurría con dos satélites en el caso bidimensional).

Sería lógico pensar que deberíamos añadir un cuarto satélite para definir de manera única la posición del móvil, pero volviendo al ejemplo ilustrativo bidimensional y teniendo en cuenta que la superficie de la Tierra está delimitada por el arco azul en la Fig. 3.9, nos damos cuenta de que el otro punto que cumpliría la restricción está a miles de kilómetros de la superficie terrestre, con lo que el receptor puede descartarlo directamente.

Por tanto, la lógica nos volvería a decir que tres satélites son suficientes para calcular la posición de un móvil en tres dimensiones. No obstante, debemos tener en cuenta un factor importante: la sincronización temporal entre los satélites y el receptor, que provocará tener que añadir un satélite más en el cálculo de la posición del móvil.

Los satélites siguen su órbita a lo largo del tiempo, por lo que la distancia entre satélite y móvil también varía. Es por ello que los satélites envían de manera continua una señal con información de sus efemérides y su reloj al receptor. A partir de esta señal, el receptor puede estimar su tiempo de propagación, y haciendo el producto con la velocidad de la luz en el vacío estima la distancia que le separa del satélite en cuestión.

$$\text{distancia} = c \cdot t_{\text{prop}}$$

Cabe resaltar que lo que el receptor calcula es una pseudodistancia¹¹. El prefijo *pseudo* se añade porque el valor no representa una verdadera distancia, ya que los instantes de transmisión y recepción de la señal GPS hacen referencia a escalas de tiempo diferentes: los relojes del satélite y del receptor no tienen por qué estar perfectamente sincronizados.

Este error de sincronismo provoca que tengamos que tener en cuenta la incertidumbre del tiempo en el cálculo de la posición, y es la razón por la que tenemos que añadir un cuarto satélite. En el siguiente esquema (Fig. 3.10) podemos observar el principio de funcionamiento del sistema de radionavegación GPS para posicionar un móvil.

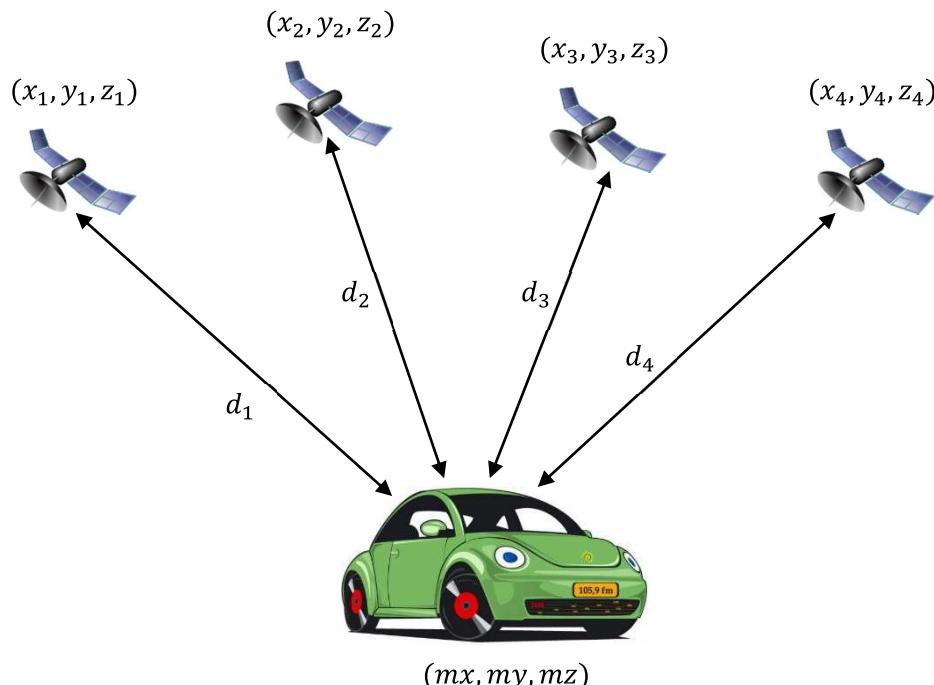


Fig. 3.10. Principio de funcionamiento del Sistema GPS para el cálculo de la posición

¹¹ Para la estimación de la distancia mediante el DLL hablaremos de pseudodistancia, pseudorango u observable de código, indistintamente. Para la estimación de la distancia mediante el PLL, hablaremos de observable de fase. En capítulos posteriores analizaremos ambas técnicas.

De esta manera, podemos plantear un sistema 4x4 de ecuaciones de navegación que nos permitirá corregir el error entre los relojes del satélite y el usuario, además de calcular las coordenadas del móvil:

$$d_1^2 = (x_1 - mx)^2 + (y_1 - my)^2 + (z_1 - mz)^2 = (d_{1medida} - c \cdot \text{error}_{reloj\,usuario})^2$$

$$d_2^2 = (x_2 - mx)^2 + (y_2 - my)^2 + (z_2 - mz)^2 = (d_{2medida} - c \cdot \text{error}_{reloj\,usuario})^2$$

$$d_3^2 = (x_3 - mx)^2 + (y_3 - my)^2 + (z_3 - mz)^2 = (d_{3medida} - c \cdot \text{error}_{reloj\,usuario})^2$$

$$d_4^2 = (x_4 - mx)^2 + (y_4 - my)^2 + (z_4 - mz)^2 = (d_{4medida} - c \cdot \text{error}_{reloj\,usuario})^2$$

De todas las variables existentes en las anteriores ecuaciones, las coordenadas de los satélites $(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), (x_4, y_4, z_4)$ son conocidas ya que el receptor recibe las efemérides en el mensaje de navegación, y es el propio receptor quien calcula los pseudorrangos $d_{1medida}, d_{2medida}, d_{3medida}, d_{4medida}$ que le separan de cada satélite. Por tanto, tenemos 4 ecuaciones con las 4 incógnitas: las coordenadas del receptor (mx, my, mz) y su error de reloj $\text{error}_{reloj\,usuario}$.

Es decir, si un receptor tiene visibilidad sobre 4 satélites es capaz de obtener las coordenadas de cada uno de ellos y medir la pseudodistancia que les separa a partir de cada una de las 4 señales GPS que recibe. Con estos datos, el receptor resuelve el sistema de ecuaciones para obtener su posición corrigiendo su error de reloj.

Con el cálculo de la posición de un móvil terrestre finalizamos el tercer capítulo, donde además hemos presentado la arquitectura del Sistema GPS, y la manera de situar en el espacio a un satélite y a un usuario sobre la superficie terrestre. En el siguiente capítulo analizaremos las componentes y la estructura de la señal GPS que envía el satélite, y que posteriormente recibirá el receptor.

4. La señal GPS

4.1 La señal GPS

Este capítulo presentará la estructura de la señal que transmite el satélite por el espacio: la señal GPS. Analizaremos cada componente de la señal GPS por separado, discutiremos sus características y parámetros principales, y veremos cómo todas ellas se engranan para formar la señal GPS que viaja por el espacio y es captada por el receptor terrestre.

La señal GPS está formada por los siguientes componentes:

Frecuencia portadora. Tiene la función de modular el mensaje de navegación y el código pseudoaleatorio. Típicamente se utilizan las frecuencias portadoras $f_{L1} = 1575.42$ MHz y $f_{L2} = 1227.60$ MHz.

Mensaje de navegación. Está compuesto por las efemérides y la información de reloj de los satélites.

Código pseudoaleatorio. Independiente para cada satélite, es su identificador. Típicamente se usa el código C/A y el código P.

Además, como toda señal que viaja por el espacio, está distorsionada por diferentes efectos. Nosotros trataremos con tres importantes, como son el ruido, la frecuencia Doppler y el *multipath*¹², además del filtrado. En las siguientes secciones explicaremos detalladamente cada uno de los componentes de la señal GPS.

4.2 Frecuencia portadora

El Sistema GPS utiliza tres bandas frecuenciales para transmitir la señal GPS y, dependiendo de qué aplicación quiera usarse, se transmitirá en una o en varias de ellas. Las tres están ubicadas en la banda L de la región UHF (*Ultra High Frequency*) del espectro electromagnético. Las tres bandas frecuenciales están referenciadas por la nomenclatura L1, L2 y L5 y derivan de una frecuencia común, utilizada por el oscilador

¹² En comunicaciones inalámbricas, *multipath* es el fenómeno que describe la llegada de la señal de interés a la antena receptora a través de dos o más caminos.

atómico del satélite $f_o = 10.23$ MHz. Concretamente, las bandas utilizadas en las que se encuentra cada frecuencia portadora son las siguientes:

L1. Banda centrada en la frecuencia portadora $f_{L1} = 154f_o = 1575.42$ MHz, y de una longitud de onda $\lambda \approx 19$ cm. Es utilizada para transmitir los códigos C/A, P y M¹³, de los que hablaremos en las siguientes secciones.

L2. Banda centrada en la frecuencia portadora $f_{L2} = 120f_o = 1227.60$ MHz, y de una longitud de onda $\lambda \approx 24$ cm. Es utilizada para transmitir los códigos P, M y L2C¹⁴.

L5. Banda centrada en la frecuencia portadora $f_{L5} = 115f_o = 1176.45$ MHz, y de una longitud de onda $\lambda \approx 25$ cm. Es utilizada para transmitir el código *Safety-of-Life Signal*¹⁵.

4.3 Mensaje de navegación

El mensaje de navegación proporciona al receptor terrestre información para calcular dos parámetros clave del Sistema GPS: permite determinar la posición del satélite visible y el tiempo de transmisión de la señal GPS. Está formado por la información de las efemérides y de reloj, entre otros parámetros. Además, también proporciona información auxiliar, como el estado de otros satélites, para facilitar una posible adquisición posterior. El mensaje de navegación se transmite utilizando una modulación BPSK a 50 bps. Por tanto, está formado por valores +1 y -1.

$$Tasa_{bit} = 50 \text{ bits/s}$$

De esta manera, se transmite 1 bit cada 20 ms:

$$T_{bit} = \frac{1}{Tasa_{bit}} = \frac{1}{50 \text{ bits/s}} = 20 \text{ ms}$$

¹³ El código M es una nueva señal militar transmitida en las portadoras L1 y L2 orientada a ofrecer un mayor nivel de seguridad.

¹⁴ El código L2C es una nueva señal de uso civil transmitida en la portadora L2 con el objetivo de mejorar las precisiones obtenidas por el código C/A.

¹⁵ *Safety-of-Life* es una señal transmitida en la portadora L5 orientada a ofrecer servicios de búsqueda y salvamento.

En la siguiente figura (Fig. 4.1) observamos la secuencia temporal de 10 bits de navegación. Como hemos comentado anteriormente, cada uno tiene una duración de 20 ms y su valor es binario: +1 o -1.

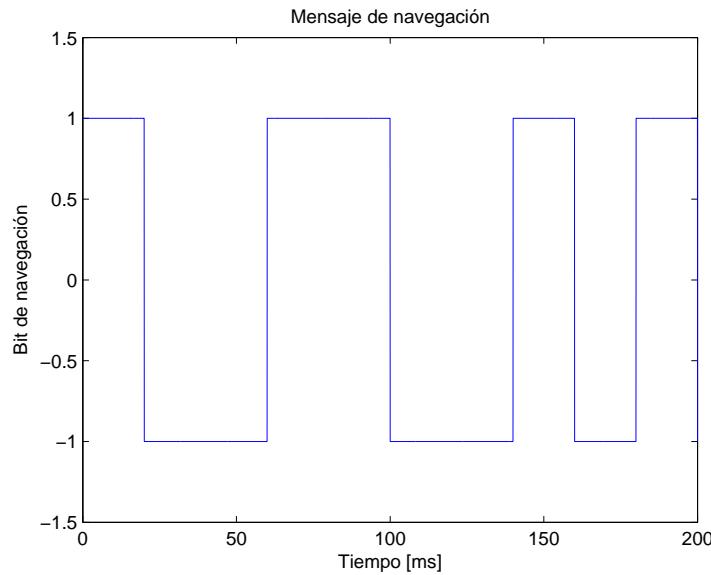


Fig. 4.1. Forma temporal del mensaje de navegación

De esta manera, los pasos por cero del espectro del mensaje de navegación se producen cada 50 Hz, tal y como podemos observar en la Fig. 4.2. Su espectro tiene forma de sinc^2 debido a que el mensaje de navegación está compuesto por pulsos rectangulares.

$$B_{\text{Mensaje Navegación}} = 50 \text{ Hz}$$

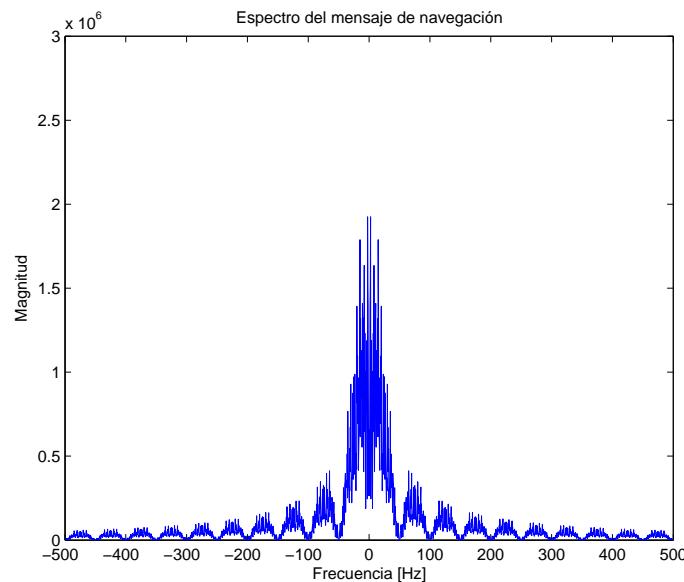


Fig. 4.2. Espectro del mensaje de navegación

En cuanto a su estructura, el mensaje de navegación se transmite en tramas, también llamadas páginas, siguiendo la estructura del siguiente esquema (Fig. 4.3):

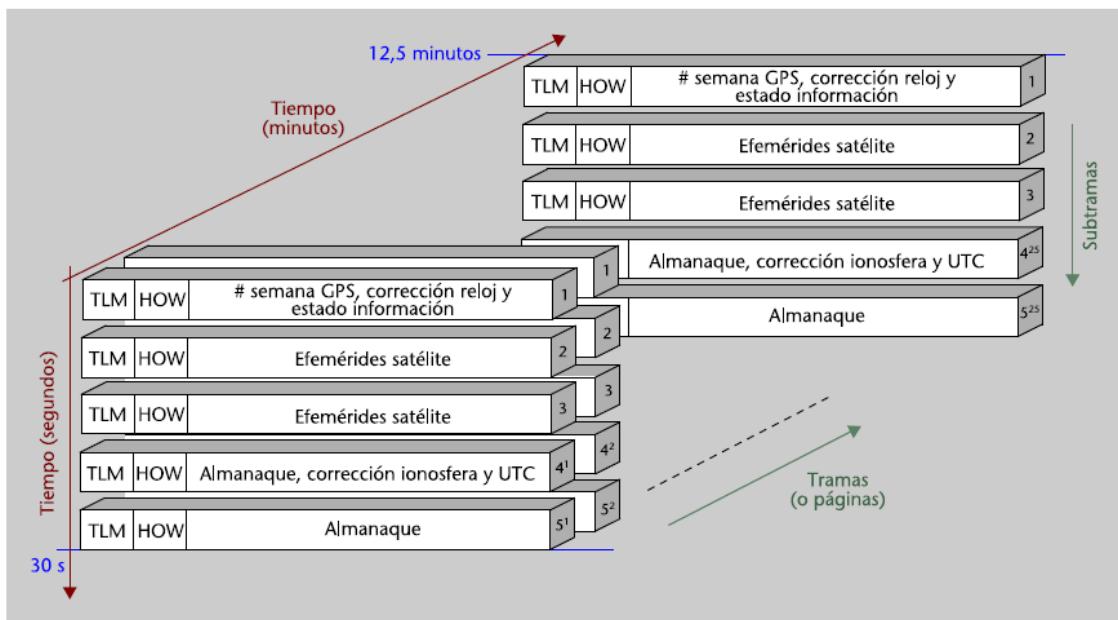


Fig. 4.3. Estructura general del mensaje de navegación

Tal y como observamos en el anterior esquema, para transmitir un mensaje de navegación completo se requieren 12.5 minutos, es decir, 750 segundos. Además, atendiendo a la duración de cada bit de navegación, se requieren 37500 bits:

$$\text{Total de bits Mensaje Navegación} = 750 \text{ s} \cdot \frac{1 \text{ bit}}{20 \text{ ms}} = 37500 \text{ bits}$$

Durante los 750 segundos se transmiten un total de 25 tramas, de duración 30 segundos y 1500 bits cada una. A su vez, cada trama está dividida en 5 subtramas, de 6 segundos y 300 bits cada una. Finalmente, cada subtrama está dividida en 10 palabras, cada una de ellas de 600 ms y 30 bits de duración (Tabla 4.1):

Tipo de dato	# en Mensaje de Navegación	Duración Unitaria	Bits Unitarios
Trama	25	30 s	1500
Subtrama	$25 \cdot 5 = 125$	6 s	300
Palabra	$125 \cdot 10 = 1250$	600 ms	30

Tabla. 4.1. Estructura de las tramas del mensaje de navegación

Como podemos observar en la Fig. 4.3, las dos primeras palabras de cada subtrama (los primeros 60 bits) son las palabras TLM y HOW. A modo introductorio, la palabra TLM (*TeLeMetry*) está formada por una secuencia de 8 bits fija llamada preámbulo. La secuencia, 10001011, sirve para que el receptor reconozca cuál es el inicio de la subtrama. Los 22 bits restantes se organizan de la siguiente manera: 14 bits reservados para usuarios autorizados, 2 bits vacíos y 6 bits de paridad. Por otro lado, la palabra HOW (*HandOver Word*) permite al receptor pasar de un código C/A a un código P. Está formada por una versión truncada de 17 bits del TOW¹⁶, seguido de 2 bits de *flags*, 3 bits para identificar la subtrama en sí, 6 bits de paridad y los dos últimos vacíos.

En cuanto a la organización del resto de las palabras de las subtramas, cada una de ellas transporta la siguiente información:

Subtrama 1. Contiene el número de semana GPS¹⁷, términos de corrección del reloj del satélite e información sobre el estado de la información. La información de esta subtrama se actualiza en cada nueva trama.

Subtramas 2 y 3. Contiene las efemérides del satélite, compuestas por los parámetros orbitales y las anomalías, para poder determinar su posición. La información de estas subtramas también se actualiza en cada trama.

Subtramas 4 y 5. Contienen información sobre el almanaque (efemérides con menor resolución) de todos los satélites para facilitar al receptor la posible adquisición de cada uno de ellos, parámetros de corrección ionosféricos y parámetros para relacionar el reloj del Sistema GPS con el Sistema de tiempo UTC (*Universal Coordinated Time*). La información contenida en estas dos subtramas se distribuye a lo largo de las 25 que componen el mensaje de navegación, de ahí que existan los superíndices en el esquema de la Fig. 4.3, haciendo patente las diferentes versiones de cada una de ellas.

¹⁶ TOW son las siglas en inglés de *Time Of Week* y es la palabra que indica en qué instante de la semana se ha iniciado la transmisión de la subtrama.

¹⁷ La semana GPS es el número de la semana módulo 1024 transcurrido desde el 5 de enero de 1980.

4.4 Código C/A

Cada satélite tiene dos códigos pseudoaleatorios asociados, diferentes también de los del resto de satélites. El primero de ellos es el código C/A (*coarse acquisition code*) y el segundo es el llamado código P (*encrypted precision code*). El código C/A es modulado en la banda frecuencial L1 y el código P en las bandas L1 y L2. En el receptor software diseñado solo trabajamos con el código C/A, por lo que a continuación presentaremos sus características básicas. Los códigos pseudoaleatorios también son llamados secuencias de ensanchamiento o secuencias PRN¹⁸.

El código C/A de un satélite k está formado a partir de una secuencia binaria p_{c_k} (+1 y -1) con una longitud de 1023 chips, que se repiten periódicamente cada $T_{CA} = 1$ ms:

$$C_k(t) = \sum_{i=-\infty}^{+\infty} p_{c_k}(t - iT_{CA})$$

En la jerga GPS los bits de los códigos pseudoaleatorios son denominados chips, debido a que no transportan información relevante. Por tanto, podemos decir que el código C/A tiene 1023 chips y la duración de un periodo completo de su código es de 1 ms. De esta manera, el periodo de chip resultante es de aproximadamente 1μs.

$$T_{CA} = 1\text{ms}$$

$$T_{chip} = \frac{1 \text{ chip}}{1023 \frac{\text{chips}}{\text{ms}}} = 0.977 \mu\text{s} \approx 1\mu\text{s}$$

En la siguiente figura (Fig. 4.4) observamos la secuencia temporal de los 10 primeros chips del código pseudoaleatorio 12. Como hemos comentado anteriormente, cada uno de ellos tiene una duración de ≈ 1μs y su valor es binario: +1 o -1.

¹⁸ PRN son las siglas en inglés de *pseudo-random noise*.

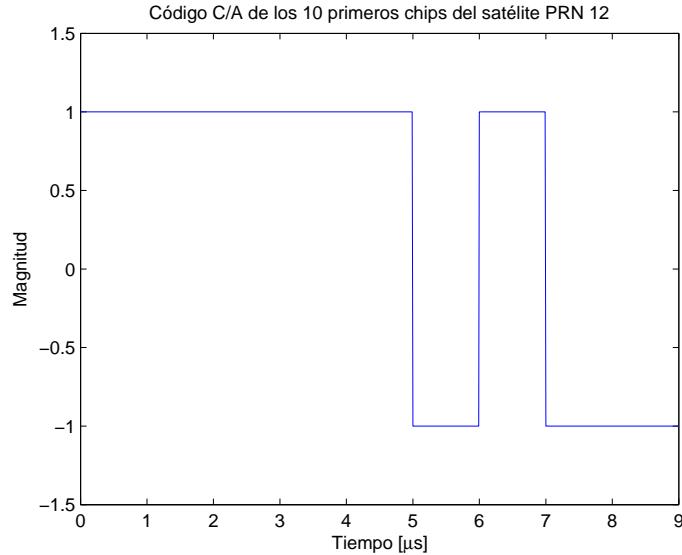


Fig. 4.4. Forma temporal de los 10 primeros chips del código PRN 12

Siguiendo con un razonamiento similar al ofrecido para el mensaje de navegación:

$$f_{chip} = \frac{1}{T_{chip}} = \frac{1}{0.977 \text{ } \mu\text{s}} = 1.023 \text{ MHz} = Tasa_{chip} = 1.023 \text{ Mchips/s}$$

Con lo que los pasos por cero del espectro del código C/A (Fig. 4.5) serán aproximadamente cada 1 MHz:

$$B_{CA} = 1.023 \text{ MHz} \approx 1 \text{ MHz}$$

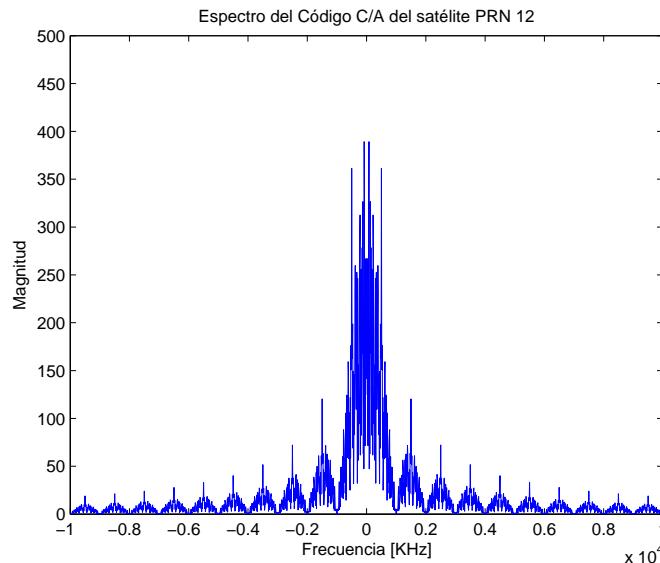


Fig. 4.5. Espectro del código C/A del satélite asociado al PRN 12

4.4.1 Generación del código C/A

El código C/A utilizado en GPS pertenece a una familia de secuencias llamadas códigos Gold. Los códigos Gold son secuencias deterministas con propiedades semejantes al ruido blanco, y se generan usando dos registros de desplazamiento de 10 bits. El código C/A se genera haciendo la suma en módulo 2 de la combinación de ciertos bits de ambos registros de desplazamiento, tal y como podemos observar en la Fig. 4.6.

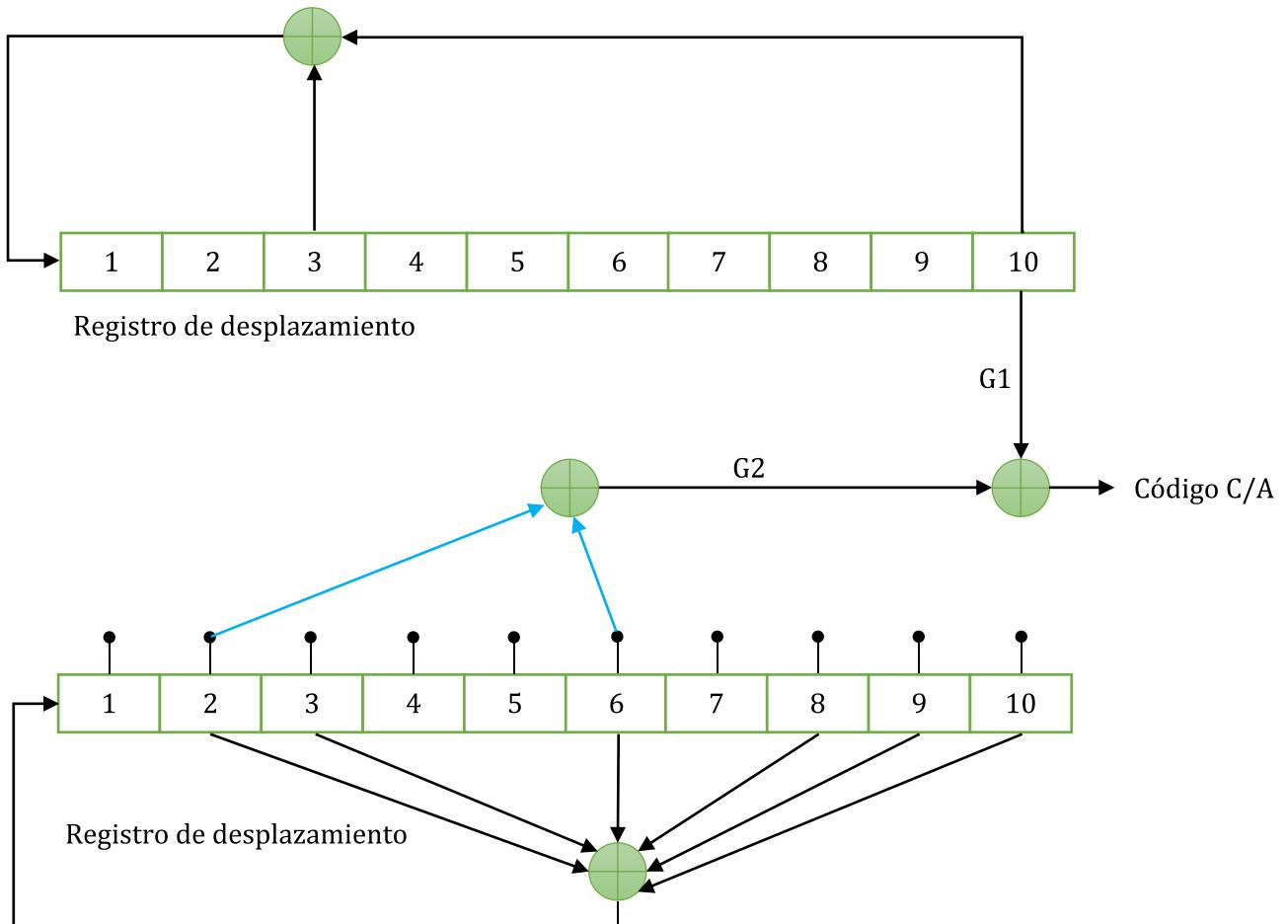


Fig. 4.6. Esquema de generación de un código C/A

La generación del código C/A se realiza bit a bit, como la suma en módulo 2 de G1 y G2 (Fig. 4.6). Para generar el código C/A inicializamos todas las posiciones de los dos registros de desplazamiento a 1. La última posición del primer registro ya es G1, mientras que para generar G2 debemos hacer la suma en módulo 2 de dos posiciones específicas del segundo registro. Cada par de posiciones de generación de G2 identifica a un satélite y son fijas para la generación de los 1023 chips del código C/A, por lo que es importante conocer cuáles son las adecuadas. A continuación las especificamos (Tabla 4.2):

Satélite	Par de posiciones G2	Satélite	Par de posiciones G2
1	2 ⊕ 6	17	1 ⊕ 4
2	3 ⊕ 7	18	2 ⊕ 5
3	4 ⊕ 8	19	3 ⊕ 6
4	5 ⊕ 9	20	4 ⊕ 7
5	1 ⊕ 9	21	5 ⊕ 8
6	2 ⊕ 10	22	6 ⊕ 9
7	1 ⊕ 8	23	1 ⊕ 3
8	2 ⊕ 9	24	4 ⊕ 6
9	3 ⊕ 10	25	5 ⊕ 7
10	2 ⊕ 3	26	6 ⊕ 8
11	3 ⊕ 4	27	7 ⊕ 9
12	5 ⊕ 6	28	8 ⊕ 10
13	6 ⊕ 7	29	1 ⊕ 6
14	7 ⊕ 8	30	2 ⊕ 7
15	8 ⊕ 9	31	3 ⊕ 8
16	9 ⊕ 10	32	4 ⊕ 9

Tabla 4.2. Identificación de pares de posiciones para generar G2

Una vez elegidas las posiciones del segundo registro de desplazamiento (observamos que en la Fig. 4.6 estamos generando el código C/A del primer satélite), únicamente restará hacer la suma en módulo 2 de las posiciones especificadas en la Fig. 4.6 para generar el primer bit de cada registro, y volveremos a repetir la operación 1022 veces para generar un código C/A completo.

4.4.2 Propiedades de correlación del código C/A

Enmarcados dentro de la familia de los códigos Gold, el código C/A presenta propiedades similares al ruido blanco en términos de correlación.

Propiedad 1. Todos los códigos C/A están prácticamente incorrelados con el resto: presentan niveles de correlación cruzada muy bajos. Para dos códigos C^i y C^k correspondientes a los satélites i y k , la correlación cruzada se puede expresar como:

$$r_{ik}(m) = \sum_{l=0}^{1022} C^i(l)C^k(l+m) \approx 0 \quad \text{para todo } m$$

Los valores de correlación no siguen ningún patrón de comportamiento entre códigos C/A diferentes, como podemos observar en el ejemplo de correlación cruzada entre los satélites 12 y 29 (Fig. 4.7):

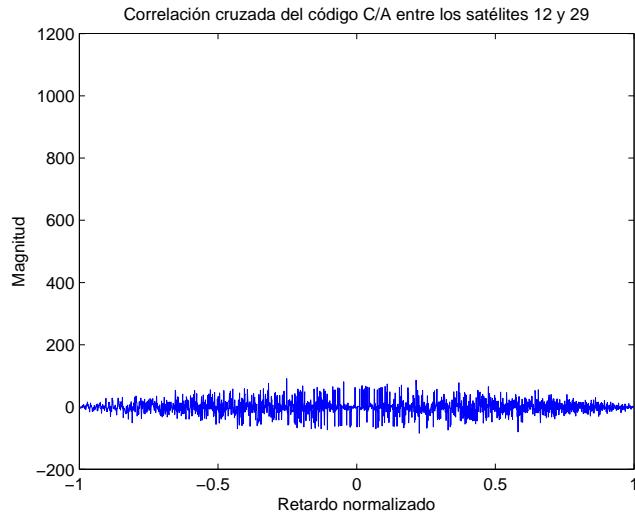


Fig. 4.7. Correlación cruzada del código C/A entre los satélites 12 y 29

Propiedad 2. Todos los códigos C/A están prácticamente incorrelados con ellos mismos excepto para un retardo nulo (Fig. 4.8). Es decir, solo cuando multiplicamos muestra a muestra exactamente la señal por ella misma se produce un pico de correlación. El sumatorio de los productos con una de las señales desplazadas produce un nivel de autocorrelación prácticamente nulo. La autocorrelación para un satélite k se puede expresar como:

$$r_{kk}(m) = \sum_{l=0}^{1022} C^k(l)C^k(l+m) \approx 0 \quad \text{para } |m| \geq 1$$

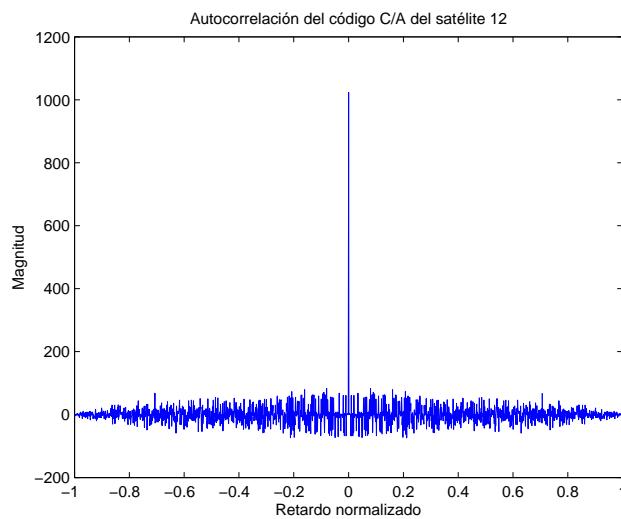


Fig. 4.8. Autocorrelación del código C/A del satélite 12

4.5 Estructura de la señal GPS

Una vez estudiadas las señales que intervienen en la generación de la señal GPS, analizaremos cómo se engrana cada una de ellas para formar el esquema de transmisión de la señal GPS (Fig. 4.9):

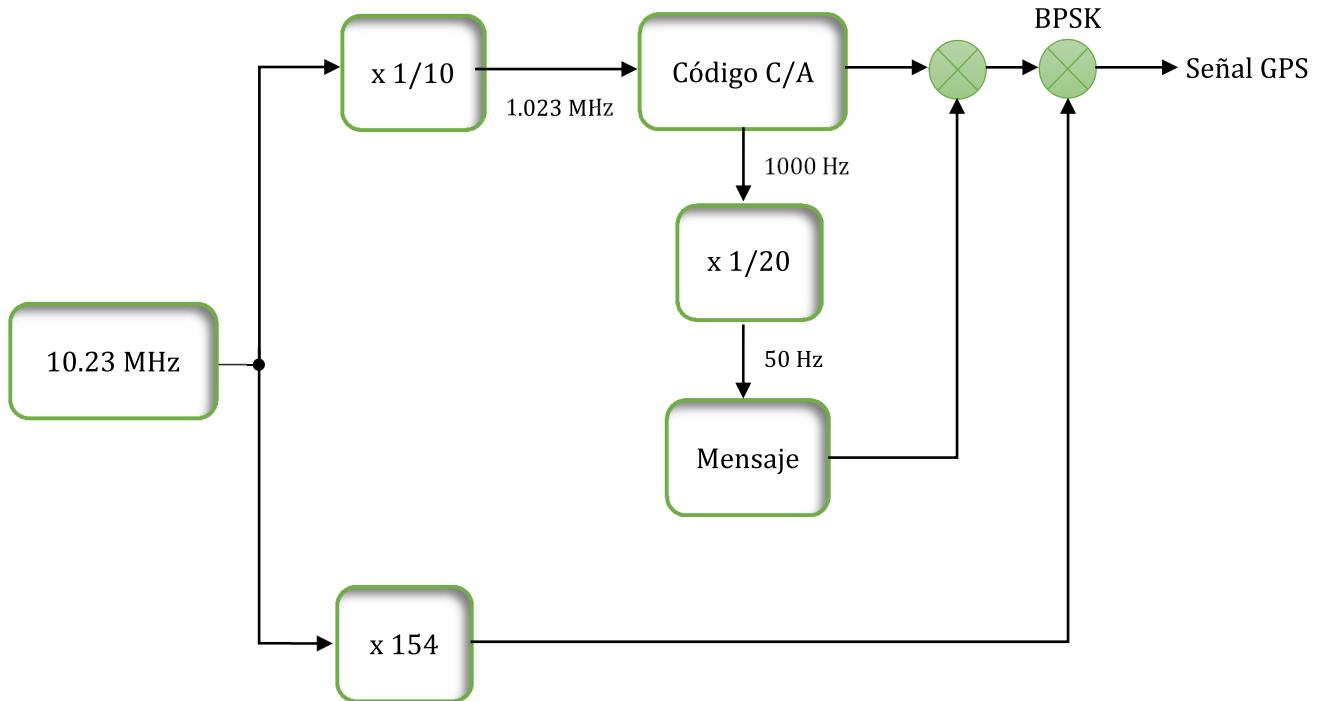


Fig. 4.9. Esquema de transmisión de la señal GPS

El anterior esquema (Fig. 4.9) solo tiene en cuenta la frecuencia portadora L1, pero en un caso más general donde también interviniere la frecuencia portadora L2, la señal GPS resultante que transmite el satélite k viene dada por la siguiente expresión:

$$s_k(t) = \sqrt{2P_c}C_k(t)D_k(t) \cos(2\pi f_{L1}t) + \sqrt{2P_{P_{L1}}}P_k(t)D_k(t) \sin(2\pi f_{L1}t) \\ + \sqrt{2P_{P_{L2}}}P_k(t)D_k(t) \sin(2\pi f_{L2}t)$$

donde P_c , $P_{P_{L1}}$ y $P_{P_{L2}}$ son las potencias de los códigos C/A en la banda L1, y P en la banda L1 y L2 respectivamente. $C_k(t)$ y $P_k(t)$ son las secuencias pseudoaleatorias asociadas al satélite k , mientras que $D_k(t)$ es el mensaje de navegación transmitido por dicho satélite.

En este proyecto solamente trabajaremos con el código C/A, presente únicamente en la banda L1, por lo que los términos afectados por el código P y la banda L2 se anulan. De esta manera, nuestra señal GPS se ve simplificada de la siguiente forma:

$$s_k(t) = \sqrt{2P_c}C_k(t)D_k(t) \cos(2\pi f_{L1}t)$$

Ajustando la potencia del código C/A a $P_c = \frac{1}{2}$, podemos trabajar con una expresión aún más simplificada de la señal GPS:

$$s_k(t) = C_k(t)D_k(t) \cos(2\pi f_{L1}t)$$

Como podemos observar en la Fig. 4.10, el código C/A $C(t)$, se repite periódicamente cada milisegundo. El bit de navegación representado por $D(t)$ se mantiene constante durante 20 milisegundos, momento en el que permuta al siguiente bit. De esta manera, podemos concluir que en cada bit de navegación existen 20 códigos C/A. Por tanto, nominalmente, un bit consta de 20460 chips.

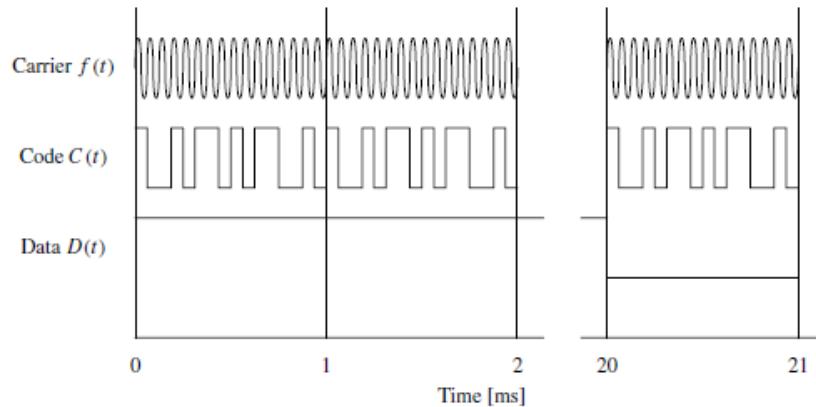


Fig. 4.10. Forma temporal comparativa de las señales que forman la señal GPS

4.6 Transmisión de la señal GPS

En anteriores secciones hemos analizado de manera detallada las componentes que forman la señal GPS que se envía a través del espacio, y que llega al receptor con una serie de efectos que ampliaremos en capítulos posteriores. El mensaje de navegación es la información que envía el satélite para que el receptor posteriormente pueda calcular su propia posición. Y usamos la portadora para modular esa información a una frecuencia superior, y así conseguir multiplexar la señal y optimizar los recursos en recepción. Pero, ¿por qué utilizamos un código pseudoaleatorio?

El código C/A se utiliza en la modulación DSSS (*Direct Sequence Spread Spectrum*) como señal de espectro ensanchado, y el uso de esta modulación permite utilizar la técnica de acceso al medio CDMA (*Code Division Multiple Access*). A continuación analizaremos qué ventajas tienen ambos sistemas y por qué el uso de un código pseudoaleatorio es vital para el planteamiento de los sistemas de radionavegación por satélite.

4.6.1 Modulación DSSS

La modulación DSSS se basa en el producto de dos señales. La primera de ellas es la señal original que transporta la información, en nuestro caso el mensaje de navegación, con un tiempo de bit mayor que provoca que el ancho de banda sea menor que el de la segunda señal. La segunda señal, llamada de espectro ensanchado y que en nuestro caso será el código C/A, dispone de un tiempo de bit más pequeño, lo que provoca que el espectro tenga un ancho de banda mayor que el de la primera señal. De esta manera, conseguimos que la señal resultante de este producto tenga un ancho de banda prácticamente igual a la señal de espectro ensanchado. El proceso de modulación se completa con el producto entre la señal resultante DSSS y la frecuencia portadora (Fig. 4.11).

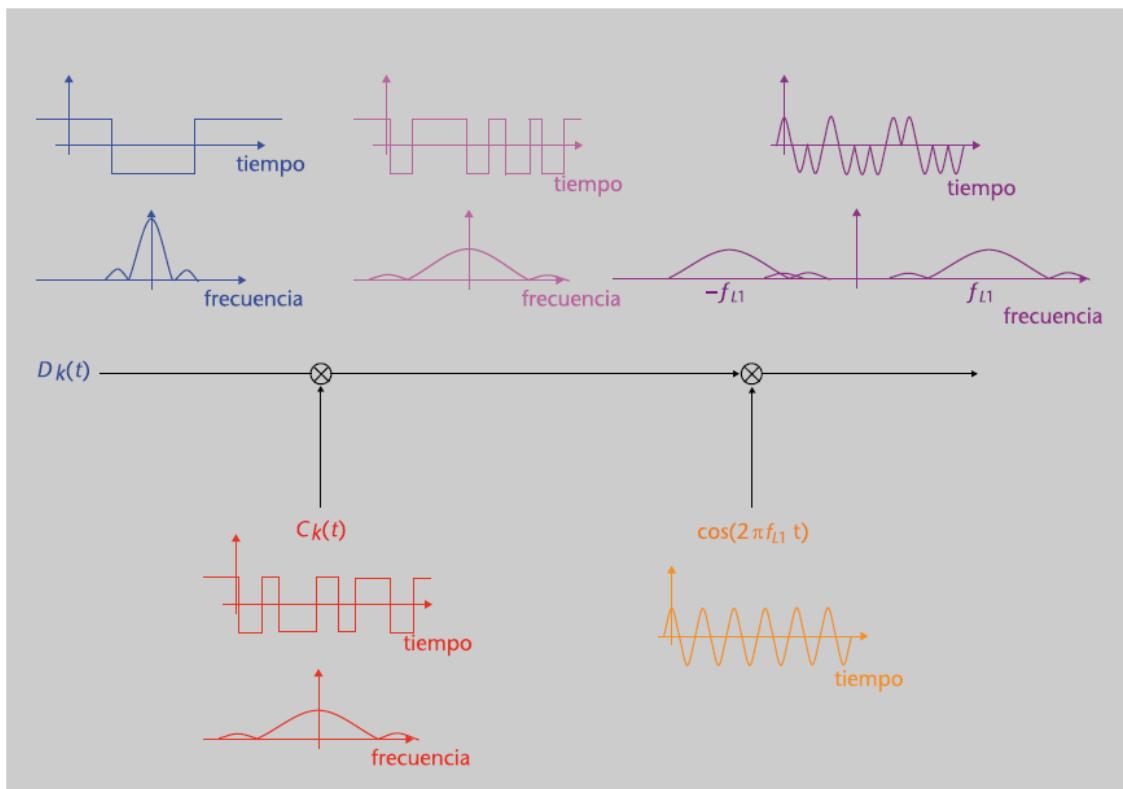


Fig. 4.11. Proceso de modulación de la señal GPS

El código C/A, $C_k(t)$ señal roja, tiene un ancho de banda mayor que el mensaje de navegación, $D_k(t)$ señal azul, porque el tiempo de chip es mucho menor que el tiempo de bit. Al multiplicar ambas señales, obtenemos la señal rosa DSSS resultante con un ancho de banda prácticamente igual a la señal de espectro ensanchado, que posteriormente modularemos a la frecuencia portadora que corresponda.

Podemos modelar el código C/A haciendo uso de una modulación BPSK, incorrelada consigo misma y aleatoria. Su pulso conformador es el rectángulo del chip, con magnitud unidad. De igual manera, también podemos considerar el mensaje de navegación como una modulación BPSK, incorrelada consigo misma y aleatoria. El pulso conformador es el rectángulo del bit, también con magnitud unidad.

Como el espectro del código C/A es mucho más ancho que el espectro del mensaje de navegación, podemos modelar este último como una delta que, convolucionada con el espectro del código C/A, provoca que el espectro de la señal DSSS sea razonablemente parecido al de la señal de espectro ensanchado.

La modulación DSSS es interesante en varios sentidos. En primer lugar, los códigos están basados en secuencias pseudoaleatorias que permiten obtener buenas precisiones en las estimaciones de las distancias entre el usuario y los satélites. En segundo lugar, el hecho de distribuir la energía de la señal en un espectro más amplio permite mejorar la robustez del sistema ante interferencias de banda estrecha. Es decir, la señal interferente de banda estrecha causará un impacto menor ya que afectará a un rango de frecuencias menor de la señal GPS. Y, en tercer lugar, el uso de la modulación DSSS

genera más transiciones y de mayor pendiente de las que se tendrían en una modulación BPSK normal.

Para ilustrar esta última ventaja, vamos a pensar en las transiciones que se producen en un cambio de bit. Dos de los parámetros que definen el mensaje de navegación son los siguientes:

$$B_{Mensaje\ Navegación} = 50\ Hz$$

$$T_{bit} = 20\ ms$$

Observamos que ambos parámetros son inversamente proporcionales y definen cómo son las transiciones. Los sistemas reales no funcionan con cambios abruptos como los observados en la Fig. 4.1 y, por tanto, existen transiciones al pasar de un bit al siguiente. Resulta razonable pensar que cuanto más abruptas sean estas transiciones, mejor funcionará el sistema porque el receptor terrestre cometerá menos errores en detección.

Las transiciones son inversamente proporcionales al ancho de banda, por tanto, proporcionales al tiempo de bit. Si deseamos tener transiciones pequeñas, abruptas, debemos buscar valores grandes de ancho de banda para así tener un tiempo de bit lo más pequeño posible. El $B_{Mensaje\ Navegación} = 50\ Hz$ es un valor pequeño comparado con el $B_{CA} \approx 1\ MHz$. Nos gustaría acercarnos lo máximo posible a este último valor, ya que eso nos aseguraría tener transiciones abruptas, y es precisamente lo que conseguimos con la modulación DSSS, ya que conseguimos transmitir el mensaje de navegación con un ancho de banda de 1 MHz en la señal resultante DSSS.

4.6.2 Técnica de acceso al medio CDMA

El uso de diferentes códigos en cada uno de los satélites permite que el receptor pueda distinguir señales recibidas de distintos satélites simultáneamente en la misma frecuencia. Este proceso se conoce como técnica de acceso al medio CDMA. La señal GPS que envía el satélite es la compuesta por la señal resultante DSSS modulada a una frecuencia portadora determinada. Como hemos visto en el apartado anterior, la señal DSSS resultante se forma como el producto del mensaje de navegación y el código C/A del satélite.

En recepción, obtendremos la señal GPS distorsionada por una serie de efectos. Una vez el receptor extraiga la portadora y tenga la señal compleja en banda base, deberá tratar la señal DSSS resultante enviada por el satélite. La clave de la técnica CDMA reside en que el producto entre la señal DSSS resultante y la versión alineada del código C/A modulada a una determinada frecuencia residual, permitirá al receptor conocer la distancia que le separa del satélite, obtener su posición y extraer el mensaje de

navegación. Esta técnica, además, nos permitirá saber qué satélite envía la señal GPS, ya que utilizaremos su código C/A en recepción para demodular la información.

Para reforzar todo lo expuesto anteriormente, mostramos dos esquemas a alto nivel (Figs. 4.12 y 4.13) para entender de manera básica cómo se transmiten y se reciben las señales en el Sistema GPS.

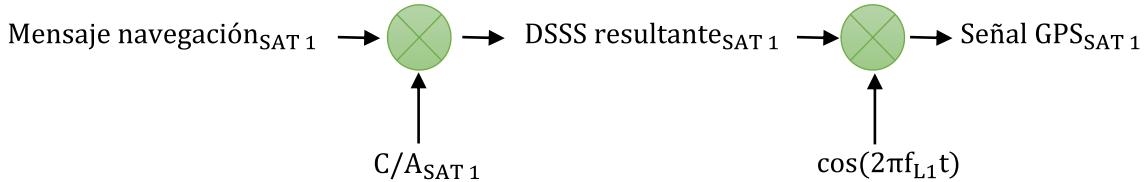


Fig. 4.12. Esquema de transmisión a alto nivel de la señal GPS

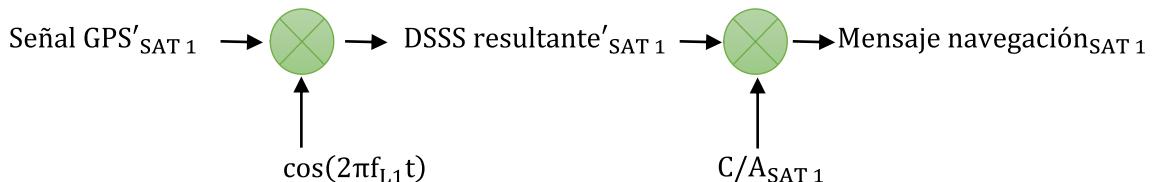


Fig. 4.13. Esquema de recepción a alto nivel de la señal GPS

Como hemos comentado anteriormente, el uso de la técnica CDMA permite saber qué satélite envía la señal GPS, porque utilizaremos su código C/A en recepción. Analizando este hecho desde otro punto de vista, la técnica CDMA permite separar las señales GPS de los diferentes satélites, ya que el receptor solo adquirirá la señal GPS del satélite k correctamente si utiliza el código C/A asociado al satélite.

El código C/A se calcula en la función *cacode.m*, la forma temporal y los espectros del código C/A y del mensaje de navegación están implementados en la función *graficas.m*, y la señal GPS está generada en la función *main_ideal.m*. Todas ellas contenidas en el apartado Anexo.

Aquí finaliza el cuarto capítulo, donde hemos presentado el esquema de transmisión del Sistema GPS, así como las señales que intervienen en él. En este punto, el satélite está listo para enviar la señal a través del espacio y, en el siguiente capítulo, Implementación del receptor GPS, ampliaremos los procesos que debe realizar el receptor para conseguir calcular la distancia que le separa del satélite y obtener su posición.

5. Implementación del receptor GPS

En este capítulo estudiaremos los principios básicos de un receptor GPS: analizaremos la estimación de una serie de parámetros contenidos en la señal GPS, que ha sufrido variaciones a causa de su viaje por el espacio hasta llegar al receptor (Fig. 5.1). En el siguiente capítulo nos aprovecharemos de las estimaciones para calcular la distancia que separa al receptor del satélite.

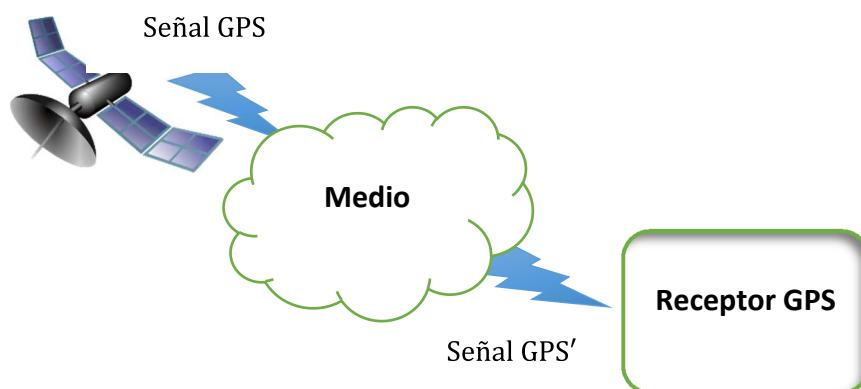


Fig. 5.1. Las señales GPS transmitida y recibida no son exactamente iguales

5.1 Bloques de un receptor GPS

El objetivo de un receptor GPS es determinar su posición. Para ello, el receptor debe realizar una serie de tareas de manera secuencial a partir de la señal GPS captada por la antena, ilustradas en el siguiente diagrama de bloques (Fig. 5.2).

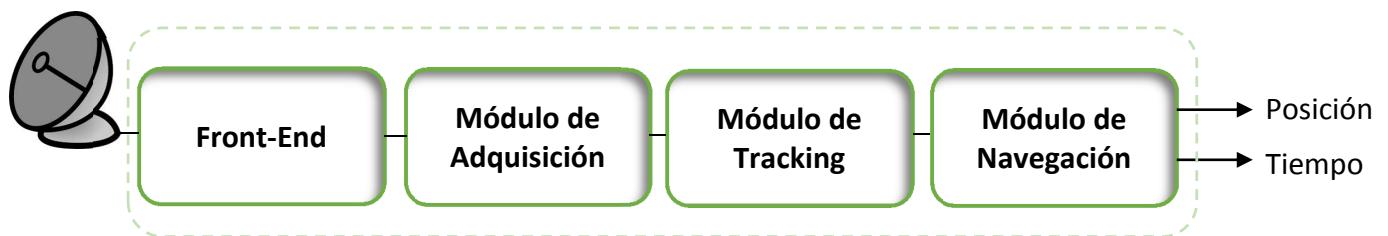


Fig. 5.2. Arquitectura de un receptor GPS

En las siguientes secciones analizaremos cada uno de los cuatro bloques que componen un receptor GPS, haciendo especial hincapié en los módulos de adquisición y tracking, objetos de análisis del presente proyecto.

5.2 Front-End

Para poder empezar a trabajar con un receptor software GPS, la única información necesaria es la señal GPS compleja en banda base muestreada. Por tanto, debemos realizar una conversión a banda base y posteriormente muestrear la señal resultante. La señal GPS, real y paso banda, es recibida por la antena y es el módulo de *front-end* el que se encarga de hacer su conversión a banda base, para obtener así la señal GPS compleja.

Una vez en banda base, deberemos muestrear la señal. El muestreo se realiza a una frecuencia que cumpla la condición de Nyquist y los datos provenientes del muestreo serán guardados en memoria para posteriormente ser procesados por los siguientes módulos. Es decir, a la salida del *front-end* obtendremos la señal GPS en banda base y muestreada, por lo que la etapa de adquisición del receptor dispondrá de un flujo constante de muestras discretas (Fig. 5.3).

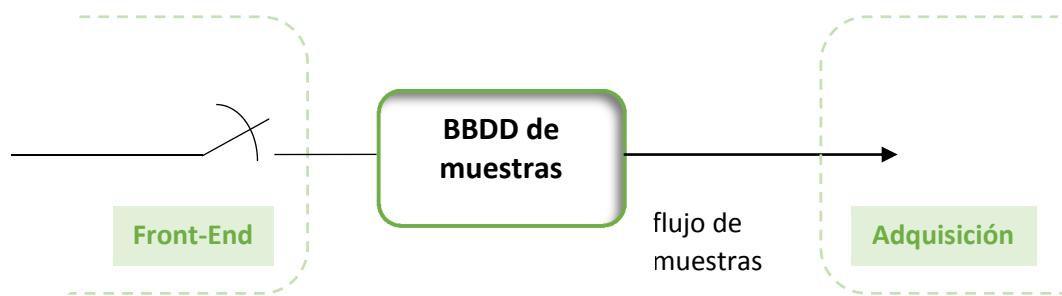


Fig. 5.3. Esquema funcional de la salida del *front-end* de un receptor software

Típicamente, la etapa de *front-end* consta de un amplificador de bajo ruido, que filtra la señal paso banda y amplifica la señal resultante, una o varias etapas de conversión a frecuencia intermedia, un filtrado antialiasing ya en banda base y una conversión analógico – digital (ADC) (Fig. 5.4).



Fig. 5.4. Diagrama de bloques de la etapa de *front-end*

5.3 Módulo de adquisición

A la entrada del módulo de adquisición tenemos un flujo constante de muestras de la señal GPS listas para ser procesadas. El objetivo de esta etapa es determinar qué satélites son visibles por el usuario y utilizar las señales recibidas de aquellos que son visibles para estimar dos parámetros clave: la desviación Doppler y el retraso de código de la señal GPS recibida. La desviación Doppler de la señal GPS está íntimamente ligada a su fase y veremos su relación en el siguiente capítulo. Por ahora, únicamente decir que si encontramos la desviación Doppler, también encontraremos la fase de la señal GPS, parámetro vital en el módulo de tracking.

Como hemos comentado en el capítulo introductorio, hemos implementado la señal GPS generándola a la salida del *front-end*. De esta manera, disponemos de un flujo constante de muestras que recogen la estructura de la señal GPS y los efectos que se habrían producido en ella durante su viaje por el espacio hasta llegar al receptor.

Durante esta sección únicamente consideraremos el efecto Doppler que sufre la portadora de la señal GPS recibida. La existencia de la frecuencia Doppler la estudiaremos en el siguiente capítulo, por ahora decir que el movimiento del satélite a través del espacio provoca un desplazamiento en frecuencia de la señal GPS en recepción. Este desplazamiento hace que, aunque la etapa de *front-end* haya realizado la conversión de la señal GPS a banda base demodulando por la portadora, aún quede una frecuencia residual que haya que estimar. En el siguiente capítulo ampliaremos este y otros efectos para analizarlos detalladamente.

Recordemos que nuestra señal GPS está modulada en la banda L1, y está compuesta por el mensaje de navegación y el código C/A. Además, ha sufrido un desplazamiento Doppler en frecuencia. Por tanto, si somos capaces de encontrar el inicio del código C/A del satélite en cuestión y el valor del desplazamiento Doppler que ha experimentado la señal, seremos capaces también de obtener la pseudodistancia. Una vez calculada, el módulo de navegación procesará el valor para obtener la posición del usuario.

De esta manera, debemos hacer una búsqueda en tiempo y frecuencia, mediante la generación de una réplica de la señal captada en recepción. El objetivo es obtener una réplica alineada y modulada a la frecuencia Doppler, respecto a la señal GPS.

5.3.1 Remuestreo de la señal réplica

Para lo que resta de sección nos referiremos a la señal GPS como la señal a la salida del *front-end*, y la señal réplica como el código C/A generado por la etapa de adquisición. Un aspecto vital para empezar la etapa de adquisición es acondicionar el número de muestras del código C/A de la señal réplica.

Como hemos comentado anteriormente, la señal GPS ha sido muestreada a una determinada frecuencia de muestreo antes de llegar a la etapa de adquisición. Por tanto, el número de muestras contenidas en un chip de la señal GPS es:

$$\#muestras_{chip} = f_m \cdot T_{chip}$$

Recordando que el tiempo de chip del código CA es:

$$T_{chip} \approx 1\mu s$$

Y con una frecuencia de muestreo dada:

$$f_m = 5.456 \text{ MHz}$$

Resolvemos que el número de muestras que hay en cada chip de la señal GPS es:

$$\#muestras_{chip} = 5.456 \text{ MHz} \cdot 1 \mu s = 5.3 \text{ muestras/chip}$$

Por tanto, concluimos que si utilizamos esta frecuencia de muestreo, cada chip de la señal GPS que entra en la etapa de adquisición está representada por 5.3 muestras. Por el contrario, cada chip del código C/A original de la señal GPS enviada por el satélite está representado por 1 muestra. El hecho de muestrear en recepción ha provocado que un chip ya no esté representado por 1 muestra, sino por 5.3.

Por esta razón, debemos acondicionar el código C/A, la señal réplica, y conseguir que cada chip del código C/A que generemos tenga 5.3 muestras para así compararlo con el entrante en la etapa de adquisición.

En primer lugar, el receptor debe generar el código C/A del satélite visible de 1023 chips y 1023 muestras, resultado de la función *cacode.m* alojada en el capítulo Anexo. Si un chip está representado por 5.3 muestras, una muestra representa 0.1875 chips. Como queremos representar los 1023 chips, debemos generar un vector que, en intervalos de 0.1875, llegue hasta 1023 – 0.1875. Por tanto, debemos repetir la muestra $i + 1$ del código C/A original tantas veces como aparezca la parte entera i en el vector generado, con $0 \leq i < 1023$.

Acompañaremos la explicación anterior con un ejemplo. Con la sentencia:

```
a=(0:5.333*1023-1)*(1/5.333);
```

obtenemos el siguiente vector:

$$a = [0 \ 0.1875 \ 0.3750 \ 0.5625 \ 0.75 \ 0.9375 \ 1.1250 \ \dots \ 1022.8125]$$

Posteriormente, únicamente tenemos que obtener la parte entera del vector anterior, y quedarnos con la muestra módulo 1023 que representa, sumándole uno porque las muestras en MATLAB® empiezan en 1:

```
b=floor(a);
Réplica=CA(mod(b,1023)+1);
```

De esta manera:

$$b = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ \dots \ 1022]$$

Con lo que las primeras seis muestras de Réplica son la primera muestra del código CA original, y así sucesivamente:

$$\text{R\'eplica} = [CA(1) \ CA(1) \ CA(1) \ CA(1) \ CA(1) \ CA(1) \ CA(2) \ CA(2) \ \dots \ CA(1023)]$$

De esta manera, en media, cada chip de la señal réplica está formado por $5.\hat{3}$ muestras, igual que el código C/A de la señal GPS que entra en el módulo de adquisición. En este punto sí que ambas señales son comparables, y podemos empezar con el proceso de extracción de los parámetros de interés.

5.3.2 Búsqueda bidimensional en tiempo y frecuencia

Una vez el receptor haya generado y remuestreado el código C/A, la señal réplica, debemos saber en qué muestra empieza el código C/A de la señal GPS que entra en el módulo de adquisición. Por tanto, el receptor hará una búsqueda en tiempo para obtener el retraso del código. Además, como hemos apuntado anteriormente, el movimiento relativo entre satélite y usuario provoca un desplazamiento Doppler en la señal GPS que capta la antena receptora, por lo que, aunque el *front-end* hace la conversión a banda base (idealmente a frecuencia nula), existe una frecuencia residual, frecuencia Doppler, que debemos extraer para calcular la pseudodistancia. Es lo que denominamos búsqueda en frecuencia.

La existencia de este error en frecuencia juntamente con el retardo que experimenta el código C/A de la señal GPS, ambos desconocidos, hace que el proceso de adquisición implique una búsqueda bidimensional en tiempo y frecuencia. Hay múltiples métodos para realizar esta búsqueda, y en este proyecto se ha utilizado la *adquisición serial*, que se implementa haciendo la correlación entre la señal GPS y la señal réplica para todos los valores posibles de retraso de código y frecuencia.

Buscaremos el desplazamiento Doppler en intervalos de 500 Hz, entre -10 KHz y 10 KHz ; ver Tsui (2000), página 39. Para cada uno de ellos, realizaremos dos operaciones. La primera de ellas consiste en extraer este desplazamiento de la señal GPS, obteniendo (si acertamos con el desplazamiento) el código C/A del satélite en cuestión junto con el mensaje de navegación. Y en segundo lugar, haremos la correlación entre la señal que acabamos de obtener junto con la señal réplica, que es el código C/A que genera el receptor, para obtener el retraso de código. Esta última operación la repetiremos 41 veces, para todos los desplazamientos, e iremos guardando el resultado de cada una de ellas por filas en una matriz:

$$\text{Matriz}_{\text{correlación}} = \begin{pmatrix} \text{muestras de correlación para el primer desplazamiento} \\ \text{muestras de correlación para el segundo desplazamiento} \\ \vdots \\ \vdots \\ \text{muestras de correlación para el desplazamiento 41} \end{pmatrix}$$

Finalmente, haremos el módulo al cuadrado de los valores de la matriz anterior, buscaremos la posición de su máximo y obtendremos el desplazamiento Doppler y el retraso del código. El siguiente esquema (Fig. 5.5) resume lo anteriormente expuesto.

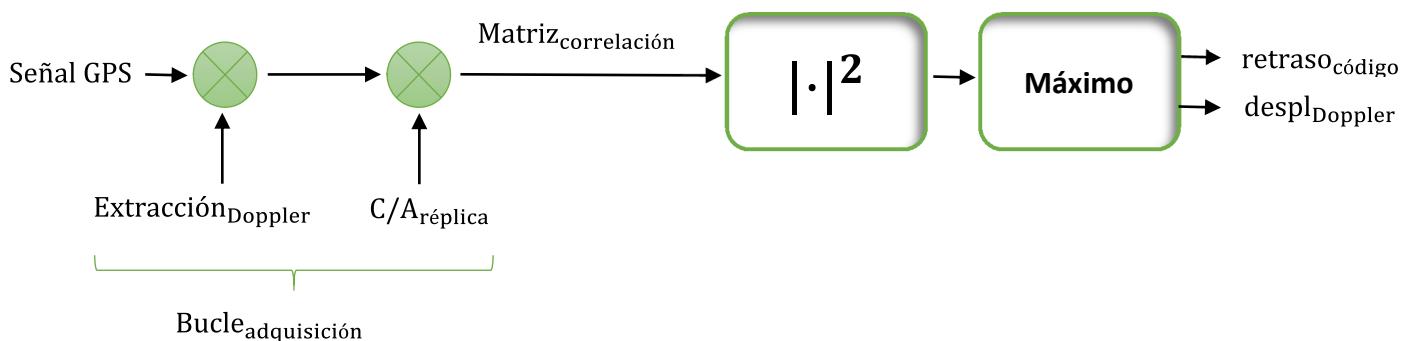


Fig. 5.5. Operaciones del módulo de adquisición

Podemos diferenciar diversos métodos de *adquisición serial* atendiendo a las longitudes de las señales involucradas. En el presente proyecto hemos trabajado con dos ejemplos concretos de longitudes, que a continuación detallaremos, y para cada uno de los dos métodos presentaremos las gráficas de adquisición para dos tipos de señales GPS: en primer lugar, la señal GPS que hemos generado a la salida del *front-end* (señal GPS simulada); y, en segundo lugar, generaremos las gráficas de adquisición para una señal GPS real, con muestras reales captadas por un receptor GPS.

5.3.2.1 Dos períodos de la señal GPS y uno de la réplica

En este subapartado trabajaremos con 2 períodos de la señal GPS y 1 periodo de la señal réplica, para que no haya ambigüedades y podamos obtener siempre un periodo completo del código C/A dentro de la señal GPS. En el siguiente subapartado, donde permutaremos las longitudes, observaremos más claramente la ventaja de utilizar dos períodos de la señal GPS.

Un periodo de la señal réplica, el código C/A, tiene:

$$\#muestras_{periodo} = \#muestras_{chip} \cdot N_{chips} = 5.3 \cdot 1023 = 5456 \text{ muestras}$$

Y este código C/A, como lo genera el receptor, está perfectamente alineado, es decir, su primera muestra corresponde a la primera muestra del primer chip. Las muestras tienen valores +1 o -1 y el esquema en muestras es el siguiente (Fig. 5.6):

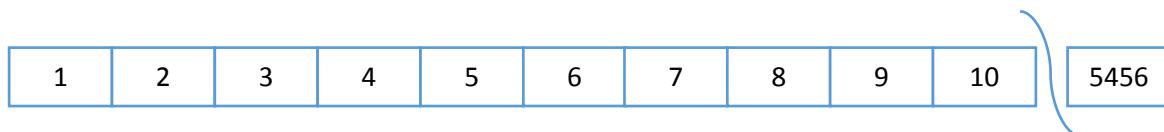


Fig. 5.6. Esquema en muestras de la señal réplica perfectamente alineada

Por otro lado, hemos utilizado 2 períodos de la señal GPS, por lo que trabajaremos con 10912 muestras de señal. Además, como hemos comentado anteriormente, no sabemos en qué muestra empieza su código C/A, por lo que es más que probable que se encuentren desalineadas (Fig. 5.7):

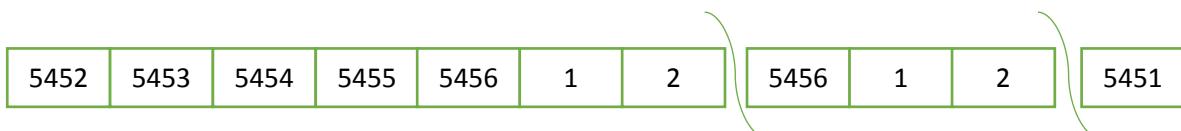


Fig. 5.7. Esquema en muestras de la señal GPS desalineada

Podemos observar como la primera muestra del primer chip del código C/A de la señal GPS está retrasada 5 muestras, con lo que nuestro algoritmo de adquisición nos debe devolver un $retraso_{código} = 5$, ya que este valor es el número de muestras de la señal GPS que debemos recorrer hasta dar con la primera del primer chip. Posteriormente estudiaremos la manera de conseguir este valor.

Siguiendo con el ejemplo, para cada uno de los 41 desplazamientos Doppler realizaremos la correlación entre las señales representadas anteriormente. Para hacer la correlación más cómodamente, giraremos la señal réplica y haremos la convolución de las dos señales. Como nuestro objetivo es encontrar el inicio del código, únicamente guardaremos las muestras resultado del producto de 5457 muestras de ambas señales (muestras centrales de la convolución), el resto las descartaremos porque en el resultado de esas muestras seguro que no habrá un máximo de correlación. Veremos más claramente este razonamiento en el siguiente estudio de casos de la convolución.

Siendo la señal verde de 10912 muestras la señal GPS con el desplazamiento Doppler extraído, y la señal azul de 5456 muestras la réplica, vamos a ver esquemáticamente los estados de la convolución entre ambas señales, siendo la señal azul sombreada la misma señal azul al final del estado de la convolución:

Estado 1

Empieza la convolución y se produce el producto de ambas señales hasta que la señal réplica se multiplica muestra a muestra por completo con la señal GPS con el desplazamiento extraído (Fig. 5.8):

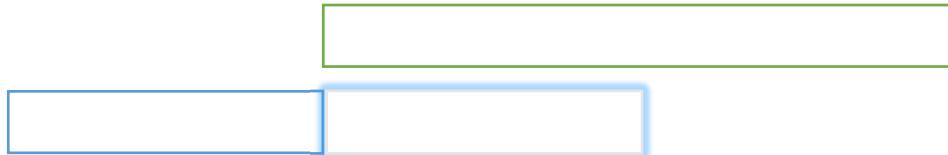


Fig. 5.8. Esquema del primer estado de la convolución

Estado 2

La parte central de la convolución. Durante 5457 muestras ambas señales se multiplican muestra a muestra, hasta el momento en que la señal réplica se desplaza fuera de la señal GPS con el desplazamiento extraído (Fig. 5.9):



Fig. 5.9. Esquema del segundo estado de la convolución

Estado 3

Parte final de la convolución. Se produce el producto desde que empieza a salir la señal réplica hasta la última muestra de la convolución (Fig. 5.10):



Fig. 5.10. Esquema del tercer estado de la convolución

Aprovechándonos de las propiedades de los códigos pseudoaleatorios que hemos analizado en el capítulo anterior, cuando la señal réplica esté perfectamente alineada con la señal GPS se producirá un pico de correlación. Es por ello que únicamente nos interesan las muestras resultado de la convolución del estado 2, porque dentro de las primeras 5456 muestras de la señal verde seguro que encontramos el inicio de su código C/A, y posteriormente recorremos la señal verde hasta la muestra final del estado 2.

Por tanto, la correlación de ambas señales tiene el siguiente número de muestras:

$$\#muestras\ totales_{corr} = 10912 + 5456 - 1 = 16367$$

Pero únicamente nos quedaremos con las 5457 centrales:

$$\#muestras\ necesarias_{corr} = 5457$$

Así, la matriz de correlación tendrá 41 filas y 5457 columnas:

$$Matriz_{correlación\ 41x5457}$$

De esta manera conseguimos no consumir tanta memoria y aligerar el procesado de los algoritmos. Como vemos, el resultado del proceso de adquisición es una matriz en cuyas celdas se almacenan los valores a la salida de la correlación. En aquellas celdas cuya hipótesis de retardo de código y desplazamiento Doppler se acerque más al valor exacto que presenta la señal GPS se observará un pico de correlación. Si este pico es suficientemente grande como para superar un umbral de decisión previamente fijado,

confirmaremos la presencia del satélite que se está analizando, y sabremos los valores del retraso de código y desviación Doppler de la señal GPS.

Una vez obtenida la matriz de correlación, haremos el módulo al cuadrado de cada una de sus celdas, y debemos calcular la fila y la columna de su valor máximo. Tal y como hemos diseñado la matriz, la fila tendrá relación con el desplazamiento Doppler y el valor de la columna estará íntimamente ligado con el retraso de código.

Utilizando las siguientes definiciones:

$$x = \text{fila donde se encuentra el máximo}$$

$$y = \text{columna donde se encuentra el máximo}$$

podemos obtener los valores del desplazamiento Doppler y el retraso del código de la siguiente manera:

$$\text{desplazamiento}_{\text{doppler}} = -10 \text{ KHz} + (x - 1) * 500 \text{ Hz}$$

$$\text{retraso}_{\text{código}} = y - 1$$

Podemos pensar ambas expresiones anteriores con dos ejemplos.

Si el máximo se encuentra en la primera fila, el valor del desplazamiento Doppler debe ser de -10 KHz , ya que es el valor inicial. Si el máximo está en la fila 23, el valor del desplazamiento Doppler debe ser de 1 MHz .

Para el retraso de código, vamos a recurrir a un ejemplo gráfico. Recordemos que el retraso del código es el número de muestras que hay que desplazarse hasta encontrar la primera del primer chip del código C/A de la señal GPS. Si tenemos los siguientes valores de la convolución (Fig. 5.11):

0.8	9	27	3	1.9	1050	59	10	9	98
-----	---	----	---	-----	------	----	----	---	----

Fig. 5.11. Ejemplo de retraso de código igual a 5

El máximo se encuentra en la sexta columna. Por lo tanto, el código C/A de la señal GPS empieza justo en esa muestra. Para alinear esta señal con la señal réplica debemos desplazarnos a lo largo de 5 muestras hasta llegar hasta ella. Por lo tanto, en este

ejemplo, el retraso del código son 5 muestras. Si el máximo hubiera estado en la primera columna, las señales hubieran estado alineadas y el retraso de código hubiera sido nulo.

5.3.2.1.1 Señal GPS simulada

Una vez analizado el algoritmo y estudiado sus detalles, generaremos las gráficas de la matriz de correlación para ver cómo se comporta. En este primer subapartado la señal GPS es la señal GPS simulada generada a la salida del *front-end*. La siguiente figura representa los valores de la matriz de correlación en función del desplazamiento Doppler y el retraso de código normalizado a 1023 chips (Fig. 5.12):

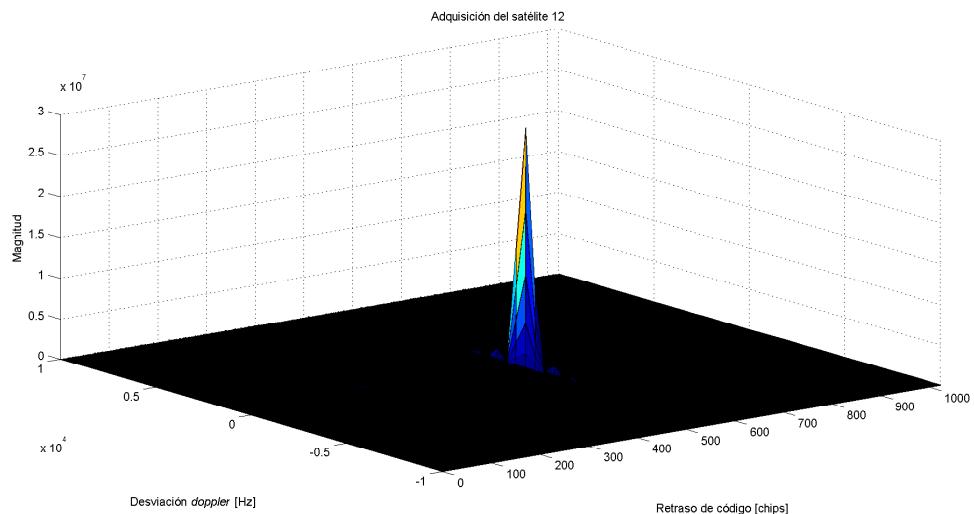


Fig. 5.12. Adquisición positiva del satélite asociado al código PRN 12

La Fig. 5.12 ejemplifica el hecho de que el usuario recibe la señal GPS proveniente del satélite 12 y el receptor genera la señal réplica asociada al mismo número, alineada y modulada, observando claramente el pico de correlación. De esta manera, podemos concluir que se ha adquirido el satélite correctamente, y disponemos del retraso de código y del desplazamiento Doppler de la señal GPS.

En cambio, si el receptor genera un código C/A diferente al que recibe, no existirá el pico de correlación y no podremos adquirir la señal del satélite 12 correctamente. En la siguiente figura observamos el comportamiento de la matriz de correlación si la señal réplica corresponde al código C/A del satélite 11 (Fig. 5.13):

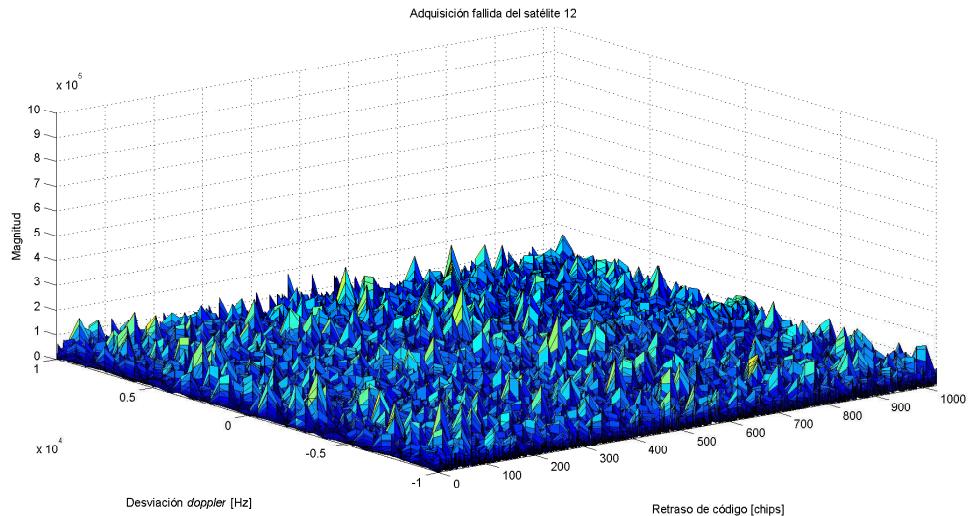


Fig. 5.13. Adquisición fallida del satélite asociado al código PRN 12

Como podemos observar en la figura anterior (Fig. 5.13), no existe pico de correlación: el código C/A de la señal GPS no se ha podido alinear con la señal réplica porque son códigos diferentes.

5.3.2.1.2 Señal GPS real

En este nuevo subapartado utilizaremos muestras reales de señal GPS¹⁹ captadas por un receptor GPS. La señal GPS real, a diferencia de la anterior, llega al receptor afectada por múltiples perturbaciones. Además, la desviación Doppler afecta a toda la señal GPS, no solo a la portadora. Por todo ello, y desconociendo los demás efectos que pueda incluir la señal, observaremos cómo los picos de la matriz de correlación, si bien están definidos, no presentan una magnitud comparable a los de la señal GPS simulada.

En figuras posteriores veremos como la desviación Doppler parte de una frecuencia intermedia, ya que durante la adquisición de la señal GPS real trabajamos con un valor de frecuencia intermedia igual a 1.364 MHz. Los dos siguientes subapartados únicamente presentarán los valores de la matriz de correlación, ya que se ha utilizado el mismo código que en los subapartados anteriores.

¹⁹ Hemos utilizado el modo 3 de la señal GPS captada por el receptor descrito en ccar.colorado.edu/gnss

A continuación representamos la adquisición positiva del satélite asociado al código PRN 12 (Fig. 5.14).

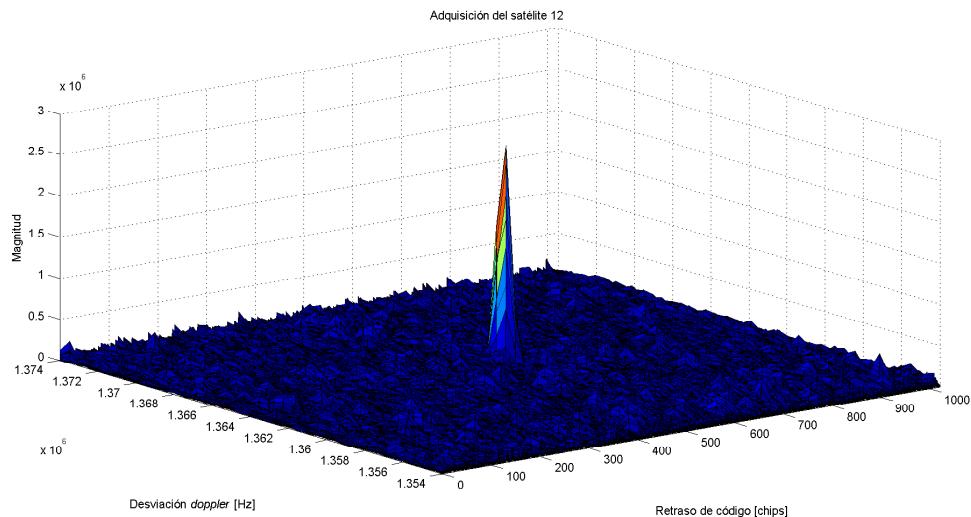


Fig. 5.14. Adquisición positiva del satélite asociado al código PRN 12

Comparando la Fig. 5.12 con la Fig. 5.14, observamos que el pico de correlación de esta última tiene un valor menor, y también presenta valores de correlación más altos para el resto de valores de desplazamiento Doppler y retraso de código.

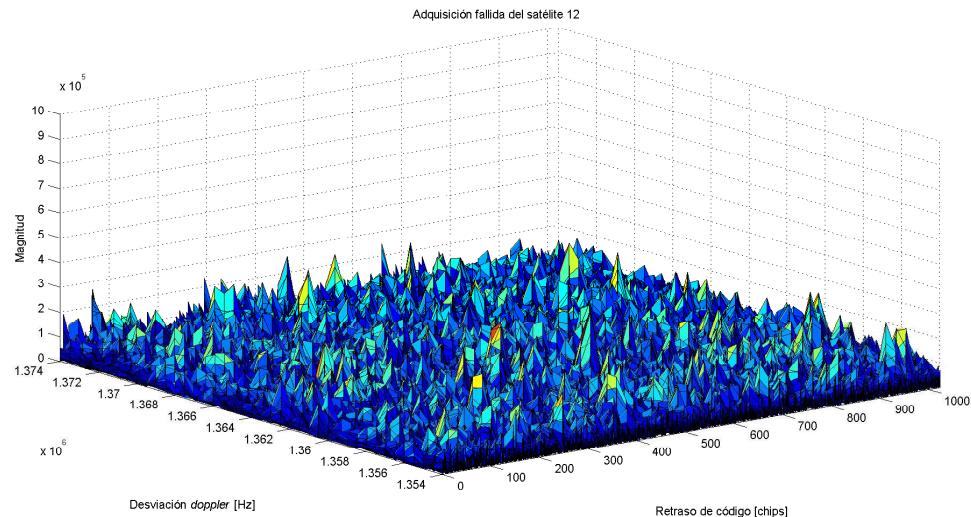


Fig. 5.15. Adquisición fallida del satélite asociado al código PRN 12

Cuando se produce la no visibilidad de un satélite para el receptor (Fig. 5.15), observamos cómo los valores de correlación también son más altos, comparándolos con los de la Fig. 5.13, porque la señal GPS real es más ruidosa que la simulada.

5.3.2.2 Un periodo de la señal GPS y dos de la réplica

Este subapartado sigue los mismos pasos que el anterior, aunque con algunos matices debido a las nuevas longitudes de las señales involucradas. Ahora disponemos de los siguientes esquemas de muestras (Figs. 5.16 y 5.17):

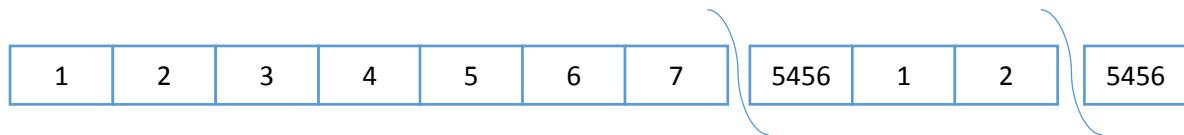


Fig. 5.16. Esquema en muestras de la señal réplica perfectamente alineada

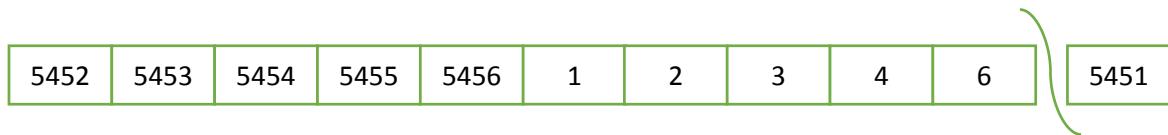


Fig. 5.17. Esquema en muestras de la señal GPS desalineada

Con esta elección de longitudes es más que probable que no podamos obtener un periodo completo seguido del código C/A de la señal GPS (señal verde), ya que la primera muestra de la señal debería ser la primera muestra asociada al primer chip, y es improbable que eso ocurra. Esta limitación provoca que, al hacer la correlación con la réplica, se produzcan intervalos de muestras donde ambas señales coincidan parcialmente, produciendo picos de correlación de un valor menor al pico original (Fig. 5.18).

Ahora no podemos reducir las medidas de la matriz, por lo que tendremos 16367 columnas, siendo el programa más ineficiente que en el subapartado anterior:

$$\text{Matriz}_{\text{correlación}} \ 41 \times 16367$$

La expresión para encontrar el retraso de código también cambia, siendo ambas expresiones las siguientes:

$$\text{desplazamiento}_{\text{doppler}} = -10 \text{ KHz} + (x - 1) * 500 \text{ Hz}$$

$$\text{retraso}_{\text{código}} = y - 5456$$

5.3.2.2.1 Señal GPS simulada

Con los anteriores resultados, podemos representar los valores de la matriz de correlación asociados a la adquisición positiva de la señal GPS del satélite 12 (Fig. 5.18):

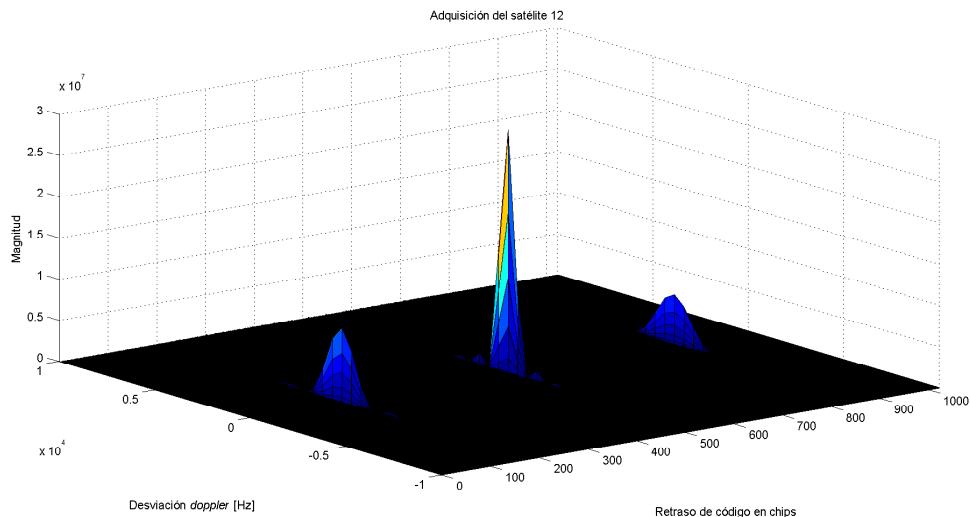


Fig. 5.18. Adquisición positiva del satélite asociado al código PRN 12

Como podemos observar en la figura anterior, aparecen dos lóbulos que representan dos picos de menor magnitud que el original y, dependiendo del buen funcionamiento del receptor, este hecho podría provocar algún error en detección (por ejemplo, si el umbral de decisión estuviera mal definido).

En caso de que no se detecte el satélite, observamos los siguientes valores de la matriz de correlación (Fig. 5.19), concluyendo que el satélite no es visible para el receptor.

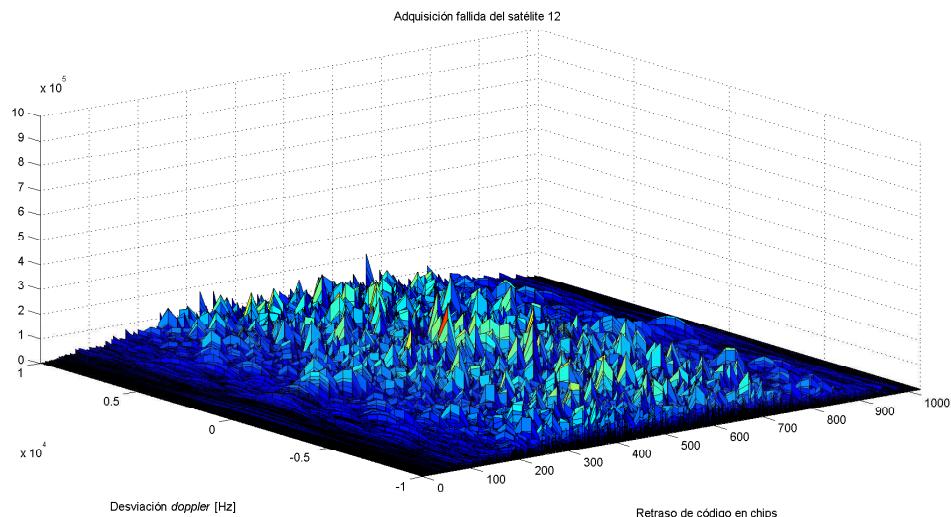


Fig. 5.19. Adquisición fallida del satélite asociado al código PRN 12

Acabamos de ver gráficamente lo que postulábamos de manera teórica, y la señal GPS real también tendrá un comportamiento parecido. Por tanto, es preferible trabajar con dos períodos de la señal GPS, así nos aseguramos de tener un periodo completo de su código C/A, el procesador es más eficiente, y evitamos posibles problemas en detección.

5.3.2.2.2 Señal GPS real

También observamos la diferencia de magnitud en los valores de la matriz de correlación cuando comparamos la señal GPS simulada con la señal GPS real. Este hecho lo podemos comprobar comparando, por un lado, las gráficas Fig. 5.18 y Fig. 5.20 y, por otro, las gráficas Fig. 5.19 y la Fig. 5.21.

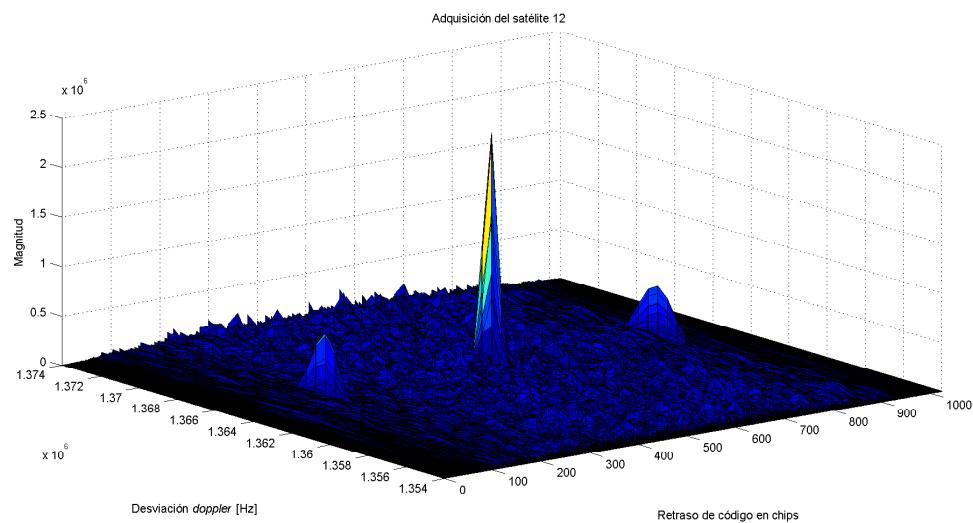


Fig. 5.20. Adquisición positiva del satélite asociado al código PRN 12

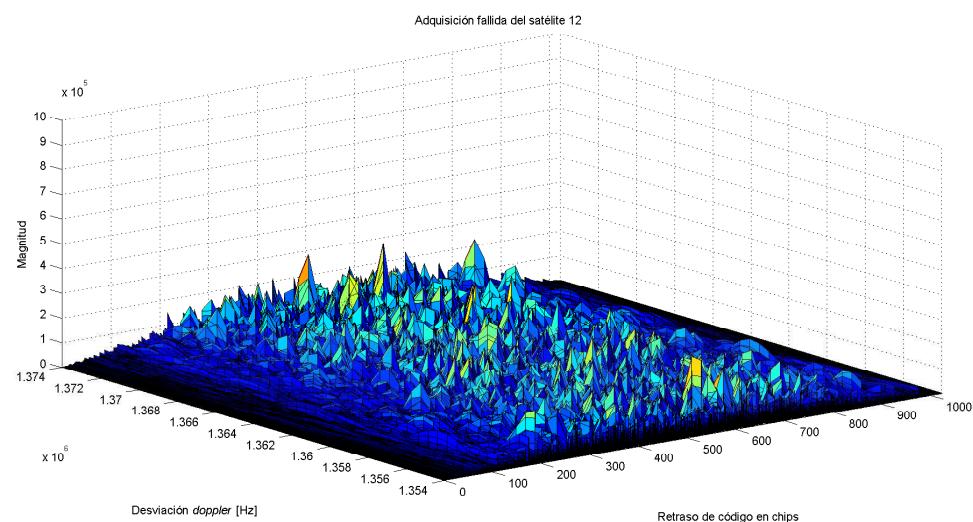


Fig. 5.21. Adquisición fallida del satélite asociado al código PRN 12

Hasta este punto, hemos demodulado la modulación CDMA en transmisión extrayendo la frecuencia portadora conocida y, posteriormente, hemos estimado los valores de retraso de código y desplazamiento Doppler. Lo hemos realizado de dos maneras diferentes, atendiendo a la longitud de las señales involucradas en la *adquisición serial*, y para cada una de ellas hemos utilizado una señal GPS simulada y una real. A partir de ahora utilizaremos la señal GPS simulada con la adquisición de dos períodos de la señal GPS y uno de la réplica.

Si las estimaciones de la etapa de adquisición fueran perfectas haríamos una demodulación precisa pero, aún y así, debemos hacer un seguimiento y refinar los dos parámetros obtenidos a la salida del módulo de adquisición ya que pueden variar a lo largo del tiempo, y de eso es justamente de lo que se encarga el módulo de tracking.

Las dos funciones que realizan la adquisición son *adq_1Inc2CA.m* y *adq_2Inc1CA.m*. La primera de ellas la realiza con un período de la señal GPS y dos de la réplica, y la segunda la realiza mediante dos períodos de la señal GPS y uno de la réplica.

5.4 Módulo de tracking

Como hemos comentado anteriormente, el módulo de tracking se encargará de refinar los dos valores obtenidos en el módulo de adquisición, ya que pueden experimentar pequeñas variaciones a lo largo del tiempo, por ejemplo, debido al movimiento relativo entre el satélite y el receptor. Ambos parámetros serán seguidos en el tiempo por dos arquitecturas en paralelo: el retraso de código será monitorizado por el DLL (*delay lock loop*) y la frecuencia Doppler lo será por el PLL (*phase lock loop*).

Tanto el DLL como el PLL son arquitecturas en lazo cerrado que obtienen una estimación precisa del parámetro de interés a base de comparar una señal de referencia con una réplica local. Esta comparación genera una señal error, que será utilizada para calcular una nueva réplica local en la siguiente iteración para ajustar de nuevo el error. A medida que el error tienda a 0, la señal de referencia y la réplica estarán más alineadas, objetivo del módulo de tracking para estimar los parámetros de forma precisa.

Antes de adentrarnos en los detalles del módulo de tracking tenemos que utilizar el valor de retraso de código obtenido en la etapa de adquisición. La señal GPS que entra al módulo de tracking empezará en la muestra $retraso_{código} + 1$. En el ejemplo de la Fig. 5.11 observamos cómo el retraso de código era igual a cinco muestras. En ese caso, deberíamos quedarnos con la señal a partir de la muestra seis ya que, si hemos adquirido correctamente, ahí empieza el código C/A de la señal GPS.

En cuanto a la desviación Doppler, al inicio del módulo de adquisición la señal GPS estaba modulada a una frecuencia residual y, tras la adquisición, hemos obtenido una estimación de ésta. Tal y como hemos diseñado el módulo de tracking, extraeremos la estimación del desplazamiento Doppler dentro de cada integración usándola como punto de partida para estimar la fase de la señal GPS.

5.4.1 Delay Lock Loop. DLL

En el DLL iremos integrando secuencialmente un periodo de la señal x_{DLL} junto con un periodo del código *prompt*, que es la señal réplica que usábamos en la adquisición. El proceso que sigue el DLL es el siguiente: el código *prompt* se correla con la señal de entrada x_{DLL} , y el resultado se introduce a un discriminador. La salida del discriminador ofrece una señal proporcional al error del parámetro que se está siguiendo, en este caso el retraso de código, que será suavizado por un filtro y entregado a un NCO_{DLL}²⁰ para que remuestre el código *prompt* y este se vuelva a correlar con la señal x_{DLL} (Fig. 5.22). De esta manera iremos refinando el valor del parámetro estimado.

A continuación explicaremos de manera detallada las operaciones que realiza cada uno de los bloques para una iteración i , sabiendo que son las mismas para el resto de iteraciones programadas. Además, al final del apartado realizaremos un ejemplo para entender cómo funciona la arquitectura DLL.

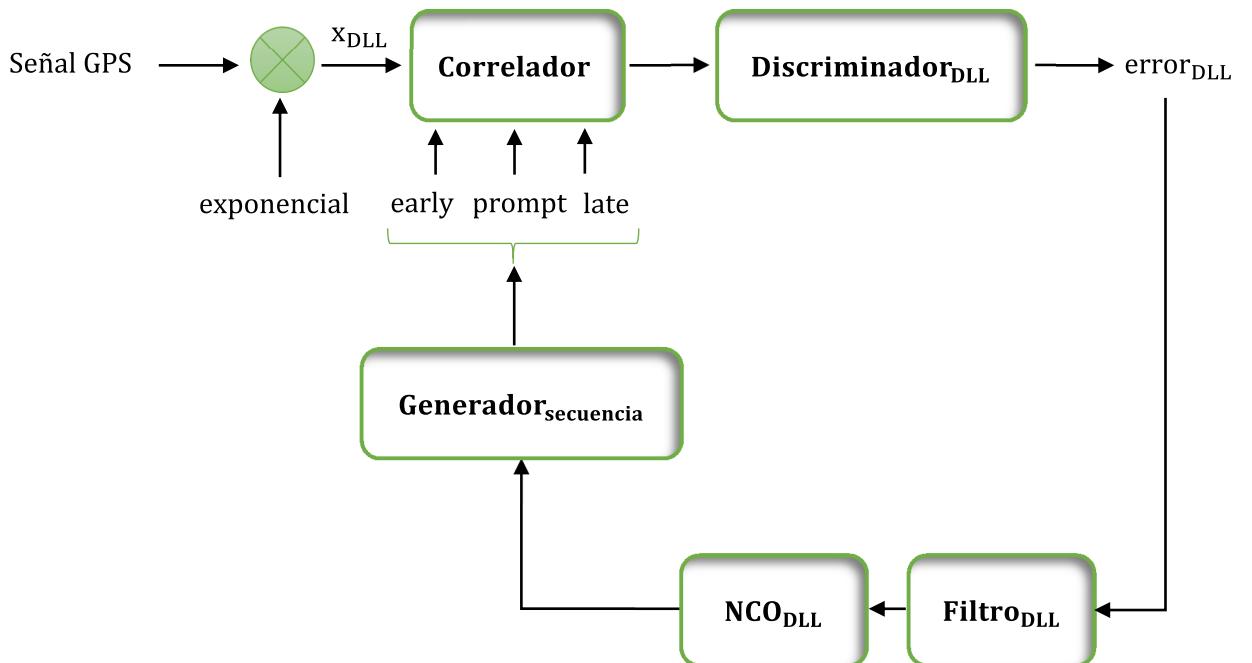


Fig. 5.22. Diagrama de bloques de la arquitectura DLL

²⁰ NCO son las siglas de *Numerically Controlled Oscillator*

5.4.1.1 Generación de x_{DLL}

La señal x_{DLL} se genera como el producto entre la señal GPS y la exponencial compleja, derivada de la salida de un generador de exponencial según la fase indicada por el NCO_{PLL} (Fig. 5.22). Como debemos integrar un periodo de la señal x_{DLL} junto con un periodo del código *prompt* asumiremos, como hasta ahora, que el periodo de ambas señales es de 5456 muestras. En el siguiente capítulo estudiaremos el caso en que el número de muestras por periodo es variable debido a que el desplazamiento Doppler también afecta al código C/A y al mensaje de navegación de la señal GPS, no solo a la portadora, como en este caso.

En la primera iteración debemos trabajar con las primeras 5456 muestras de ambas señales, lo que generará las 5456 primeras muestras de la señal x_{DLL} . En la siguiente tabla (Tabla 5.1) se muestra un ejemplo del rango de muestras para las primeras 4 iteraciones:

Iteración	Rango de muestras
1	1 - 5456
2	5457 - 10912
3	10913 - 16368
4	16369 - 21824

Tabla 5.1. Rango de muestras de la señal x_{DLL} , señal GPS y exponencial

5.4.1.2 Módulo correlador

El DLL necesita tres correladores: uno conocido como *prompt* y centrado en el retraso de interés, y dos correladores más situados simétricamente antes y después del *prompt*, conocidos como *early* y *late*. El código *prompt* es el código C/A local, y se irá refinando hasta coincidir y estar alineado con el de la señal GPS entrante. El *early* y el *late* serán el mismo código *prompt* pero desplazados circularmente una muestra a izquierda y a derecha respectivamente, aproximadamente un cuarto de chip.

Una vez definidos los correladores generaremos las envolventes. Para ello, debemos hacer el producto, muestra a muestra (m.a.m), de la señal x_{DLL} junto con los tres correladores. Una vez generados los tres vectores, sumaremos todas sus muestras. Esto nos dará tres valores complejos, uno para cada código, que podemos normalizar dividiendo por el número de muestras que estamos comparando, 5456. El módulo del valor complejo p será la envolvente del código *prompt* y su fase corresponderá a la señal error de fase de la señal GPS (Fig. 5.23). Analizaremos en detalle este proceso más adelante.

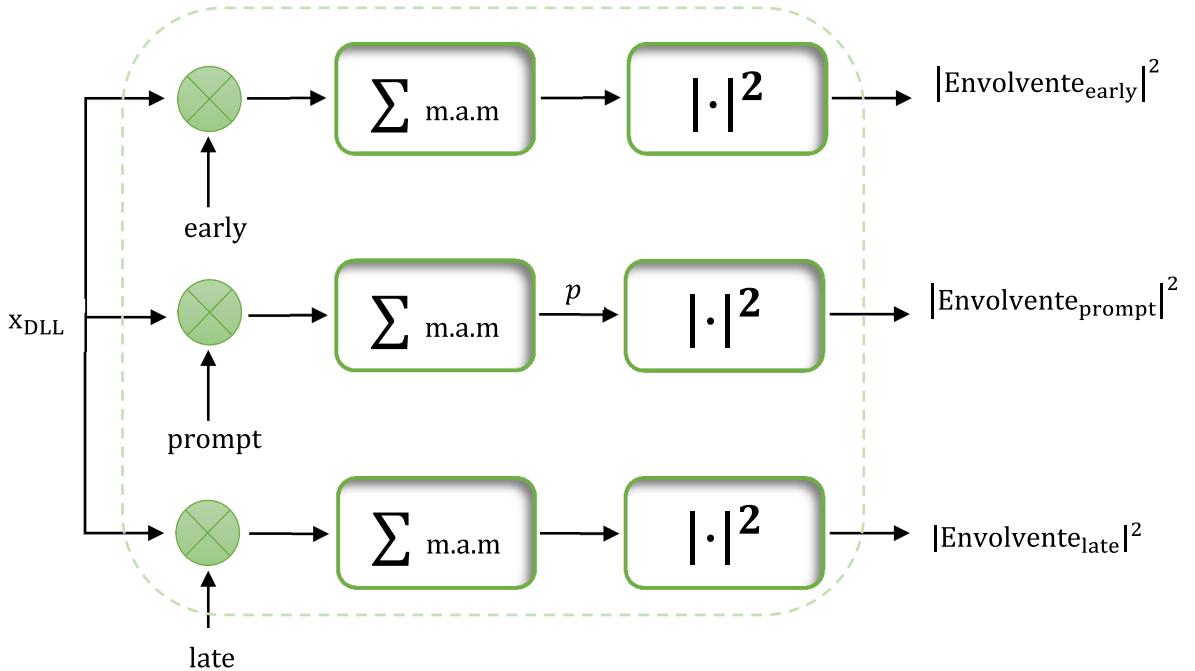


Fig. 5.23. Módulo correlador del DLL

5.4.1.3 Discriminador DLL

El discriminador es el encargado de generar la señal error que sirve como referencia para aplicar las correcciones necesarias y alinear así el código *prompt* con el código C/A de la señal GPS. Generaremos un discriminador coherente, que requiere una perfecta adquisición de la portadora, a partir de la resta del módulo al cuadrado de las envolventes *early* y *late*.

$$\text{Discriminador}_{\text{DLL}} = |\text{Envolvente}_{\text{early}}|^2 - |\text{Envolvente}_{\text{late}}|^2$$

5.4.1.4 Filtro de lazo DLL

La salida del discriminador (Fig. 5.22) ofrece una señal de referencia proporcional al error cometido entre la señal recibida y la réplica local que está siendo generada en el receptor. Esta señal de referencia es vital a la hora de implementar las correcciones necesarias para generar la réplica local con el mayor parecido posible a la que se está recibiendo, con el objetivo de que ambas señales estén sincronizadas.

Tanto la señal recibida como la salida del discriminador están afectadas por ruido térmico, y si conectamos directamente la salida del discriminador al NCO_{DLL}, este último absorbería toda la inestabilidad y variabilidad del ruido de entrada, enmascarando la corrección que realmente hay que aplicar en cada caso.

Es por ello que entre la salida del discriminador y el NCO_{DLL} se coloca un filtro de lazo, que reduce la variabilidad del ruido y proporciona una señal más suave que se entregará al NCO_{DLL} como elemento correctivo de retraso de código. El filtro de lazo utiliza factores de escala e integradores que se encargan de suavizar la señal de entrada, tal y como podemos observar en la siguiente figura (Fig. 5.24).

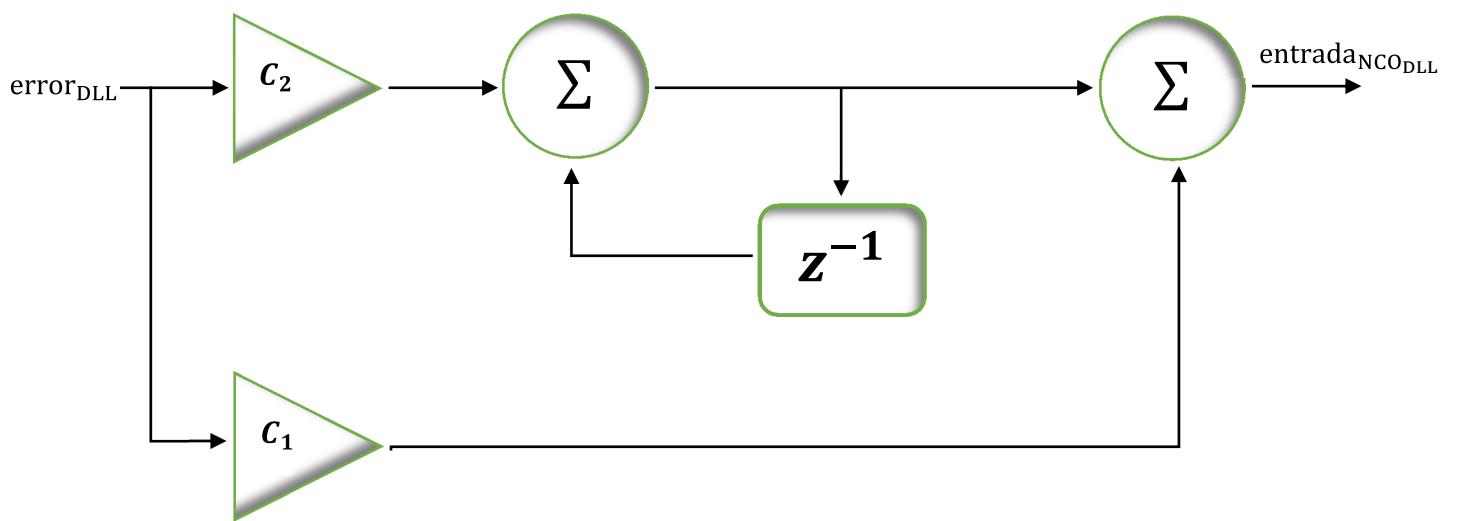


Fig. 5.24. Esquema de un filtro de lazo de orden 1

5.4.1.4.1 Filtro de lazo DLL de orden 0

El filtro de lazo de orden 0 se obtiene fijando la constante $C_2 = 0$ (Fig. 5.24). De esta manera, el esquema del filtro queda simplificado (Fig. 5.25):

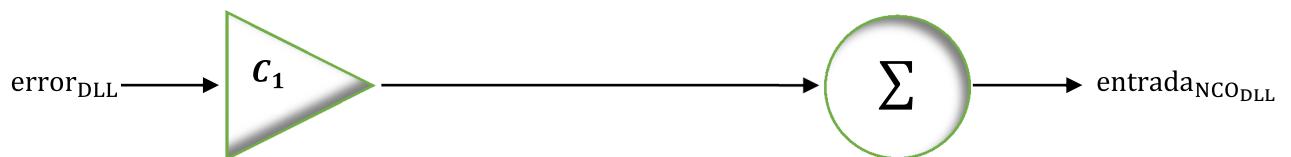


Fig. 5.25. Esquema de un filtro de lazo de orden 0

El esquema de un filtro de orden 0 se expresa mediante la siguiente relación matemática:

$$salida_{Filtro_0} = entrada_{NCO_{DLL}} = C_1 \cdot error_{DLL} = C_1 \cdot entrada_{Filtro_0}$$

Filtrar con un filtro de lazo de orden 0 equivale a aplicar únicamente un factor de escala a la señal de entrada, sin que se realice un filtrado propiamente dicho. Una vez se haya sincronizado la señal recibida con la réplica local, la salida del discriminador debe ser nula, por lo que la salida del filtro de lazo también debería anularse y el NCO_{DLL} se mantendría fijo, generando una señal constante. Por tanto, un esquema de tracking que incorpora un filtro de lazo de orden 0 únicamente es capaz de seguir señales constantes.

5.4.1.4.2 Filtro de lazo DLL de orden 1

El filtro de lazo de orden 1 corresponde al esquema de la Fig. 5.24. Este filtro de lazo incorpora un integrador que procesa la salida del discriminador escalada un factor C_2 . Una vez que la salida del discriminador se anula en régimen estacionario, la salida del filtro ya no se anula como en el caso anterior, sino que es capaz de ofrecer una salida constante gracias al valor acumulado en el registro retroalimentado del integrador. Esto hará mover de manera constante el NCO_{DLL} , con lo que la réplica podrá seguir señales cuya dinámica presente una derivada constante.

El esquema de un filtro de lazo de primer orden se expresa mediante la siguiente relación matemática:

$$salida_{Filtro_1}(i) = C_1 \cdot entrada_{Filtro_1}(i) + C_2 \cdot \sum_{k=1}^i entrada_{Filtro_1}(k)$$

Ampliaremos el estudio de los filtros de lazo cuando analicemos el PLL.

5.4.1.5 NCO_{DLL}

La salida del filtro de lazo anterior representa la corrección en fracción de chip que hay que añadir al desplazamiento del código *prompt* para que se alinee con el código C/A de la señal GPS. El desplazamiento en chips debe partir de un valor inicial y acumularse para generar el valor de desplazamiento en chips en cada iteración. La siguiente expresión ejemplifica el razonamiento anterior:

$$NCO_{DLL}(i + 1) = NCO_{DLL}(i) + \text{salida}_{\text{FiltroLazo}}(i)$$

5.4.1.6 Generador de secuencia

En el módulo de adquisición analizamos cómo remuestrear la señal réplica para que cada chip tuviera las muestras necesarias para ser comparado con el chip de la señal GPS. Ahora utilizaremos la misma idea del remuestreo, pero aplicando la corrección que nos ha devuelto la salida del NCO_{DLL}. Recordamos que los correladores *prompt*, *early* y *late* no son más que códigos C/A, más o menos alineados según el caso y, por tanto, también tienen 5456 muestras una vez remuestreados.

```
a=(0:5.333*1023-1)*(1/5.333)+NCO DLL(i);
b=floor(a);
prompt=CA(mod(b,1023)+1);
```

De esta manera, el generador de secuencia actualiza los valores de las muestras de los correladores utilizando la información del NCO_{DLL}. Explicaremos el proceso de remuestreo y generación de los nuevos códigos *prompt*, *early* y *late* con un estudio de casos según el desplazamiento en muestras representado por NCO_{DLL} para una iteración dada.

CASO 1

Si Discriminador_{DLL} = 0 → Envolvente_{late} = Envolvente_{early} → NCO_{DLL} = cte

Si el valor del discriminador es nulo, las muestras del código *prompt* deben ser las mismas en la siguiente iteración porque el retraso de código adquirido es el correcto y no se requiere ninguna variación. Como consecuencia, las muestras de los códigos *early* y *late* también se mantendrán.

CASO 2

Si $\text{Discriminador}_{\text{DLL}} < 0 \rightarrow \text{Envolvente}_{\text{late}} > \text{Envolvente}_{\text{early}} \rightarrow NCO_{\text{DLL}} \downarrow$

Un valor del discriminador negativo provocará que el valor del NCO_{DLL} decrezca y, por lo tanto, debemos efectuar un desplazamiento a la derecha en muestras según el valor del NCO_{DLL} para permitir a la $\text{Envolvente}_{\text{early}}$ recuperar fuerza en la siguiente iteración.

Cuando estudiamos el remuestreo en la adquisición obtuvimos el siguiente vector de muestras por chip:

$$a = [0 \ 0.1875 \ 0.3750 \ 0.5625 \ 0.75 \ 0.9375 \ 1.1250 \ \dots \ 1022.8125]$$

Si en la iteración anterior el valor del NCO_{DLL} era nulo, en la presente iteración será negativo debido al valor negativo del discriminador. Si, por ejemplo, en la presente iteración $NCO_{\text{DLL}} = -0.1$, el nuevo vector tomará los siguientes valores:

$$a = [-0.1 \ 0.0875 \ 0.2750 \ 0.4625 \ 0.65 \ 0.8375 \ 1.0250 \ \dots \ 1022.7125]$$

Este hecho provoca que cuando hagamos la parte entera del vector anterior obtengamos:

$$b = [-1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ \dots \ 1022]$$

Y cuando hagamos el módulo $1023 + 1$ muestra a muestra obtengamos:

$$\text{prompt}_{i+1} = [\text{prompt}_i(1023) \ \text{prompt}_i(1) \ \text{prompt}_i(1) \ \dots \ \text{prompt}_i(1023)]$$

El nuevo código prompt de la iteración $i + 1$ será el código prompt de la iteración i remuestreado con el valor de retardo en fracción de chip proporcionado por NCO_{DLL} . Esto hará que los códigos *early* y *late* también se desplacen una muestra a la derecha, y este hecho provocará que en la siguiente iteración del DLL la $\text{Envolvente}_{\text{early}}$ aumente y la $\text{Envolvente}_{\text{late}}$ disminuya.

CASO 3

Si $Discriminador_{DLL} > 0 \rightarrow Envolvente_{late} < Envolvente_{early} \rightarrow NCO_{DLL} \uparrow$

Haciendo un razonamiento similar al del caso anterior, llegamos a la conclusión que el código *prompt* de la iteración $i + 1$ tiene la siguiente distribución de muestras:

$$prompt_{i+1} = [prompt_i(1) \ prompt_i(1) \ ... \ prompt_i(1023) \ prompt_i(1)]$$

El nuevo código *prompt* de la iteración $i + 1$ será el código *prompt* de la iteración i remuestreado con el valor de retardo en fracción de chip proporcionado por NCO_{DLL} . Esto hará que los códigos *early* y *late* también se desplacen una muestra a la izquierda, y este hecho provocará que en la siguiente iteración del DLL la *Envolvente_{late}* aumente y la *Envolvente_{early}* disminuya. Si todo funciona correctamente, este desplazamiento en muestras a la izquierda provocará que el código *prompt* esté mejor alineado con el código C/A de la señal GPS.

En el siguiente esquema (Fig. 5.26) podemos observar gráficamente este efecto acordeón de las envolventes en dos iteraciones consecutivas obteniendo una alineación perfecta proviniendo del caso 3:

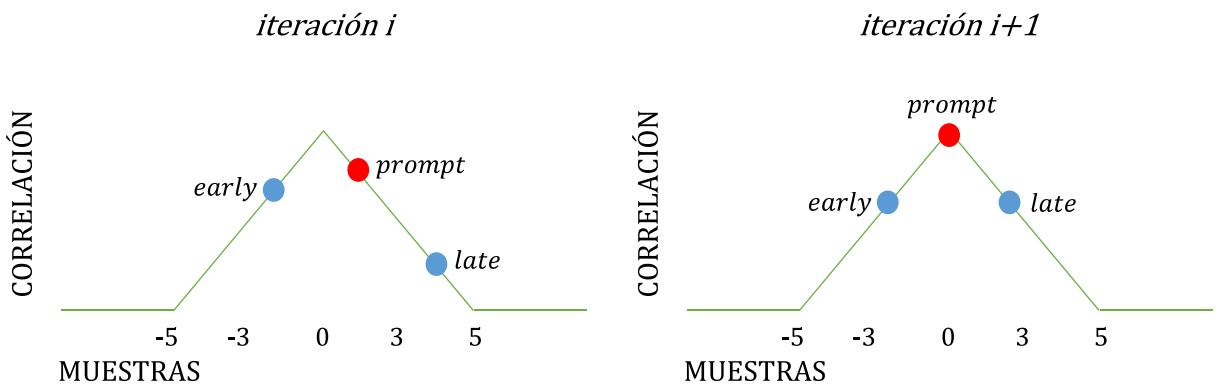


Fig. 5.26. Efecto en las envolventes del remuestreo correcto del código

5.4.1.7 DLL. Ejemplo de funcionamiento

A continuación presentaremos un ejemplo completo de funcionamiento de la arquitectura DLL. Después de generar la señal x_{DLL} , analizaremos el módulo correlador. Si el DLL consigue realizar un seguimiento perfecto de retraso de código, obtendremos el siguiente valor de $Envolvente_{prompt}$ para una iteración dada:

$$Envolvente_{prompt} = +1$$

A continuación demostraremos esta relación siguiendo paso a paso el esquema de la arquitectura DLL explicada anteriormente. Recordemos que el código *prompt* está formado por 5456 muestras de valores +1 y -1. Si obtenemos el valor del retraso de código de manera perfecta, el código *prompt* estará alineado con la señal GPS que entra al DLL, formada por su código C/A (igual al *prompt*) y el mensaje de navegación, que no hace otra cosa que multiplicar 20 periodos consecutivos del código C/A por +1 o -1. Asumiendo que la fase de la señal GPS es nula, tenemos:

$$Señal\ GPS = \pm prompt$$

Para simplificar la explicación consideraremos que las dos señales anteriores están formadas por 5 muestras:

$$prompt = [1 \ -1 \ -1 \ 1 \ -1]$$

Ahora estudiaremos los dos casos posibles:

CASO 1

Señal GPS = + prompt

Entonces:

$$Señal\ GPS = [1 \ -1 \ -1 \ 1 \ -1]$$

La suma normalizada del producto de ambas señales muestra a muestra es la siguiente:

$$\text{Suma}_{\text{normalizada}} = \frac{\text{Suma}_{\text{producto m.am}}}{\# \text{ muestras señal}} = \frac{1 + 1 + 1 + 1 + 1}{5} = 1$$

Calculando el módulo del anterior valor:

$$|p| = \text{Envolvente}_{\text{prompt}} = |\text{Suma}_{\text{normalizada}}| = 1$$

CASO 2

Señal GPS = - prompt

Entonces:

$$\text{Señal GPS} = [-1 + 1 + 1 - 1 + 1]$$

La suma normalizada del producto de ambas señales muestra a muestra es la siguiente:

$$\text{Suma}_{\text{normalizada}} = \frac{\text{Suma}_{\text{producto m.am}}}{\# \text{ muestras señal}} = \frac{-1 - 1 - 1 - 1 - 1}{5} = -1$$

Calculando el módulo del anterior valor:

$$|p| = \text{Envolvente}_{\text{prompt}} = |\text{Suma}_{\text{normalizada}}| = 1$$

En la siguiente figura (Fig. 5.27) vamos a comprobar cómo, efectivamente, el DLL tiene el comportamiento esperado:

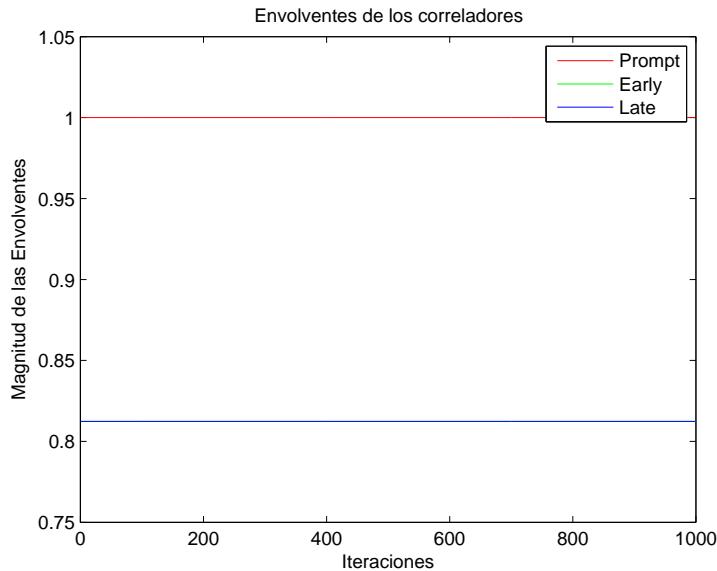


Fig. 5.27. Magnitud de las envolventes con un seguimiento perfecto de retraso de código

A continuación vamos a explicar por qué, si tenemos un seguimiento perfecto de retraso de código, la magnitud de las envolventes *early* y *late* son las siguientes (Fig. 5.27):

$$\text{Envolvente}_{\text{early}} = \text{Envolvente}_{\text{late}} = 0.8125$$

Como hemos comentado en capítulos anteriores, los códigos pseudoaleatorios están prácticamente incorrelados con el resto y con ellos mismos, excepto para un retardo nulo. Cuando el código *prompt* está perfectamente alineado con la señal GPS, el DLL genera una envolvente igual a la unidad. Si el código *prompt* estuviera desplazado un chip a izquierda o derecha, la correlación con la señal GPS debería ser prácticamente nula.

Los códigos *early* y *late* están desplazados una muestra a izquierda y derecha del código *prompt* en cada iteración. Ya hemos visto que un chip está representado por 5.3 muestras. Por tanto, una muestra representa 0.1875 chips. La siguiente figura presenta un esquema que relaciona las envolventes con el retraso en muestras y chips de los diferentes códigos mencionados (Fig. 5.28).

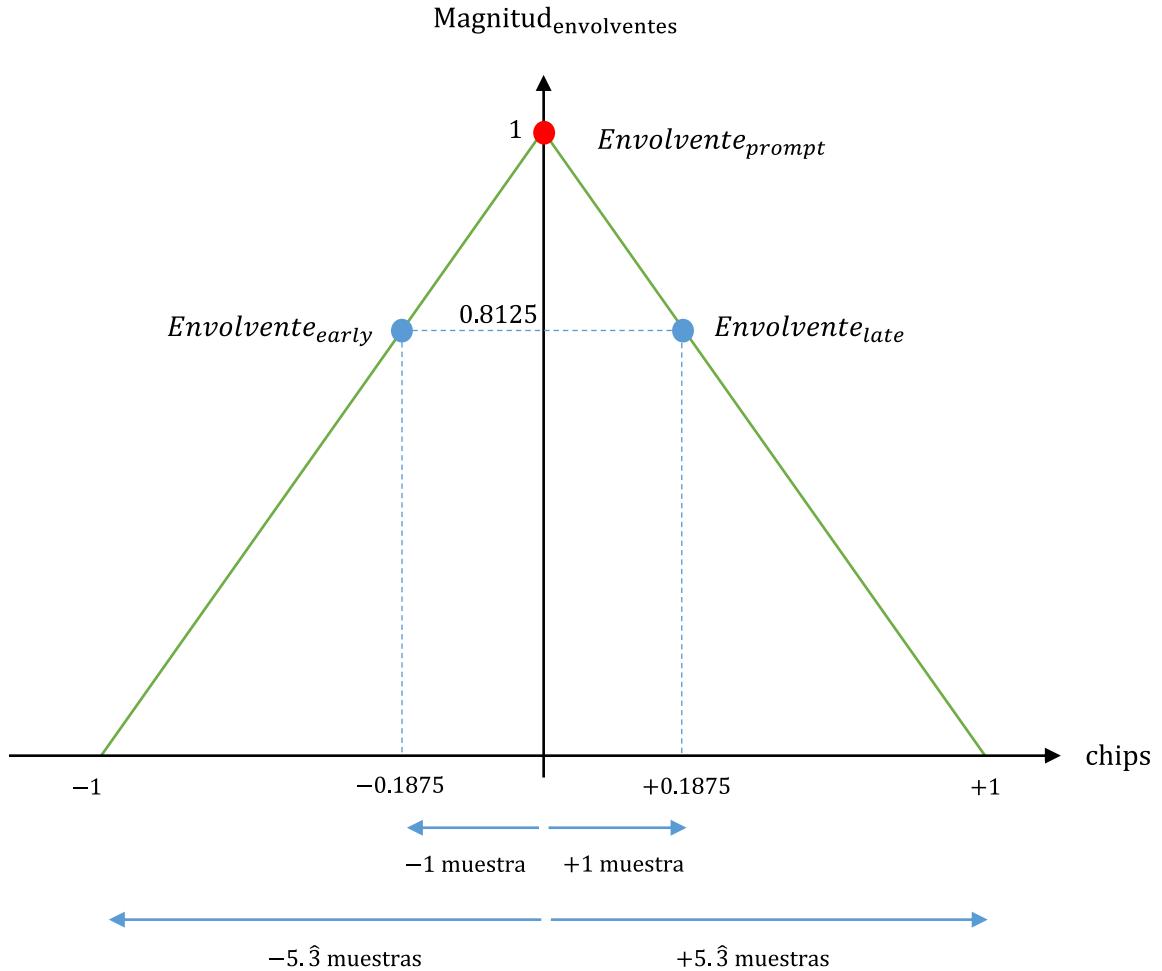


Fig. 5.28. Magnitud de las envolventes en función del retraso en chips

De esta manera, podemos calcular los valores de las envolventes *early* y *late* utilizando la ecuación de cada una de las rectas que forman el triángulo. Cada una de ellas de pendiente unidad, positiva y negativa respectivamente.

$$y = x + 1 \rightarrow x = -0.1875 \rightarrow y = 0.8125 \rightarrow \text{Envolvente}_{\text{early}} = 0.8125$$

$$y = -x + 1 \rightarrow x = +0.1875 \rightarrow y = 0.8125 \rightarrow \text{Envolvente}_{\text{late}} = 0.8125$$

Otra manera de enfocarlo es utilizando el número de muestras en que ambos códigos se encuentran desplazados respecto al código *prompt*:

$$\text{Envolvente}_{\text{early}} = \text{Envolvente}_{\text{late}} = 1 - \#\text{muestras}_{\text{desplazadas}} \cdot 0.1875$$

$$\text{Envolvente}_{\text{early}} = \text{Envolvente}_{\text{late}} = 1 - 1 \cdot 0.1875 = 0.8125$$

Aprovechando el razonamiento anterior, y observando la Fig. 5.28, podemos inferir que si generamos códigos *early* y *late* separados más de una muestra del código *prompt*, cada vez la magnitud de sus envolventes será más baja, hasta anularse cuando estén desplazados 5.3 muestras respecto al código *prompt*. En la siguiente figura (Fig. 5.29) observamos cómo se cumple este hecho. Generamos 4 códigos *early* y *late* adicionales, separados 2, 3, 4 y 5 muestras respectivamente del código *prompt*, y observamos cómo sus envolventes decrecen siguiendo la fórmula mostrada anteriormente:

$$\text{Envolvente} = 1 - \# \text{muestras}_{\text{desplazadas}} \cdot 0.1875$$

$$\text{Envolvente}_{\text{prompt}} = 1$$

$$\text{Envolvente}_{\text{early}} = \text{Envolvente}_{\text{late}} = 1 - 1 \cdot 0.1875 = 0.8125$$

$$\text{Envolvente}_{\text{early2}} = \text{Envolvente}_{\text{late2}} = 1 - 2 \cdot 0.1875 = 0.625$$

$$\text{Envolvente}_{\text{early3}} = \text{Envolvente}_{\text{late3}} = 1 - 3 \cdot 0.1875 = 0.4375$$

$$\text{Envolvente}_{\text{early4}} = \text{Envolvente}_{\text{late4}} = 1 - 4 \cdot 0.1875 = 0.25$$

$$\text{Envolvente}_{\text{early5}} = \text{Envolvente}_{\text{late5}} = 1 - 5 \cdot 0.1875 = 0.0625$$

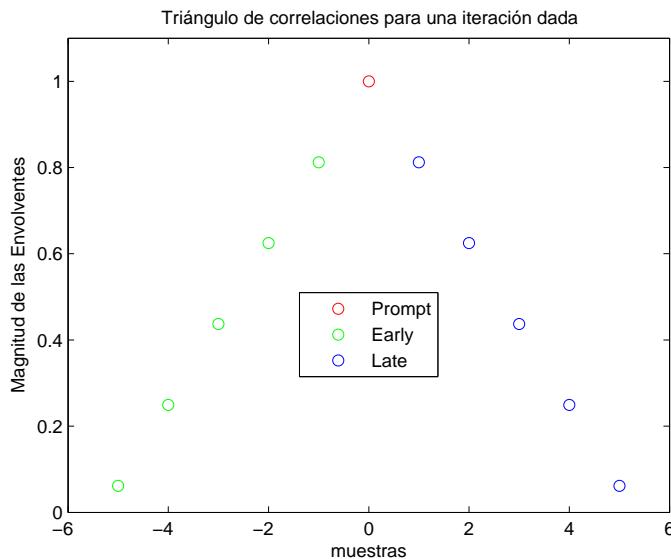
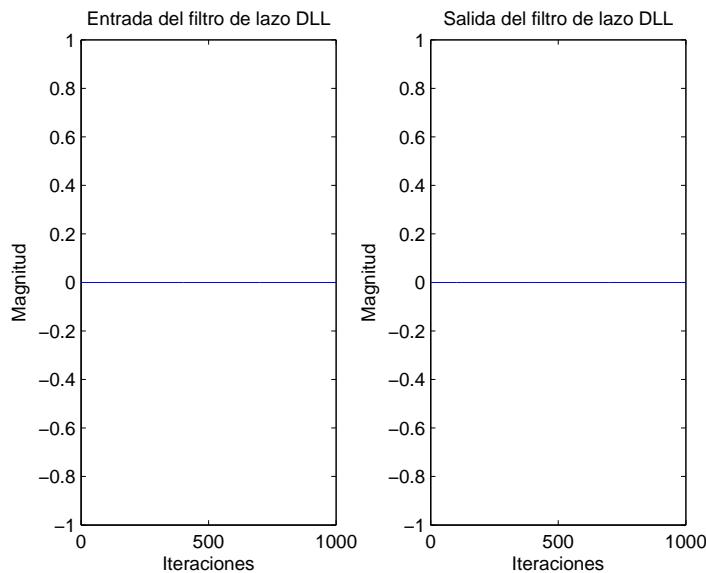


Fig. 5.29. Triángulo de correlaciones de las envolventes

Una vez generada la señal x_{DLL} y analizado el módulo correlador, el siguiente módulo de la arquitectura DLL es el discriminador. Si el seguimiento es perfecto, las envolventes de los códigos *early* y *late* serán iguales (Fig. 5.27), el código *prompt* estará perfectamente alineado con la señal GPS, y esto provocará que la salida del discriminador sea nula (imagen izquierda Fig. 5.30). Por tanto, si la salida del

discriminador es nula y utilizamos un filtro de lazo de orden 0, la salida del filtro de lazo también se anulará (imagen derecha Fig. 5.30).



Para finalizar el ejemplo de funcionamiento de la arquitectura DLL, únicamente apuntar que si la salida del filtro de lazo es nula, el desplazamiento en fracción de chip NCO_{DLL} para el remuestreo del código *prompt* se mantendrá constante, ya que el código se ha alineado perfectamente, y el código *prompt* de la siguiente iteración será el mismo que el de la actual. A continuación, analizaremos la arquitectura que corre en paralelo al DLL para realizar el seguimiento del desplazamiento Doppler, el PLL.

5.4.2 Phase Lock Loop. PLL

Sería muy complicado explicar el PLL sin antes haber analizado el DLL y sus fundamentos en cuanto al remuestreo, el NCO y los correladores. Es por ello que presentamos la arquitectura completa del módulo de tracking (Fig. 5.31), donde se puede observar claramente la retroalimentación entre el DLL y el PLL.

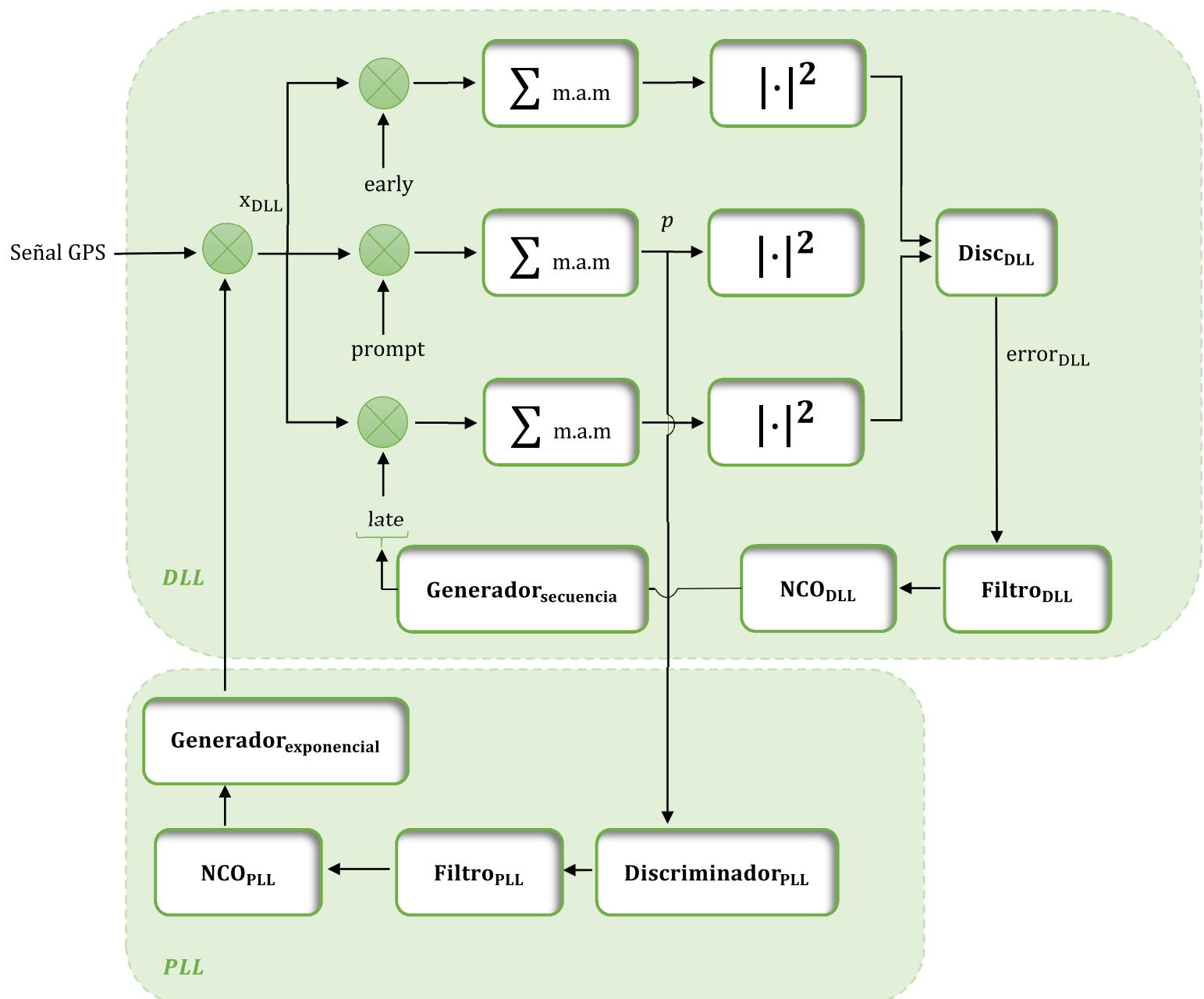


Fig. 5.31. Arquitectura completa del módulo de tracking

El objetivo de la arquitectura PLL es ajustar la frecuencia de un oscilador local, NCO_{PLL} , para igualarla a la fase de la señal GPS φ , y así extraer su desviación Doppler. Para ello, el proceso que sigue el PLL es el siguiente: el valor complejo p , cuyo módulo es la envolvente del código *prompt* y su fase es la señal error φ_{error} , entra al bloque discriminador. Este ofrece una señal φ_{error} proporcional al error del parámetro que se está siguiendo φ . Esta señal error será suavizada por el filtro de lazo, que finalmente entregará el valor del error de fase que se debe corregir a un generador de exponencial con la finalidad de extraer el desplazamiento Doppler de la señal GPS (Fig. 5.31). A continuación, detallaremos cada uno de los bloques que forma el PLL, para finalizar con un ejemplo de su funcionamiento.

5.4.2.1 Discriminador PLL

El seguimiento de la desviación Doppler se realiza, tal y como podemos observar en la Fig. 5.31, tomando como referencia el valor complejo p . El discriminador PLL, al igual que su homónimo DLL, generará una señal error que proporcionará información acerca de la corrección necesaria que se debe aplicar sobre la fase del generador de exponencial. De esta manera, estimaremos el desplazamiento en frecuencia y la fase de la señal GPS.

En el presente proyecto utilizaremos el discriminador de fase tipo Costas, muy utilizado en la práctica porque posee un comportamiento robusto a los cambios de fase que presenta la señal debido a la modulación de los bits del mensaje de navegación. Existen diversos discriminadores de fase de tipo Costas, y nosotros utilizaremos el dado por la siguiente expresión:

$$\text{Costas} = \text{Re}[p] \cdot \text{Im}[p]$$

siendo $p = Ae^{j\varphi_{\text{error}}}$ el valor definido en la Fig. 5.31 para cada iteración del bucle de tracking.

Desarrollando la expresión:

$$\text{Costas} = \text{Re}[Ae^{j\varphi_{\text{error}}}] \cdot \text{Im}[Ae^{j\varphi_{\text{error}}}] = A^2 \sin(\varphi_{\text{error}}) \cos(\varphi_{\text{error}})$$

Recordando la fórmula trigonométrica del seno del ángulo doble:

$$\sin(2\varphi_{error}) = 2\sin(\varphi_{error})\cos(\varphi_{error})$$

Llegamos a la fórmula final del discriminador de fase tipo Costas:

$$Costas = \frac{A^2}{2} \sin(2\varphi_{error})$$

Si bien es cierto que este tipo de discriminador no presenta una relación lineal entre su salida y el error de fase φ_{error} , en régimen estacionario cuando $\varphi_{error} \rightarrow 0$, podemos aprovecharnos de la serie de Taylor del $\sin(x)$:

$$\underset{x \rightarrow 0}{\sin(x)} = x$$

De esta manera, el discriminador Costas tendrá la siguiente expresión:

$$Costas = \frac{A^2}{2} 2\varphi_{error} = A^2 \varphi_{error}$$

Existiendo así una relación lineal entre la salida del discriminador y el valor de interés φ . Una vez definido el discriminador, podemos hacer un análisis entrada-salida (Fig. 5.32):

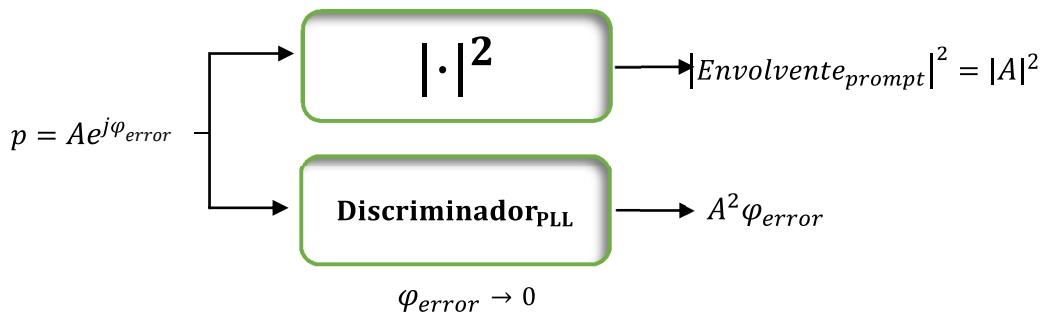


Fig. 5.32. Discriminador PLL de fase tipo Costas

5.4.2.2 Filtro de lazo PLL

El filtro de lazo PLL sigue la misma estructura que el del DLL. Por tanto, podéis encontrar toda la información en el subapartado 5.4.1.4 Filtro de lazo DLL. En el ejemplo de funcionamiento del PLL que analizaremos en un apartado posterior trabajaremos con filtros de lazo de orden 0 y 1.

5.4.2.3 NCO_{PLL}

La salida del filtro de lazo representa la corrección de fase en radianes que se debe aplicar al NCO_{PLL} para que este último genere la fase que se le entregará al generador de exponencial. Sigue la misma idea que el NCO_{DLL}, y su expresión es también la misma:

$$NCO_{PLL}(i + 1) = NCO_{PLL}(i) + \text{salida}_{\text{FiltroLazo}}(i)$$

5.4.2.4 Generador de exponencial

El último bloque de la arquitectura PLL es el generador de exponencial, el cual generará una exponencial incorporando la fase en radianes entregada por NCO_{PLL}. En el apartado 6.1.3 explicaremos en más detalle los parámetros de la fase de la señal GPS. Por tanto:

$$\text{Generador}_{\text{exponencial}} = \exp(-j2\pi \cdot \text{desviac}_{\text{doppler}_{ADQ}} \cdot \Delta t + \theta) * \exp(-jNCO_{PLL})$$

5.4.2.5 PLL. Ejemplo de funcionamiento

Empezaremos a analizar la arquitectura PLL por la entrada a su discriminador. Recuperando el esquema de la Fig. 5.32, para cada iteración, p es un número complejo diferente con una cierta amplitud A y fase φ_{error} .

En el ejemplo presentado $A = \text{Envolvente}_{prompt} = 1$ para todas las iteraciones porque el código se encuentra alineado permanentemente, y queremos comprobar que, efectivamente, a medida que el bucle de tracking avanza, la fase de p , φ_{error} , tiende a anularse, convirtiendo a p en un número real.

Este es el comportamiento que observamos en la Fig. 5.33, donde en la imagen izquierda observamos como la parte imaginaria de p se anula, originando una fase nula que nos advierte de un seguimiento perfecto de la fase de la señal GPS. Por otro lado, en la imagen derecha nos damos cuenta de que durante todas las iteraciones del bucle de tracking el módulo de p es constante y de valor unidad, por lo que el seguimiento de código es perfecto durante toda la simulación.

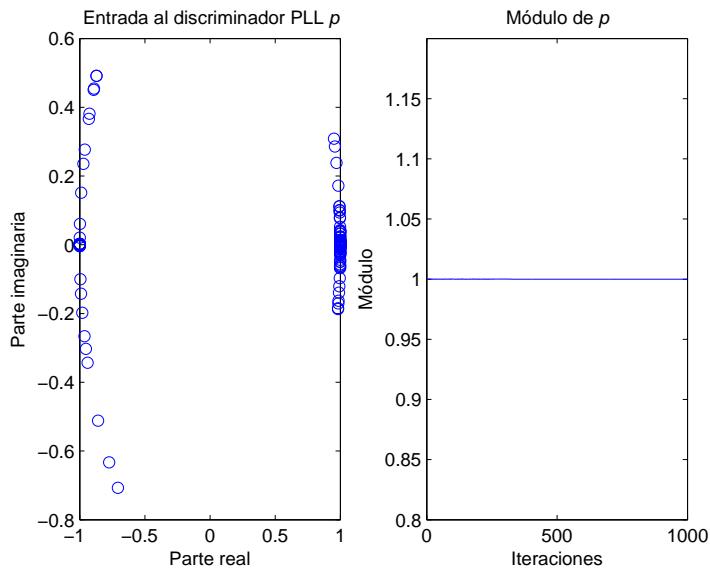


Fig. 5.33. Entrada al discriminador PLL

A la salida del discriminador (imagen izquierda de la Fig. 5.34), observamos como aproximadamente en 75 iteraciones alcanzamos el régimen estacionario ya que el error de fase $\varphi_{error} = 0$. En la imagen de la derecha de la misma figura, observamos como la salida del filtro de lazo de orden 0 únicamente escala la entrada proporcionada por la salida del discriminador, en este caso por un factor 1/10.

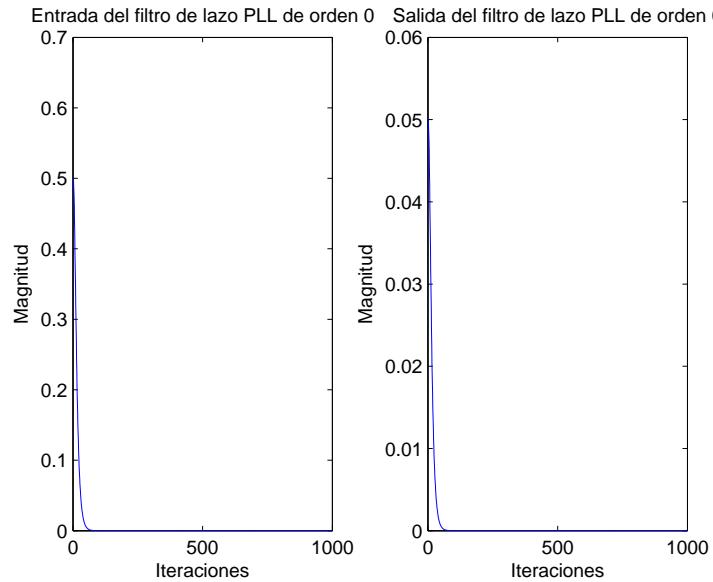


Fig. 5.34. Entrada y salida del filtro de lazo PLL de orden 0

Siguiendo el esquema del filtro de lazo de orden 0, lo que deberíamos esperar es que en aproximadamente 75 iteraciones el valor del NCO_{PLL} se estabilice, convergiendo a la fase de la señal GPS en banda base, especificada en 45° , ya que la desviación Doppler de este ejemplo es nula (Fig. 5.35).

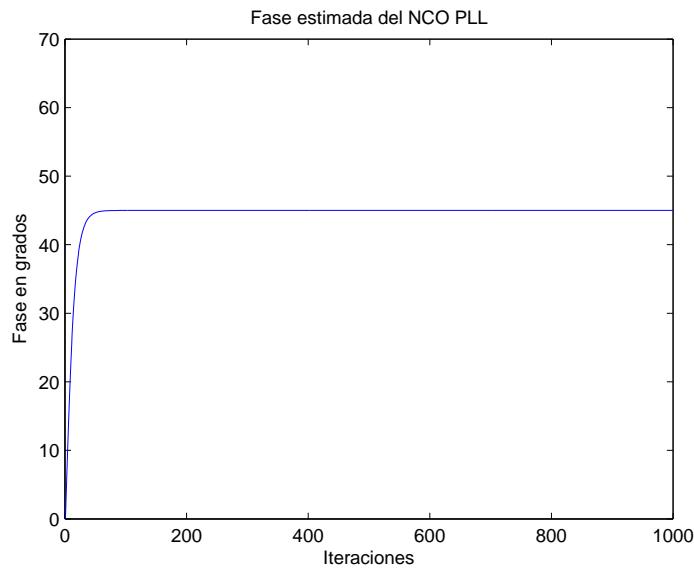


Fig. 5.35. Salida del NCO_{PLL} con la entrada de un filtro de lazo PLL de orden 0

Observamos cómo a partir de la iteración 75 la estimación del NCO_{PLL} converge al valor esperado, y el último bloque del módulo de tracking generará la exponencial con esta corrección de fase para finalmente estimar la fase de la señal GPS.

A continuación observaremos de qué manera converge el NCO_{PLL} si en vez de colocar un filtro de lazo de orden 0, colocamos uno de orden 1. Recordemos que, utilizando un filtro de lazo de orden 1, una vez que la salida del discriminador se anula en régimen estacionario, la salida del filtro ya no se anula como en el caso anterior, sino que es capaz de ofrecer una salida constante gracias al valor acumulado en el registro retroalimentado del integrador. Esto hará mover de manera constante el NCO_{PLL} con lo que la réplica podrá seguir señales cuya dinámica presente una derivada constante, como en el caso de la introducción de la desviación Doppler en la Fig. 5.36. En este caso, la entrada y la salida del filtro de lazo PLL de orden 1 vienen determinadas por la Fig. 5.36:

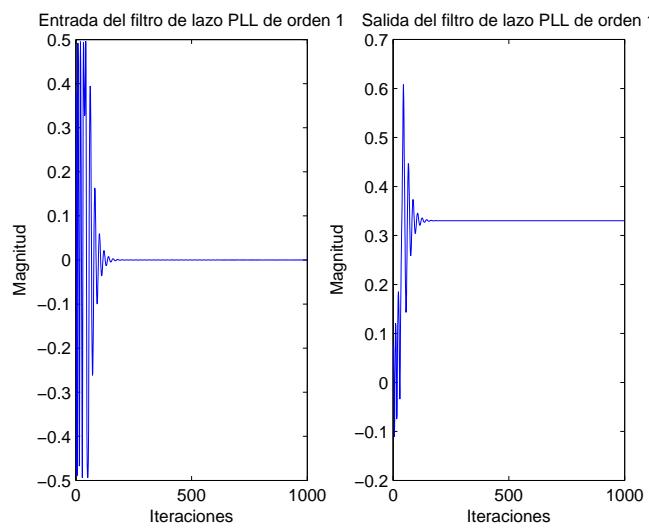


Fig. 5.36. Entrada y salida del filtro de lazo PLL de orden 1

La fase de la señal GPS y la desviación Doppler tienen una dependencia lineal como veremos en el siguiente capítulo. Por tanto, la estimación de fase del NCO_{PLL} tiene forma de recta (Fig. 5.37):

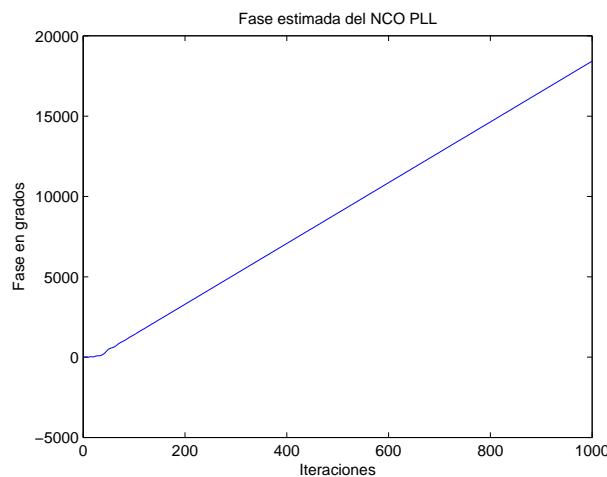


Fig. 5.37. Salida del NCO_{PLL} con la entrada de un filtro de lazo PLL de orden 1

Una vez que hemos explicado ambas arquitecturas y las señales que intervienen, resumiremos la idea sobre la que gira todo el tracking. La señal GPS está formada por el código C/A y el mensaje de navegación, y se encuentra modulada a un determinado desplazamiento Doppler que determina una cierta fase de la señal GPS. La arquitectura DLL persigue el retraso de código para que el código *prompt* se alinee con el código C/A de la señal GPS, a la vez que la arquitectura PLL demodula la señal GPS generando una exponencial cuya fase es la fase estimada por el PLL. De esta manera, como veremos en el siguiente capítulo, podremos estimar la distancia entre el receptor y el satélite, obtener el mensaje de navegación, y hacer otros cálculos de interés.

5.5 Módulo de navegación

El módulo de navegación es la última etapa del receptor GPS. A partir de la estimación de los valores de retraso de código y desviación Doppler, el receptor es capaz de obtener los pseudorrangos u observables de fase ($d_{kmedida}$) para cada uno de los k satélites visibles, que calcularemos en el siguiente capítulo, y con este valor calcular la posición del receptor. Para ello, como hemos comentado en la sección 3.3, en el posicionamiento en 3D necesitamos al menos 4 satélites para despejar la incógnita del error de reloj del receptor.

Para analizar visualmente el fenómeno del cálculo de la posición estudiaremos el caso en 2D por sencillez, en el que necesitaremos al menos 3 satélites visibles para calcular la posición del receptor (ver sección 3.3). La ampliación a 3D se haría de la misma manera con un cuarto satélite.

La clave del posicionamiento basado en GNSS es el hecho de que 3 circunferencias se intersectan en un único punto para un valor adecuado de error de reloj del receptor $\delta = c \cdot \text{error}_{\text{reloj usuario}}$. Por tanto, para un satélite k , la resta entre su pseudorango y δ proporcionará la distancia real entre satélite y receptor y, a su vez, uno de los tres radios necesarios para calcular la posición en 2D (Fig. 5.38):

$$d_k = d_{kmedida} - \delta$$

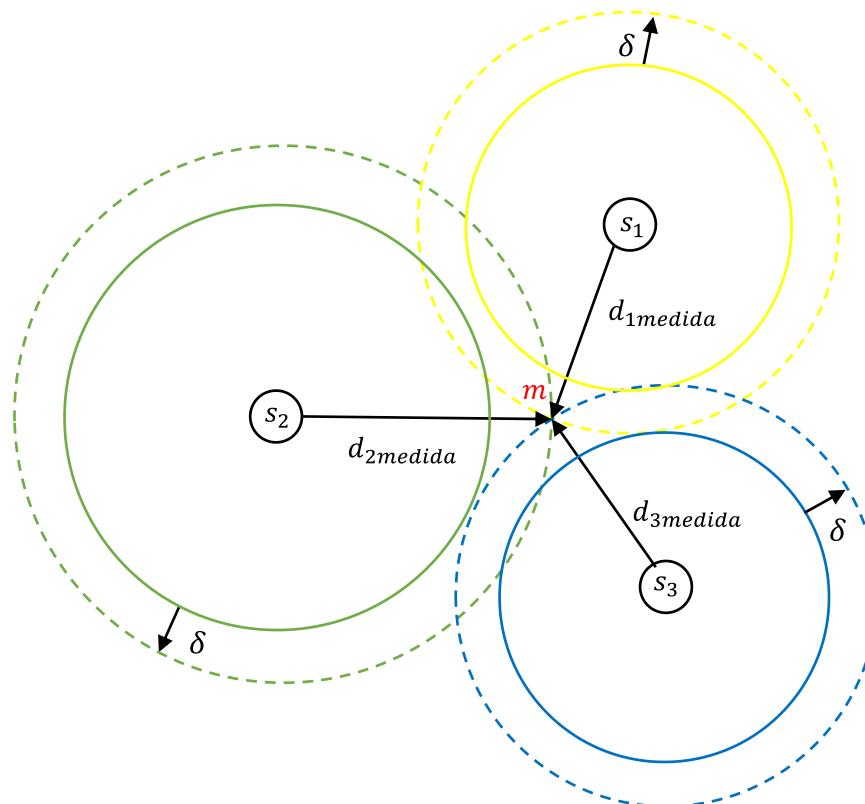


Fig. 5.38. Ilustración de la solución de navegación con error del receptor en 2D

Por tanto, recuperamos el sistema 4x4 de ecuaciones de navegación en 3D (ver sección 3.3) que nos permitirá calcular la posición del receptor:

$$d_1^2 = (x_1 - mx)^2 + (y_1 - my)^2 + (z_1 - mz)^2 = (d_{1medida} - \delta)^2$$

$$d_2^2 = (x_2 - mx)^2 + (y_2 - my)^2 + (z_2 - mz)^2 = (d_{2medida} - \delta)^2$$

$$d_3^2 = (x_3 - mx)^2 + (y_3 - my)^2 + (z_3 - mz)^2 = (d_{3medida} - \delta)^2$$

$$d_4^2 = (x_4 - mx)^2 + (y_4 - my)^2 + (z_4 - mz)^2 = (d_{4medida} - \delta)^2$$

De todas las variables existentes en las anteriores ecuaciones, las coordenadas de los satélites $(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), (x_4, y_4, z_4)$ son conocidas ya que el receptor recibe las efemérides en el mensaje de navegación, y es el propio receptor quien calcula los pseudorrangos $d_{1medida}, d_{2medida}, d_{3medida}, d_{4medida}$ que le separan de cada satélite. Por tanto, tenemos 4 ecuaciones con las 4 incógnitas: las coordenadas del receptor (mx, my, mz) y su error de reloj δ .

De esta manera, deberemos solucionar el siguiente sistema de ecuaciones:

$$d_{1medida} - \delta = \sqrt{(x_1 - mx)^2 + (y_1 - my)^2 + (z_1 - mz)^2}$$

$$d_{2medida} - \delta = \sqrt{(x_2 - mx)^2 + (y_2 - my)^2 + (z_2 - mz)^2}$$

$$d_{3medida} - \delta = \sqrt{(x_3 - mx)^2 + (y_3 - my)^2 + (z_3 - mz)^2}$$

$$d_{4medida} - \delta = \sqrt{(x_4 - mx)^2 + (y_4 - my)^2 + (z_4 - mz)^2}$$

Como podemos observar, no se trata de un sistema de ecuaciones lineales, por lo que deberemos usar técnicas que nos permitan resolverlo de una manera ágil. A modo introductorio, existen dos grandes familias de métodos para calcular la solución de navegación: los que linealizan la dependencia de la distancia con la posición, y los que calculan la posición de una forma cerrada sin necesidad de iterar.

El valor de δ será aquel que haga que la estimación de m esté lo más próximo posible a todas las circunferencias. En función de cómo definamos matemáticamente esta proximidad se obtendrán diferentes variaciones del método de posicionamiento.

Acabamos el capítulo 5 dedicado al análisis de los módulos del receptor. En el último capítulo nos centraremos en el cálculo de las distancias, e introduciremos efectos reales en la simulación del receptor para analizar su comportamiento.

6. Análisis del funcionamiento del receptor bajo efectos reales

El espacio es el medio que soporta el viaje de la señal GPS desde el satélite hasta el receptor. Para llegar a él, la onda electromagnética tiene que atravesar la atmósfera terrestre, por lo que la señal GPS debe atravesar todas sus capas. La ionosfera o la troposfera afectan de una manera muy determinada a la onda electromagnética y, aunque su estudio se escapa del objeto del proyecto, en las siguientes secciones observaremos parte de sus efectos en la señal GPS, tales como el efecto Doppler o el *multipath*. También analizaremos los efectos introducidos por el receptor, como el filtrado y el ruido, para finalmente hacer un estudio completo con todos los efectos integrados.

6.1 Efecto Doppler

El efecto Doppler es el aparente cambio de frecuencia que percibe un receptor debido a la variación de distancia que le separa de la fuente de información. En esta sección el satélite actuará de fuente, ya que es el sistema que envía la señal GPS, y el receptor GPS ubicado en el teléfono móvil hará las veces de receptor (Fig. 6.1):

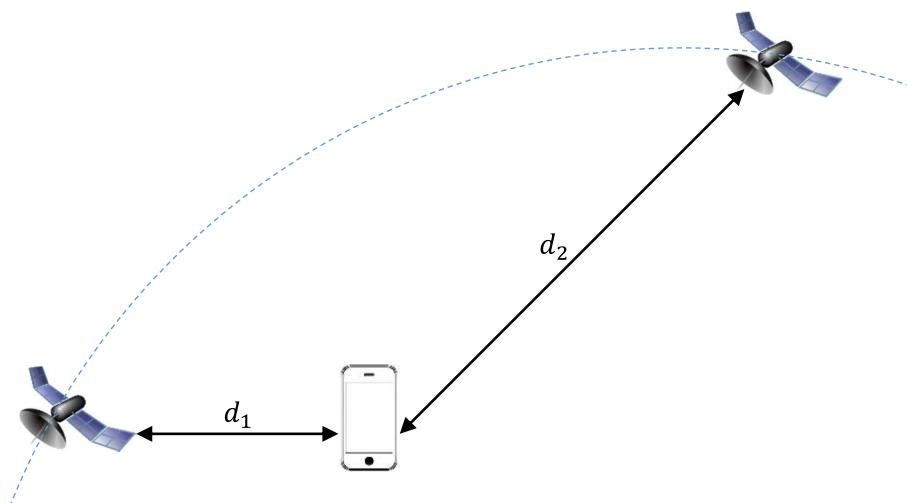


Fig. 6.1. Movimiento del satélite respecto al receptor

Consideraremos un receptor estático para facilitar la explicación. En la Fig. 6.1 observamos como a medida que el satélite sigue su órbita, su distancia respecto al receptor varía. En el momento de máxima cercanía, el satélite se encuentra a una distancia d_1 del receptor, y esta va en aumento hasta llegar a la distancia d_2 . A medida que aumenta la distancia entre satélite y receptor, la frecuencia de la señal GPS que percibe el receptor es más baja ya que la longitud de onda de la señal GPS aumenta. De esta manera, la frecuencia que percibe el receptor GPS cuando se encuentra a una distancia d_1 del satélite es más alta que cuando está separado d_2 (Fig. 6.2):

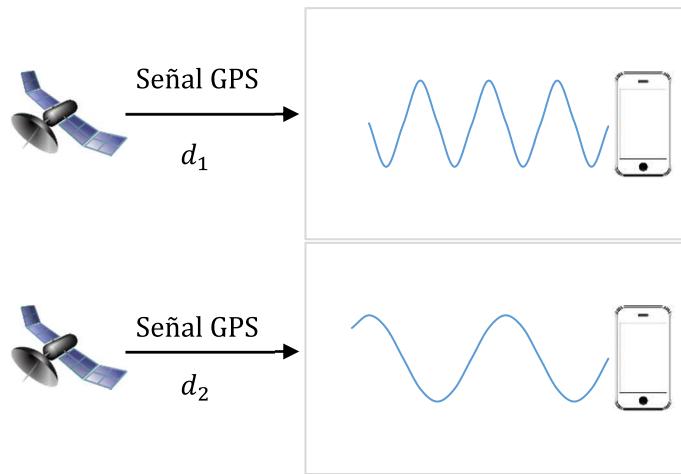


Fig. 6.2. Variación de la frecuencia percibida por el receptor

Después de haber explicado el fenómeno de manera intuitiva, generalizaremos matemáticamente el efecto Doppler. La frecuencia percibida por el receptor $f_{receptor}$ viene determinada por la siguiente ecuación:

$$f_{receptor} = f_{real} + f_{dop}$$

Donde f_{real} representa la frecuencia f_{L1} a la que el satélite envía la señal GPS y f_{dop} , la frecuencia Doppler. De esta manera:

$$f_{receptor} = f_{L1} + f_{L1} \frac{v}{c} = f_{L1} \left(1 + \frac{v}{c}\right)$$

Siendo v la velocidad relativa entre satélite y receptor, y c la velocidad de la luz.

Si satélite y receptor se acercan, la frecuencia percibida por el receptor aumentará:

$$v > 0 \rightarrow f_{dop} = f_{L1} \frac{v}{c} > 0 \rightarrow f_{receptor} \uparrow$$

Si satélite y receptor se alejan, la frecuencia percibida por el receptor disminuirá:

$$v < 0 \rightarrow f_{dop} = f_{L1} \frac{v}{c} < 0 \rightarrow f_{receptor} \downarrow$$

En los siguientes apartados del efecto Doppler analizaremos las consecuencias prácticas que tiene este efecto sobre la señal GPS y el receptor, y estudiaremos las diferentes maneras que tiene el receptor de estimar la distancia que le separa del satélite.

6.1.1 Frecuencia Doppler

Si bien es cierto que a lo largo del estudio hemos tenido en cuenta el desplazamiento Doppler que sufre la portadora de la señal GPS recibida, hemos obviado que este desplazamiento también lo sufren los otros dos componentes de la señal GPS: el código C/A asociado al satélite y el mensaje de navegación.

Es por ello que debemos tener en cuenta el término de desplazamiento Doppler en la frecuencia de chip del código C/A. En el siguiente razonamiento el subíndice 1 se utiliza para el valor con el que hemos trabajado hasta ahora, y el subíndice 2 para el nuevo valor que estamos introduciendo en el apartado. Hasta ahora, el valor de la frecuencia de chip era:

$$f_{chip_1} = \frac{1}{T_{chip}} = \frac{1}{0.977 \mu s} = 1.023 \text{ MHz}$$

Ahora tenemos que aplicarle el término corrector del desplazamiento Doppler f_{dop} :

$$f_{chip_2} = f_{chip_1} + f_{dop} = f_{chip_1} + f_{chip_1} \frac{v}{c} = f_{chip_1} \left(1 + \frac{v}{c} \right) = f_{chip_1} \left(1 + \frac{f_{dop}}{f_{L1}} \right)$$

Este resultado también variará el número de muestras que representa cada chip:

$$\#muestras_{chip_2} = \frac{f_m}{f_{chip_2}} = \frac{f_m}{f_{chip_1}} \left(1 + \frac{f_{dop}}{f_{L1}}\right)^{-1} = 5.3 \left(1 + \frac{f_{dop}}{f_{L1}}\right)^{-1}$$

Como podemos observar en la expresión anterior, a no ser que el desplazamiento Doppler sea nulo, el número de muestras por chip será diferente. Aunque $f_{dop} \ll f_{L1}$, el hecho de que exista un desplazamiento Doppler provoca que tengamos que replantear el código de tracking, como veremos a continuación.

Todo lo expuesto anteriormente repercutirá en el periodo del código C/A, que ya no será de 1 ms ni tendrá un número entero de muestras:

$$\#muestras_{periodo_2} = \#muestras_{chip_2} \cdot N_{chips} = 5.3 \left(1 + \frac{f_{dop}}{f_{L1}}\right)^{-1} \cdot 1023$$

Tomando como ejemplo el siguiente valor de desplazamiento Doppler, junto con el valor nominal de la frecuencia portadora:

$$f_{dop} = 1 \text{ KHz}$$

$$f_{L1} = 1575.42 \text{ MHz}$$

observamos que, como comentábamos, ya no tenemos un número entero de muestras en un periodo del código C/A:

$$\#muestras_{periodo_2} = \#muestras_{chip_2} \cdot N_{chips} = 5455.99653 \text{ muestras}$$

Con lo que, ahora, en el módulo de tracking no tenemos que sesgar cada iteración comparando 5456 muestras de ambas señales, sino que debemos integrar en cada iteración un periodo del código C/A, que a veces tendrá 5455 muestras y otras 5456. Por tanto, cada integración deberá acabar en la última muestra del periodo del código.

6.1.2 Estudio del sistema

La señal GPS ha sido generada con sus parámetros originales en el satélite pero, como hemos visto, se ve afectada por la frecuencia Doppler cuando llega al receptor. Es por ello que en los siguientes subapartados analizaremos las consecuencias que la desviación Doppler tiene sobre el estudio del sistema completo.

Los dos siguientes subapartados siguen el hilo argumental del proyecto, así que, para no repetir información, la explicación de cada concepto se remitirá al apartado en cuestión, centrándonos únicamente en los aspectos que sí cambian a causa del efecto Doppler.

6.1.2.1 Implementación de la señal GPS

Como hemos analizado en el apartado anterior, cada chip de la señal GPS muestreada a la salida del *front-end* está formada por $\#muestras_{chip_2}$ muestras. Por tanto, deberemos generar un código C/A ideal de 1023 chips y 1023 muestras (*cacode.m*) y, tal y como se explica en el apartado 5.3.1, remuestrear la señal GPS para simular que cada chip está formado por más de una muestra, con el añadido del uso de la nueva f_{chip_2} .

Retomando el apartado 5.3.1, generaremos cada periodo del código C/A tomando las muestras que estén comprendidas entre el chip 0 y el 1022, entre el 1023 y el 2045, y así sucesivamente. Cada 20 periodos del código C/A del satélite serán multiplicados por +1 o -1, según si el bit de navegación es positivo o negativo.

Generaremos un valor de retraso de código y de fase conocidos, para posteriormente analizar si el receptor está adquiriendo bien los valores y les hace el seguimiento esperado en el módulo de tracking. Una vez generada la señal GPS en banda base, la modularemos a una determinada frecuencia residual, que actuará de desviación Doppler. Hasta este punto hemos generado la señal GPS a la salida del *front-end*, y está lista para ser procesada por el receptor (*main_fchip.m*).

6.1.2.2 Implementación del receptor

Una vez la señal GPS entra en el módulo de adquisición, el código C/A de la réplica local también se verá afectado por la desviación Doppler y devolverá unos valores de desviación Doppler y retraso de código determinados, que serán enviados al módulo de tracking.

Como hemos comentado anteriormente, la desviación Doppler provoca que un periodo del código C/A ya no tenga un número entero de muestras, ni una duración exacta de 1 ms. Utilizando los valores que hemos comentado durante el proyecto, algunos periodos del código C/A tendrán 5455 muestras, y otros 5456. Por tanto, el objetivo del módulo de tracking es integrar un periodo de la señal GPS en cada iteración.

Para conseguirlo, hemos utilizado el siguiente razonamiento. Ahora no sabemos si un periodo del código C/A tiene 5455 o 5456 muestras, pero no habrá mucha más variación atendiendo a la desviación Doppler máxima de ± 10 KHz. Por esta razón, generamos un código *prompt* de 5459 muestras, y así nos aseguramos que el módulo de tracking cubre todas las posibilidades de periodo de señal GPS entrante.

En cada iteración del bucle de tracking integraremos un periodo de la señal GPS con un periodo del código *prompt*. Para asegurarnos de tener cada vez un periodo completo, obtendremos desde la primera muestra del primer chip, hasta la última del último, con un razonamiento similar al del apartado 5.3.1 (*track_fchip2.m*).

Una vez hayamos seleccionado correctamente la duración de cada periodo para una iteración, se ejecutarán todas las operaciones del módulo de tracking vistas en la sección 5.4. Posteriormente, se volverá a seleccionar la duración de la señal GPS y volverá a correr el tracking (*bucle_tracking2.m*), repitiendo el proceso tantas veces como tengamos programado el algoritmo, y así realizar el seguimiento de los valores de interés.

6.1.3 Análisis de la fase de la señal GPS

En este apartado analizaremos porque durante ciertos pasajes del proyecto hemos utilizado indistintamente los términos fase y frecuencia Doppler para referirnos al objetivo del PLL, que no es otro que el de estimar la fase de la señal GPS. La respuesta reside en que ambos parámetros, desviación Doppler y fase, están relacionados de manera lineal para generar la fase de la señal GPS. Por tanto, cuando la señal GPS está modulada a una frecuencia Doppler con una cierta fase, la combinación lineal de ambos parámetros genera la fase de la señal GPS, que el PLL debe estimar para demodularla.

A continuación vamos a analizar en detalle la explicación anterior. La señal GPS está modulada a una cierta frecuencia Doppler antes de entrar al bloque de adquisición:

$$\text{Señal GPS} = \text{Señal GPS}_{BB} * \exp(j \cdot 2\pi \cdot \text{desviac}_{doppler} \cdot \Delta t)$$

Si descomponemos la señal GPS en banda base en módulo y fase:

$$\text{Señal GPS} = |\text{Señal GPS}_{BB}| * \exp(j \cdot (2\pi \cdot \text{desviac}_{doppler} \cdot \Delta t + \theta))$$

De esta manera la fase de la señal GPS es la siguiente:

$$\varphi = \text{fase}_{\text{señal GPS}} = 2\pi \cdot \text{desviac}_{doppler} \cdot \Delta t + \theta$$

Y observamos su dependencia lineal con la frecuencia Doppler. De esta manera, se puede calcular la fase de la señal GPS a partir de su frecuencia Doppler $\text{desviac}_{doppler}$, de dos instantes de tiempo dados Δt y de la fase de la señal GPS en banda base θ . La expresión anterior representa la ecuación de una recta, y el término independiente θ solo generará un offset en la estimación de la fase, que no afectará demasiado en el efecto que debemos observar: la variación lineal de la estimación de la fase con la frecuencia Doppler.

Por tanto, si el PLL funciona correctamente, el valor que genera el NCO_{PLL} deberá ser igual a la fase de la señal GPS.

$$\text{NCO}_{PLL} = \varphi \text{ [rad]}$$

Y a partir de la expresión anterior, calcular la desviación Doppler:

$$\text{desviac}_{doppler} = \frac{1}{2\pi} \cdot \frac{\text{NCO}_{PLL}}{\Delta t}$$

En el caso particular de que la desviación Doppler sea nula, el valor NCO_{PLL} debe converger a la fase de la señal GPS en banda base:

$$\text{NCO}_{PLL} = \text{fase}_{\text{señal GPS}} = \theta$$

6.1.4 Cálculo del pseudorrango con la arquitectura DLL

Para una iteración dada, la salida del filtro de lazo DLL representa la corrección en fracción de chip que hay que añadir al desplazamiento del código *prompt* para que alinee con el código C/A de la señal GPS. La acumulación de las fracciones de chip, NCO_{DLL} , nos da una idea del movimiento entre satélite y receptor en unidades de chip.

Pero un valor de chip no es significativo: debemos transformarlo en metros. Para ello, el primer paso es dividir este valor por la frecuencia de chip y así obtener una estimación de tiempo. Este tiempo será el tiempo de propagación junto con el error de reloj del receptor, definido en la sección 3.3:

$$tiempo = \frac{NCO_{DLL}}{f_{chip_2}}$$

Si realizamos un análisis de unidades de la anterior ecuación:

$$[s] = \frac{[\text{chips}]}{\left[\frac{\text{chips}}{\text{s}} \right]}$$

Observamos cómo la anterior ecuación representa unidades de tiempo y únicamente hemos realizado una conversión de chips a segundos. Solo quedará multiplicar por la velocidad de la luz para obtener la pseudodistancia:

$$pseudorrango = c \cdot tiempo$$

A continuación generaremos tres simulaciones para ilustrar lo explicado en este apartado. Para cada una de ellas, generaremos la gráfica del pseudorrango y la compararemos con la variación de distancia real. A partir de la información de ambas gráficas extraeremos la velocidad relativa entre satélite y receptor, la frecuencia Doppler y analizaremos si los resultados experimentales coinciden con los teóricos.

6.1.4.1 Dinámica de un tramo

En esta primera simulación, el satélite se acercará al receptor a una velocidad constante de 10 m/s. En el apartado anterior hemos llegado a la siguiente relación:

$$pseudorrango = c \cdot \frac{NCO_{DLL}}{f_{chip_2}}$$

Por tanto, representaremos el pseudorrango a partir de la información que devuelve el NCO_{DLL} (imagen derecha de la Fig. 6.3), y lo compararemos con la variación de distancia real que experimentan satélite y receptor (imagen izquierda de la Fig. 6.3):

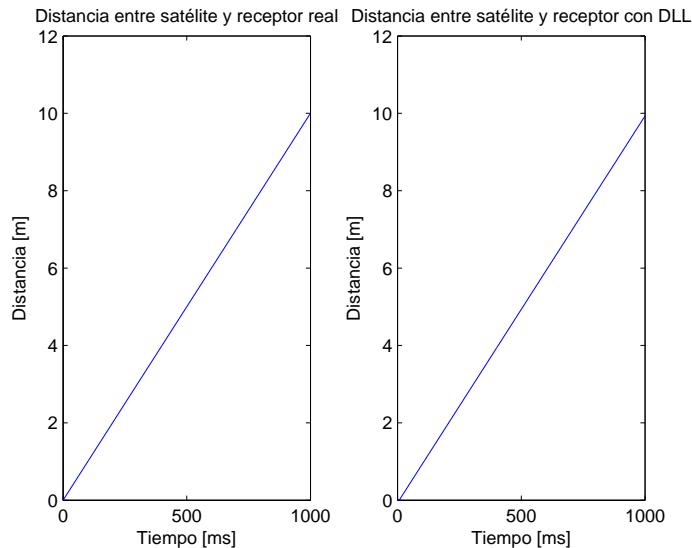


Fig. 6.3. Variación real de distancia (izquierda), variación con el DLL (derecha)

Observando la gráfica de la derecha de la figura anterior (Fig. 6.3), podemos calcular la velocidad relativa entre satélite y receptor dividiendo el pseudorrango por el tiempo:

$$v_{ESTIMADA} = \frac{\Delta pseudorrango}{\Delta tiempo} = \frac{9.931 - 0}{1 - 0} = 9.931 \text{ m/s}$$

Y con la estimación de la velocidad podremos extraer la frecuencia Doppler que experimenta la señal GPS:

$$f_{dopESTIMADA} = f_{L1} \frac{v_{ESTIMADA}}{c} = 1575.42 \cdot 10^6 \frac{9.931}{299792458} = 52.18 \text{ Hz}$$

Finalmente, calcularemos los parámetros reales de la simulación:

$$v_{REAL} = 10 \text{ m/s}$$

$$f_{dopREAL} = f_{L1} \frac{v_{REAL}}{c} = 1575.42 \cdot 10^6 \frac{10}{299792458} = 52.55 \text{ Hz}$$

La siguiente tabla resume los valores de frecuencia Doppler encontrados (Tabla 6.1):

f_{dop} [Hz]	Real	Estimada
Tramo Ascendente	52.55	52.18

Tabla 6.1 Comparación de f_{dop} : real y estimada

Por tanto, atendiendo a los valores de la Tabla 6.1, concluimos que el DLL ha hecho una estimación razonable del código, y lo ha mantenido alineado proporcionando una variación de distancia próxima a la real.

6.1.4.2 Dinámica de dos tramos

En la segunda simulación el satélite se acercará a una velocidad constante de 20 m/s al receptor y, a mitad de simulación, se alejará a la misma velocidad. Por tanto, representaremos el pseudorrango (imagen derecha de la Fig. 6.4), y lo compararemos con la variación de distancia real que experimentan satélite y receptor (imagen izquierda de la Fig. 6.4):

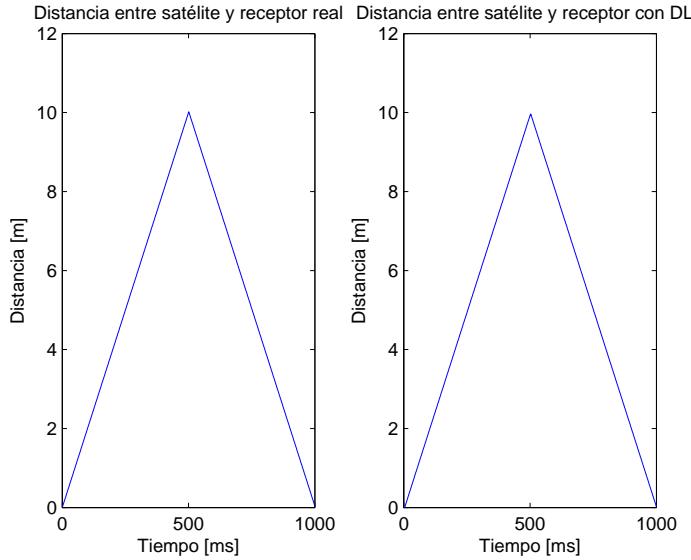


Fig. 6.4. Variación real de distancia (izquierda), variación con el DLL (derecha)

Observando la gráfica de la derecha de la figura anterior (Fig. 6.4), podemos calcular la velocidad relativa entre satélite y receptor dividiendo el pseudorrango por el tiempo:

$$v_{ESTIMADA\ ASCENDENTE} = \frac{\Delta pseudorrango}{\Delta tiempo} = \frac{9.925 - 0}{0.5 - 0} = 19.850 \text{ m/s}$$

$$v_{ESTIMADA\ DESCENDENTE} = \frac{\Delta pseudorrango}{\Delta tiempo} = \frac{0.022 - 9.925}{1 - 0.5} = -19.806 \text{ m/s}$$

Y con la estimación de la velocidad podremos extraer la frecuencia Doppler que experimenta la señal GPS en cada uno de los dos tramos:

$$f_{dopESTIMADA\ ASCENDENTE} = f_{L1} \frac{v_{ESTIM\ ASC}}{c} = 1575.42 \cdot 10^6 \frac{19.850}{299792458} = 104.31 \text{ Hz}$$

$$f_{dopESTIMADA\ DESCENDENTE} = f_{L1} \frac{v_{ESTIM\ DES}}{c} = 1575.42 \cdot 10^6 \frac{-19.806}{299792458} = -104.08 \text{ Hz}$$

Finalmente, calcularemos los parámetros reales de la simulación:

$$v_{REAL\ ASCENDENTE} = 20 \text{ m/s}$$

$$v_{REAL\ DESCENDENTE} = -20 \text{ m/s}$$

$$f_{dop_{REAL\ ASCENDENTE}} = f_{L1} \frac{v_{REAL\ ASC}}{c} = 1575.42 \cdot 10^6 \frac{20}{299792458} = 105.1 \text{ Hz}$$

$$f_{dop_{REAL\ DESCENDENTE}} = f_{L1} \frac{v_{REAL\ DES}}{c} = 1575.42 \cdot 10^6 \frac{-20}{299792458} = -105.1 \text{ Hz}$$

La siguiente tabla resume los valores de frecuencia Doppler encontrados (Tabla 6.2):

f_{dop} [Hz]	Real	Estimada
Tramo Ascendente	105.1	104.31
Tramo Descendente	-105.1	-104.08

Tabla 6.2 Comparación de f_{dop} : real y estimada

Por tanto, atendiendo a los valores de la Tabla 6.2, en esta simulación también concluimos que el DLL ha hecho una estimación razonable del código, y lo ha mantenido alineado proporcionando una variación de distancia próxima a la real.

6.1.4.3 Dinámica de tres tramos

La tercera simulación también se realizará con velocidades constantes, pero esta vez de diferente magnitud y en tres tramos. Durante el primer tramo, el satélite se acercará al receptor durante un cuarto de segundo a una velocidad de 20 m/s, por lo que le recortará 5 m. Durante el segundo tramo, será el receptor quien se acercará al satélite durante medio segundo a una velocidad de 10 m/s, y volverán a sus posiciones iniciales porque aunque el receptor se acerque a la mitad de velocidad, lo hace durante el doble de tiempo. Finalmente, en el tercer tramo el satélite se volverá a acercar al receptor a razón de 10 m/s durante un cuarto de segundo, por lo que reducirá 2.5 m su distancia.

De la misma manera que hemos hecho con las anteriores simulaciones, representaremos el pseudorrango (imagen derecha de la Fig. 6.5), y lo compararemos con la variación de distancia real que experimentan satélite y receptor (imagen izquierda de la Fig. 6.5):

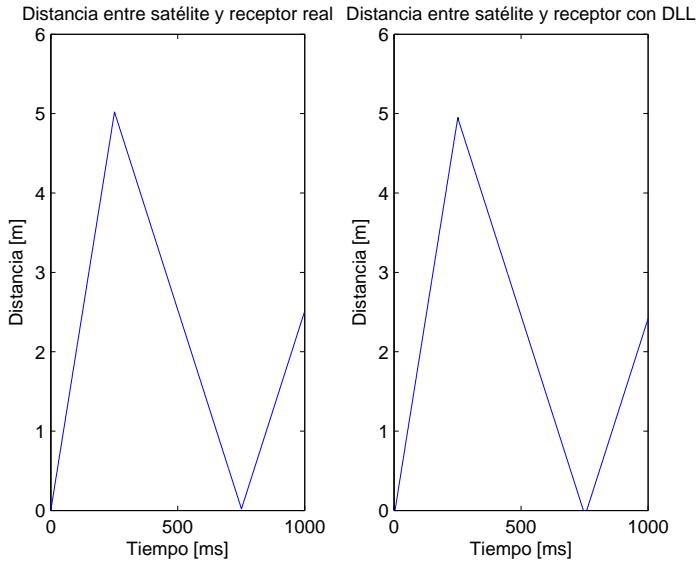


Fig. 6.5. Variación real de distancia (izquierda), variación con el DLL (derecha)

Observando la gráfica de la derecha de la figura anterior (Fig. 6.5), podemos calcular la velocidad relativa entre satélite y receptor dividiendo el pseudorrango por el tiempo:

$$v_{ESTIMADA\ TRAMO\ 1} = \frac{\Delta p_{pseudorrango}}{\Delta t_{tiempo}} = \frac{4.925 - 0}{0.25 - 0} = 19.7 \text{ m/s}$$

$$v_{ESTIMADA\ TRAMO\ 2} = \frac{\Delta p_{pseudorrango}}{\Delta t_{tiempo}} = \frac{-0.043 - 4.925}{0.75 - 0.25} = -9.936 \text{ m/s}$$

$$v_{ESTIMADA\ TRAMO\ 3} = \frac{\Delta p_{pseudorrango}}{\Delta t_{tiempo}} = \frac{2.414 - (-0.043)}{1 - 0.75} = 9.828 \text{ m/s}$$

Y con la estimación de la velocidad podemos extraer la frecuencia Doppler que experimenta la señal GPS en cada uno de los tres tramos:

$$f_{dop_{ESTIMADA\ TRAMO\ 1}} = f_{L1} \frac{v_{ESTIMADA\ TRAMO\ 1}}{c} = 1575.42 \cdot 10^6 \frac{19.7}{299792458} = 103.52 \text{ Hz}$$

$$f_{dop_{ESTIMADA\ TRAMO\ 2}} = f_{L1} \frac{v_{ESTIM\ TRAMO\ 2}}{c} = 1575.42 \cdot 10^6 \frac{-9.936}{299792458} = -52.21 \text{ Hz}$$

$$f_{dop_{ESTIMADA\ TRAMOS\ 3}} = f_{L1} \frac{v_{ESTIM\ TRAMO\ 3}}{c} = 1575.42 \cdot 10^6 \frac{9.828}{299792458} = 51.64 \text{ Hz}$$

Finalmente, calcularemos los parámetros reales de la simulación:

$$v_{REAL\ TRAMO\ 1} = 20 \text{ m/s}$$

$$v_{REAL\ TRAMO\ 2} = -10 \text{ m/s}$$

$$v_{REAL\ TRAMO\ 3} = 10 \text{ m/s}$$

$$f_{dop_{REAL\ TRAMO\ 1}} = f_{L1} \frac{v_{REAL\ TRAMO\ 1}}{c} = 1575.42 \cdot 10^6 \frac{20}{299792458} = 105.1 \text{ Hz}$$

$$f_{dop_{REAL\ TRAMO\ 2}} = f_{L1} \frac{v_{REAL\ TRAMO\ 2}}{c} = 1575.42 \cdot 10^6 \frac{-10}{299792458} = -52.55 \text{ Hz}$$

$$f_{dop_{REAL\ TRAMO\ 3}} = f_{L1} \frac{v_{REAL\ TRAMO\ 3}}{c} = 1575.42 \cdot 10^6 \frac{10}{299792458} = 52.55 \text{ Hz}$$

La siguiente tabla resume los valores de frecuencia Doppler encontrados (Tabla 6.3):

f_{dop} [Hz]	Real	Estimada
Tramo 1	105.1	103.52
Tramo 2	-52.55	-52.21
Tramo 3	52.55	51.64

Tabla 6.3. Comparación de f_{dop} : real y estimada

Por tanto, atendiendo a los valores de la Tabla 6.3, concluimos que el DLL ha hecho una estimación razonable del código, y lo ha mantenido alineado proporcionando una variación de distancia próxima a la real.

6.1.5 Cálculo del observable de fase con la arquitectura PLL

De forma similar a la arquitectura DLL, podemos calcular el observable de fase en metros entre satélite y receptor utilizando el PLL. Tal y como hemos presentado en el subapartado 6.1.3, el valor que genera la variable NCO_{PLL} del módulo de tracking debe ser igual a la fase de la señal GPS:

$$\varphi = \text{NCO}_{\text{PLL}} = \text{fase}_{\text{señal GPS}} = 2\pi \cdot f_{dop} \cdot \Delta t + \theta \quad [\text{rad}]$$

Recuperando la definición de frecuencia Doppler:

$$f_{dop} = f_{L1} \frac{v}{c}$$

Sustituyendo, calculamos el observable de fase en radianes:

$$\text{observable de fase [rad]} = \text{NCO}_{\text{PLL}} = 2\pi \cdot f_{L1} \frac{v}{c} \cdot \Delta t + \theta$$

De esta manera, aislando el producto entre la velocidad y el tiempo, podemos calcular el observable de fase en metros:

$$\text{observable de fase [m]} = v \cdot \Delta t = \frac{\text{NCO}_{\text{PLL}}}{2\pi} \cdot \frac{c}{f_{L1}}$$

Siendo el observable de fase en ciclos:

$$\text{observable de fase [ciclos]} = \frac{\text{NCO}_{\text{PLL}}}{2\pi}$$

A continuación generaremos tres simulaciones para ilustrar lo explicado en este apartado. Para cada una de ellas, generaremos la gráfica de la fase estimada, calcularemos la desviación Doppler a partir de su pendiente, posteriormente generaremos la gráfica del observable de fase en metros, y finalmente compararemos los valores teóricos con los calculados para analizar si el tracking está funcionando correctamente.

6.1.5.1 Dinámica de un tramo

En esta primera simulación, el satélite se acercará al receptor a una velocidad constante de 10 m/s. En el apartado 6.1.3 hemos llegado a la siguiente relación:

$$\text{NCO}_{\text{PLL}} = 2\pi \cdot f_{\text{dop}} \cdot \Delta t + \theta$$

De esta manera, si representamos la variable NCO_{PLL} en función del tiempo (Fig. 6.6), concluimos que la pendiente de la gráfica debe ser f_{dop} escalada un término 2π .

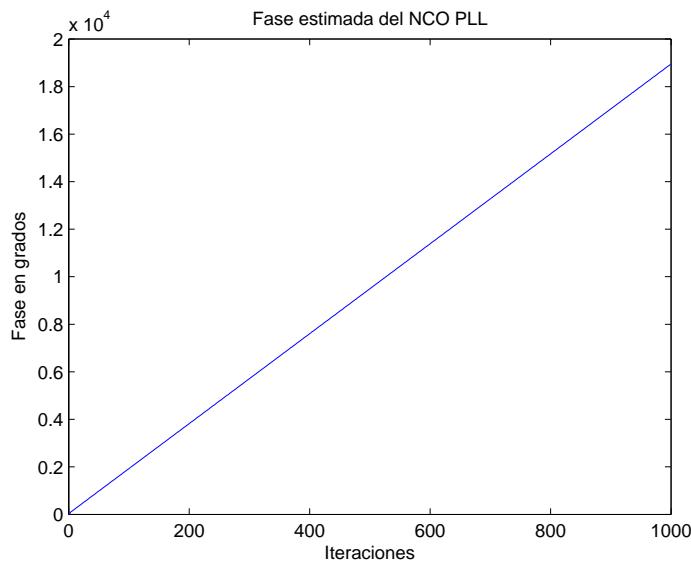


Fig. 6.6. Fase estimada por el NCO_{PLL}

Aislando:

$$f_{\text{dop}} = \frac{1}{2\pi} \frac{\Delta \text{NCO}_{\text{PLL}} - \theta}{\Delta t}$$

Por tanto, observando los valores de la gráfica anterior, podemos calcular el valor de la frecuencia Doppler despreciando el término θ porque el PLL empieza siempre su simulación en 0:

$$f_{\text{dop ESTIMADA ASCENDENTE}} = \frac{1}{2\pi} \cdot \frac{\text{NCO}_{\text{PLL}}(1000)_{\text{rad}} - \text{NCO}_{\text{PLL}}(0)_{\text{rad}}}{1 - 0}$$

$$f_{\text{dop ESTIMADA ASCENDENTE}} = \frac{1}{2\pi} \cdot \frac{18839.5 \frac{\pi}{180} - 0}{1} = 52.33 \text{ Hz}$$

Con el valor de la frecuencia Doppler, también podemos estimar la velocidad relativa entre satélite y receptor:

$$v_{ESTIMADA} = C \cdot \frac{f_{dop_{ESTIMADA\ ASCENDENTE}}}{f_{L1}} = 299792458 \cdot \frac{52.33}{1575.42 \cdot 10^6} = 9.958 \text{ m/s}$$

Una vez analizado el comportamiento de la fase estimada, vamos a estimar la distancia que separa el satélite del receptor. Recordemos que en esta primera simulación el satélite se acerca a una velocidad constante de 10 m/s al receptor.

Recuperando la expresión para graficar el observable de fase en metros:

$$\text{observable de fase [m]} = v \cdot \Delta t = \frac{\text{NCO}_{\text{PLL}}}{2\pi} \cdot \frac{c}{f_{L1}}$$

Nos damos cuenta que su gráfica debe tener la misma forma que la de la fase estimada (Fig. 6.6), con la salvedad del término $\frac{c}{2\pi f_{L1}}$, que únicamente aplica un factor de escala.

La siguiente gráfica (Fig. 6.7) compara la variación de distancia real (izquierda) con la variación de distancia estimada (derecha).

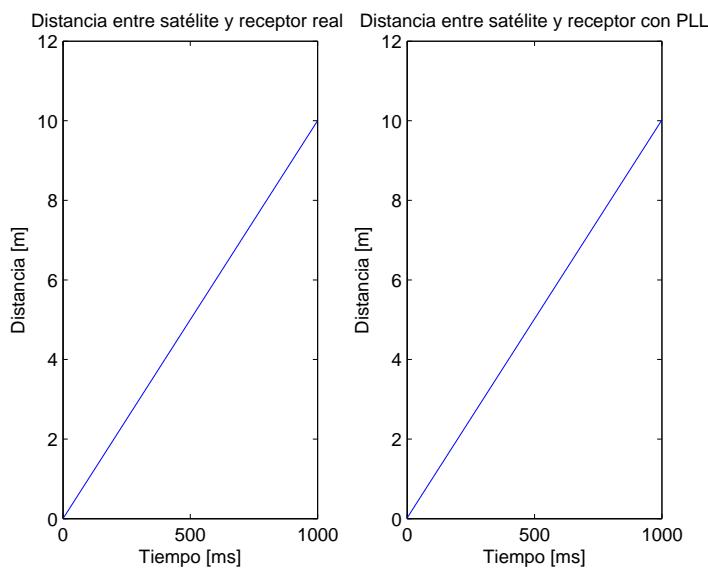


Fig. 6.7. Variación real de distancia (izquierda), variación con el PLL (derecha)

Observamos que ambas gráficas son prácticamente iguales y devuelven el resultado esperado, además de ser proporcionales a la gráfica de la fase estimada. El siguiente

paso es analizar la frecuencia Doppler real que experimenta la señal GPS al llegar al receptor:

$$f_{dopREAL} = f_{L1} \frac{v}{c} = 1575.42 \cdot 10^6 \frac{10}{299792458} = 52.55 \text{ Hz}$$

La siguiente tabla resume los valores de frecuencia Doppler encontrados (Tabla 6.4):

f_{dop} [Hz]	Real	Estimada
Tramo Ascendente	52.55	52.33

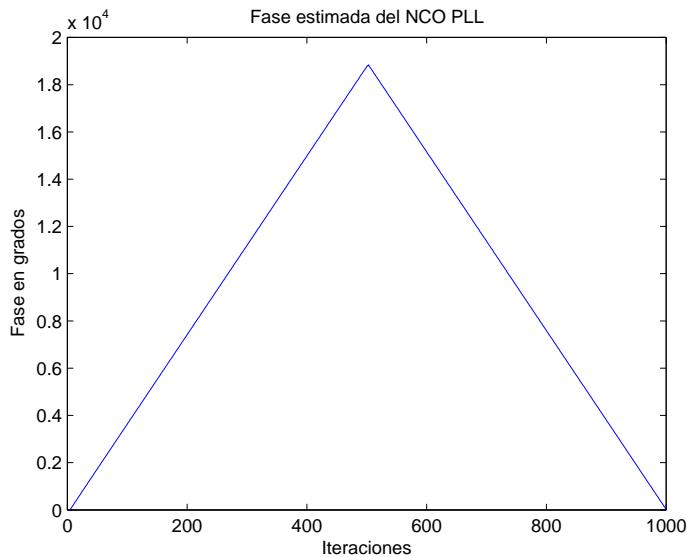
Tabla 6.4. Comparación de f_{dop} : real y estimada

Por tanto, concluimos que el tracking ha hecho una buena estimación de la fase de la señal GPS, ya que la pendiente de su estimación respecto del tiempo es proporcional a su frecuencia Doppler y eso provoca una buena estimación de su fase.

6.1.5.2 Dinámica de dos tramos

En la segunda simulación el satélite se acercará a una velocidad constante de 20 m/s al receptor y, a mitad de simulación, se alejará a la misma velocidad. Por tanto, a priori la estimación de la fase debería tener forma de triángulo, con un primer tramo ascendente y un segundo descendente, ambos de pendiente proporcional a la desviación Doppler que provoca la velocidad de 20 m/s.

Representaremos la variable NCO_{PLL} obtenida en el módulo de tracking en función del tiempo (Fig. 6.8):

Fig. 6.8. Fase estimada por el NCO_{PLL}

Y tal y como hemos visto anteriormente:

$$f_{dop} = \frac{1}{2\pi} \frac{\Delta \text{NCO}_{\text{PLL}} - \theta}{\Delta t}$$

Por tanto, observando los valores de la gráfica anterior, podemos calcular el valor de la frecuencia Doppler en cada uno de los dos tramos:

$$f_{dop \text{ESTIMADA ASCENDENTE}} = \frac{1}{2\pi} \cdot \frac{\text{NCO}_{\text{PLL}}(500)_{rad} - \text{NCO}_{\text{PLL}}(0)_{rad}}{0.5 - 0}$$

$$f_{dop \text{ESTIMADA ASCENDENTE}} = \frac{1}{2\pi} \cdot \frac{18839.5 \frac{\pi}{180} - 0}{0.5} = 104.7 \text{ Hz}$$

$$f_{dop \text{ESTIMADA DESCENDENTE}} = \frac{1}{2\pi} \cdot \frac{\text{NCO}_{\text{PLL}}(1000)_{rad} - \text{NCO}_{\text{PLL}}(500)_{rad}}{1 - 0.5}$$

$$f_{dop \text{ESTIMADA DESCENDENTE}} = \frac{1}{2\pi} \cdot \frac{\frac{\pi}{180} (28.4 - 18839.5)}{1 - 0.5} = -104.5 \text{ Hz}$$

Con los valores de la frecuencia Doppler, también podemos estimar la velocidad relativa entre satélite y receptor:

$$v_{ESTIMADA\ ASC} = C \cdot \frac{f_{dop_{ESTIMADA\ ASC}}}{f_{L1}} = 299792458 \cdot \frac{104.7}{1575.42 \cdot 10^6} = 19.923 \text{ m/s}$$

$$v_{ESTIMADA\ DES} = C \cdot \frac{f_{dop_{ESTIMADA\ DES}}}{f_{L1}} = 299792458 \cdot \frac{-104.5}{1575.42 \cdot 10^6} = -19.885 \text{ m/s}$$

Y así, estimar las distancias:

$$distancia_{ESTIMADA\ ASC} = v_{ESTIMADA\ ASC} \cdot \Delta t = 19.923 \cdot 0.5 = 9.961 \text{ m}$$

$$distancia_{ESTIMADA\ DES} = v_{ESTIMADA\ DES} \cdot \Delta t = -19.885 \cdot 0.5 = -9.942 \text{ m}$$

El siguiente paso es ver la estimación de la distancia calculada por el PLL, y compararla con la variación real. Recordemos que en esta segunda simulación el satélite se acerca a una velocidad constante de 20 m/s al receptor y, a mitad de simulación, se aleja a la misma velocidad.

Recuperando la expresión del observable de fase en metros, graficaremos su evolución (Fig. 6.9):

$$observable\ de\ fase\ [m] = v \cdot \Delta t = \frac{NCO_{PLL}}{2\pi} \cdot \frac{c}{f_{L1}}$$

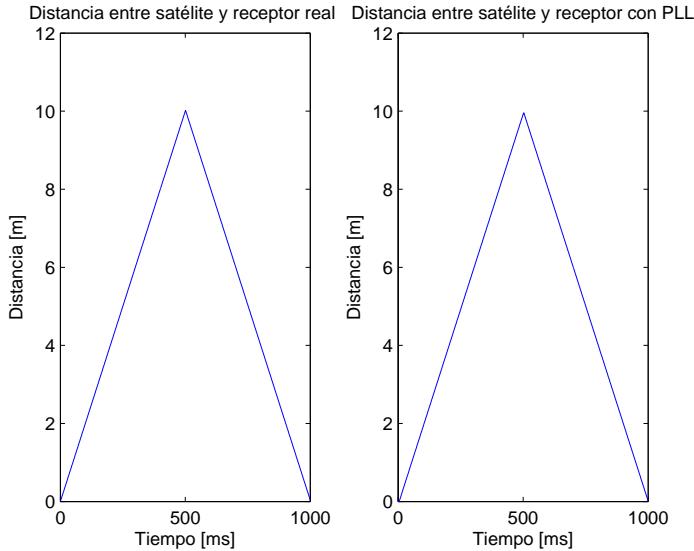


Fig. 6.9. Variación real de distancia (izquierda), variación con el PLL (derecha)

Observamos que ambas gráficas son razonablemente parecidas y devuelven el resultado esperado, además de ser proporcionales a la gráfica de la fase estimada. En el tramo ascendente el satélite se acerca 10 m al receptor a razón de 20 m/s. Si en un segundo recorre 20 m, en medio solo recorrerá 10 m. El análisis del tramo descendente es similar, y al cabo de 1 s satélite y receptor estarán a su distancia inicial.

El siguiente paso es analizar la frecuencia Doppler real que experimenta la señal GPS al llegar al receptor:

$$f_{dop\text{REAL ASCENDENTE}} = f_{L1} \frac{v}{c} = 1575.42 \cdot 10^6 \frac{20}{299792458} = 105.1 \text{ Hz}$$

$$f_{dop\text{REAL DESCENDENTE}} = f_{L1} \frac{v}{c} = 1575.42 \cdot 10^6 \frac{-20}{299792458} = -105.1 \text{ Hz}$$

La siguiente tabla resume los valores de frecuencia Doppler encontrados (Tabla 6.5):

f_{dop} [Hz]	Real	Estimada
Tramo Ascendente	105.1	104.7
Tramo Descendente	-105.1	-104.5

Tabla 6.5. Comparación de f_{dop} : real y estimada

Y, por tanto, de nuevo concluimos que el módulo de tracking ha realizado una buena estimación de la fase de la señal GPS.

6.1.5.3 Dinámica de tres tramos

La tercera simulación también se realizará con velocidades constantes, pero esta vez de diferente magnitud y en tres tramos. Durante el primer tramo, el satélite se acercará al receptor durante un cuarto de segundo a una velocidad de 20 m/s, por lo que le recortará 5 m. Durante el segundo tramo, será el receptor quien se acercará al satélite durante medio segundo a una velocidad de 10 m/s, y volverán a sus posiciones iniciales porque aunque el receptor se acerque a la mitad de velocidad, lo hace durante el doble de tiempo. Finalmente, en el tercer tramo el satélite se volverá a acercar al receptor a razón de 10 m/s durante un cuarto de segundo, por lo que reducirá 2.5 m su distancia.

De la misma manera que hemos hecho con las anteriores simulaciones, generaremos la gráfica de la fase estimada por el PLL (Fig. 6.10), y estimaremos la desviación Doppler que experimenta la señal GPS.

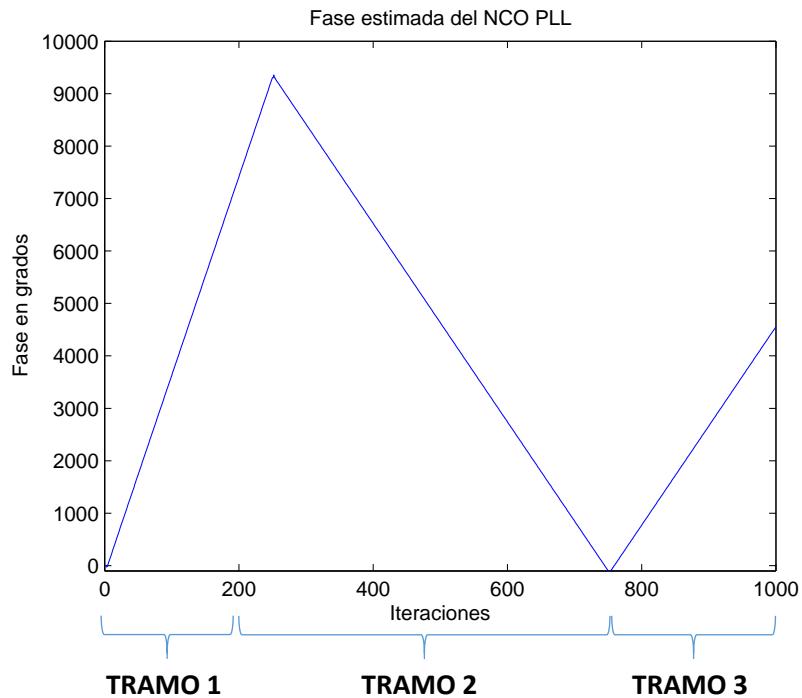


Fig. 6.10. Fase estimada por el NCO_{PLL}

Una vez obtenida la fase estimada, estimaremos también los valores de la frecuencia Doppler por tramos:

$$f_{dop\text{ESTIMADA TRAMO } 1} = \frac{1}{2\pi} \cdot \frac{\text{NCO}_{\text{PLL}}(250)_{rad} - \text{NCO}_{\text{PLL}}(0)_{rad}}{0.25 - 0}$$

$$f_{dop\text{ESTIMADA TRAMO } 1} = \frac{1}{2\pi} \cdot \frac{9304.3 \frac{\pi}{180} - 0}{0.25 - 0} = 103.38 \text{ Hz}$$

$$f_{dop\text{ESTIMADA TRAMO } 2} = \frac{1}{2\pi} \cdot \frac{\text{NCO}_{\text{PLL}}(750)_{rad} - \text{NCO}_{\text{PLL}}(250)_{rad}}{0.75 - 0.25}$$

$$f_{dop\text{ESTIMADA TRAMO } 2} = \frac{1}{2\pi} \cdot \frac{\frac{\pi}{180}(-94.7 - 9304.3)}{0.75 - 0.25} = -52.21 \text{ Hz}$$

$$f_{dop\text{ESTIMADA TRAMO } 3} = \frac{1}{2\pi} \cdot \frac{\text{NCO}_{\text{PLL}}(1000)_{rad} - \text{NCO}_{\text{PLL}}(750)_{rad}}{1 - 0.75}$$

$$f_{dop\text{ESTIMADA TRAMO } 3} = \frac{1}{2\pi} \cdot \frac{\frac{\pi}{180}(4554.3 + 94.7)}{1 - 0.75} = 51.6 \text{ Hz}$$

Con los valores de la frecuencia Doppler, estimaremos la velocidad relativa entre satélite y receptor:

$$v_{\text{ESTIMADA TRAMO } 1} = C \cdot \frac{f_{dop\text{ESTIM TRAMO } 1}}{f_{L1}} = 299792458 \cdot \frac{103.38}{1575.42 \cdot 10^6} = 19.672 \text{ m/s}$$

$$v_{\text{ESTIMADA TRAMO } 2} = C \cdot \frac{f_{dop\text{ESTIM TRAMO } 2}}{f_{L1}} = 299792458 \cdot \frac{-52.21}{1575.42 \cdot 10^6} = -9.935 \text{ m/s}$$

$$v_{\text{ESTIMADA TRAMO } 3} = C \cdot \frac{f_{dop\text{ESTIM TRAMO } 3}}{f_{L1}} = 299792458 \cdot \frac{51.6}{1575.42 \cdot 10^6} = 9.819 \text{ m/s}$$

Y así, estimar las distancias:

$$\text{distancia}_{\text{ESTIMADA TRAMO } 1} = v_{\text{ESTIMADA TRAMO } 1} \cdot \Delta t = 19.672 \cdot 0.25 = 4.918 \text{ m}$$

$$\text{distancia}_{\text{ESTIMADA TRAMO } 2} = v_{\text{ESTIMADA TRAMO } 2} \cdot \Delta t = -9.935 \cdot 0.5 = -4.967 \text{ m}$$

$$\text{distancia}_{\text{ESTIMADA TRAMO } 3} = v_{\text{ESTIMADA TRAMO } 3} \cdot \Delta t = 9.819 \cdot 0.25 = 2.454 \text{ m}$$

El siguiente paso es estimar la variación de distancia calculada por el PLL, y compararla con la variación real (Fig. 6.11):

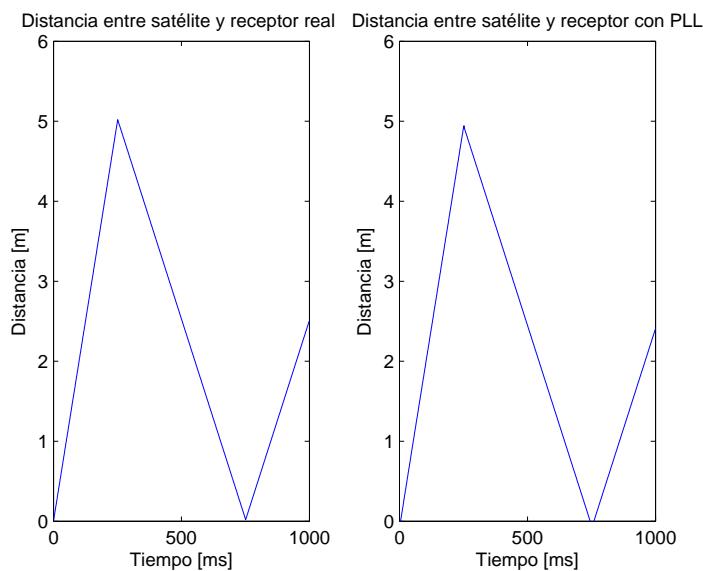


Fig. 6.11. Variación real de distancia (izquierda), variación con el PLL (derecha)

A partir de los valores de velocidad definidos al principio del subapartado, calcularemos de forma teórica los valores de la frecuencia Doppler:

$$f_{dop\text{REAL TRAMO } 1} = f_{L1} \frac{v}{c} = 1575.42 \cdot 10^6 \frac{20}{299792458} = 105.1 \text{ Hz}$$

$$f_{dop\text{REAL TRAMO } 2} = f_{L1} \frac{v}{c} = 1575.42 \cdot 10^6 \frac{-10}{299792458} = -52.55 \text{ Hz}$$

$$f_{dop\text{REAL TRAMO } 3} = f_{L1} \frac{v}{c} = 1575.42 \cdot 10^6 \frac{10}{299792458} = 52.55 \text{ Hz}$$

Los valores calculados de frecuencia Doppler son razonablemente parecidos a los teóricos. Por tanto, la fase que ha estimado el tracking es la correcta y el sistema de nuevo funciona correctamente (Tabla 6.6):

f_{dop} [Hz]	Real	Calculada
Tramo 1	105.1	103.38
Tramo 2	-52.55	-52.21
Tramo 3	52.55	51.6

Tabla 6.6. Comparación de f_{dop} : real y estimada

6.1.6 Comparativa entre DLL y PLL

En este último apartado de la sección compararemos los valores estimados y reales involucrados en el estudio de la distancia que hemos visto en los dos últimos apartados con la arquitectura DLL y PLL (Tabla 6.7).

Simulación	Sistema	v [m/s]	f_{dop} [Hz]	distancia [m]
Un tramo	Valores reales	10	52.55	10
	Estimación DLL	9.931	52.18	9.931
	Estimación PLL	9.958	52.33	9.958
Dos tramos	Valores reales	20	105.1	10
		-20	-105.1	-10
	Estimación DLL	19.850	104.31	9.925
		-19.806	-104.08	-9.903
	Estimación PLL	19.923	104.7	9.961
		-19.885	-104.5	-9.942
Tres tramos	Valores reales	20	105.1	5
		-10	-52.55	-5
		10	52.55	2.5
	Estimación DLL	19.7	103.52	4.925
		-9.936	-52.21	-4.968
		9.828	51.64	2.457
	Estimación PLL	19.672	103.38	4.918
		-9.935	-52.21	-4.967
		9.819	51.6	2.454

Tabla 6.7. Comparativa de simulaciones

Analizando la tabla anterior (Tabla 6.7), observamos cómo tanto la arquitectura DLL como la PLL se aproximan a los valores reales. Los valores sombreados son las estimaciones más próximas a los valores teóricos para cada simulación. Como las estimaciones de los tres parámetros analizados están relacionadas, observamos cómo si una arquitectura estima mejor un parámetro, también lo hará con el resto durante la misma simulación.

Ambas arquitecturas hacen unas estimaciones parecidas, si bien es cierto que la estimación PLL de las simulaciones de uno y dos tramos es levemente mejor a la del DLL, y el mismo comportamiento se da a la inversa con la última simulación.

6.2 Filtrado

Recordando el capítulo 4, donde presentamos la estructura de la señal GPS, dijimos que el ancho de banda de la señal GPS que envía el satélite era de 1.023 MHz y, además, que era prácticamente igual al de la señal de espectro ensanchado. Esta afirmación era parcialmente cierta, ya que sí es cierto que el espectro de ambas señales es muy similar, pero estrictamente lo que es igual a 1.023 MHz son los pasos por cero de ambos espectros, no el ancho de banda.

Tal y como hemos implementado la señal GPS, la hemos generado con un ancho de banda infinito y posteriormente la hemos muestreado a 5.456 MHz. No obstante, si el receptor tiene una restricción de ancho de banda deberemos filtrar la señal. Normalmente el satélite envía la señal GPS con un ancho de banda de ≈ 20 MHz, cogiendo el lóbulo principal y 9 secundarios a ambos lados (Fig. 6.12):

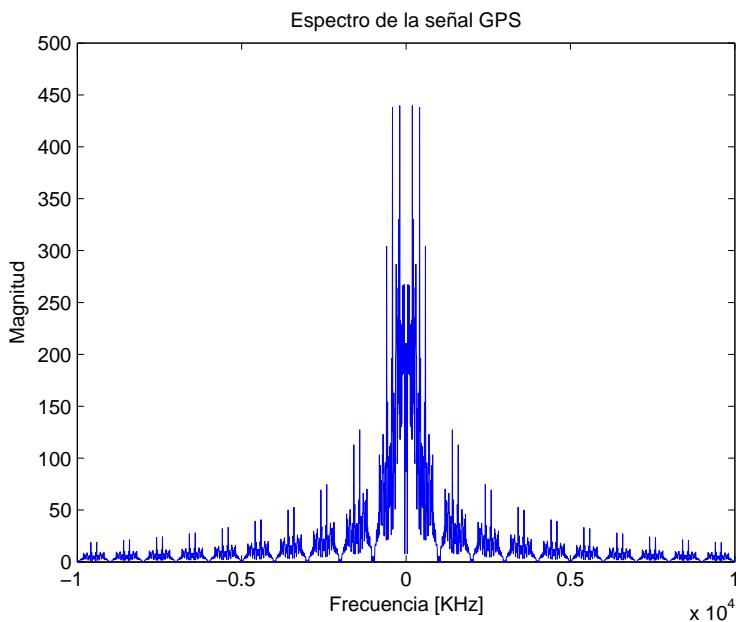


Fig. 6.12. Espectro de la señal GPS

Una vez la señal GPS llega al receptor, este normalmente la filtra a 4 MHz, quedándose con el lóbulo principal y los dos secundarios de mayor magnitud de la señal GPS. Esto no impide hacer el seguimiento de los valores de interés, porque se puede filtrar la señal y seguir midiendo su retardo y su fase. La selección del ancho de banda del receptor es una cuestión de diseño que responde al compromiso entre varias variables: a medida que aumentamos el ancho de banda mejora la precisión, a la vez que empeora su consumo de energía y su susceptibilidad a posibles interferencias.

Como ya hemos comentado a lo largo del proyecto, hemos implementado la señal GPS a la salida del *front-end*. Para ello, la hemos generado en banda base y posteriormente la hemos modulado a una frecuencia residual que simulaba la desviación Doppler.

Para evitar *aliasing*, filtraremos la señal GPS en banda base y, como hemos comentado anteriormente, nos quedaremos con un ancho de banda de 4 MHz. El ancho de banda de las señales en banda base se define como la diferencia entre la mayor y la menor frecuencia de la señal, pero sin la restricción de que sean positivas. Por tanto, la señal GPS en banda base tendrá un espectro de frecuencias entre -2 MHz y 2 MHz.

6.2.1 Filtro utilizado

Para efectuar el filtrado utilizaremos un filtro paso bajo digital de *Chebyshev*. Pero antes de adentrarnos en los detalles del filtro, analizaremos algunos aspectos importantes para comprender mejor el resultado final.

Cuando se muestrea una señal con una determinada frecuencia de muestreo f_m , las muestras de la señal se reparten en el rango $[-f_m/2, f_m/2]$ (Fig. 6.13):

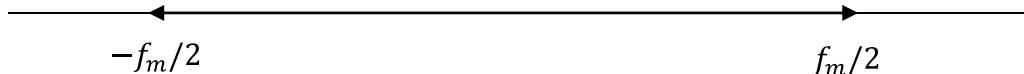


Fig. 6.13. Rango de frecuencias tras el muestreo

En nuestro caso, hemos trabajado con una $f_m = 5.456$ MHz, por lo que el rango de frecuencias de la señal GPS en banda base es de $[-2.728, 2.728]$ MHz (Fig. 6.14):

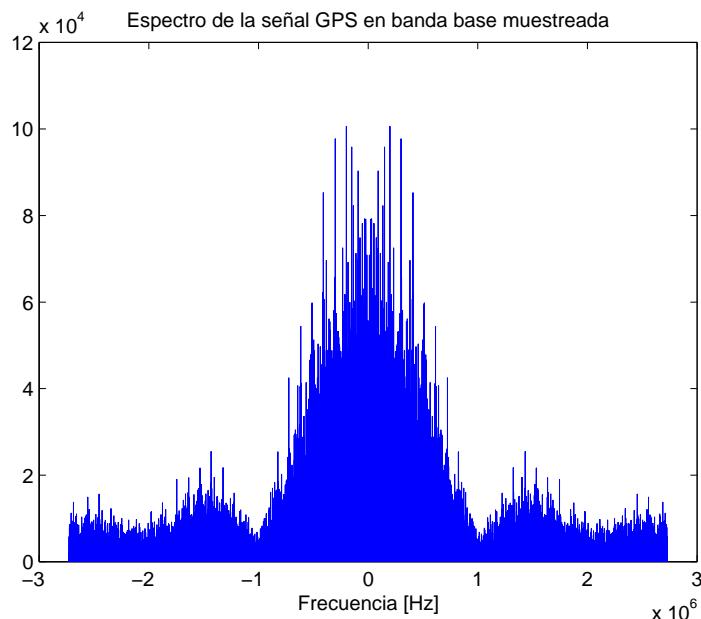


Fig. 6.14. Espectro de la señal GPS en banda base muestreada

Nuestro próximo objetivo es filtrar la señal anterior (Fig. 6.14) y así eliminar las componentes frecuenciales mayores de ± 2 MHz. Para ello utilizaremos un filtro paso bajo digital de *Chebyshev*.

Implementaremos el filtro en dos pasos. Primero generaremos sus coeficientes A y B, y luego implementaremos el filtro con la función filter utilizando los coeficientes previamente calculados.

Los coeficientes se generan mediante la siguiente instrucción:

`[B, A] = cheby2 (orden filtro, atenuación banda eliminada, frecuencia corte, 'low')`

Y durante las simulaciones utilizaremos los siguientes valores de los anteriores parámetros:

orden filtro = 3

atenuación banda eliminada = 20 dB

frecuencia corte = $2/(f_m/2)$

Con estas especificaciones obtenemos la respuesta frecuencial del filtro (Fig. 6.15):

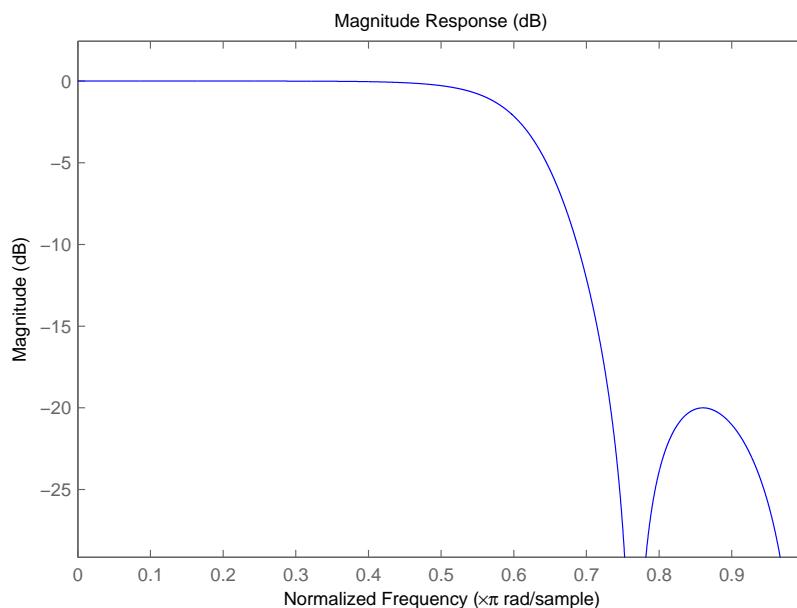


Fig. 6.15. Respuesta frecuencial del filtro paso bajo

Observamos como el eje de abscisas de la Fig. 6.15 representa la frecuencia normalizada de 0 a 1, que a su vez representan los valores comprendidos entre 0 y $f_m/2$ de la señal a filtrar (Fig. 6.14). Como podemos ver en la Fig. 6.15, existe una atenuación de 20 dB cuando la frecuencia normalizada toma un valor de 0.733:

$$\text{frecuencia}_{\text{normalizada}} = \text{frecuencia}_{\text{corte}} \left(\frac{f_m}{2} \right)^{-1}$$

Por tanto:

$$\text{frecuencia}_{\text{normalizada}} = 2 \text{ MHz} \left(\frac{5.456 \text{ MHz}}{2} \right)^{-1} = 0.733$$

De esta manera, a la frecuencia normalizada 0.733, que representa la frecuencia de corte de 2 MHz, la señal experimentará una atenuación de 20 dB y decaerá tal y como podemos comprobar en la Fig. 6.15.

Una vez definido el filtro y comprobado que cumple con las especificaciones, únicamente tenemos que aplicar los coeficientes obtenidos en el desarrollo anterior de la siguiente manera:

$$\text{Señal GPS muestreada filtrada} = \text{filter}(\text{B}, \text{A}, \text{Señal GPS muestreada})$$

Así, obtenemos la señal GPS en banda base muestreada filtrada a 2 MHz, tal y como podemos observar en la imagen de la derecha de la Fig. 6.16. La imagen de la izquierda de la Fig. 6.16 representa la misma figura que la Fig. 6.14, es decir, la señal de la derecha previa al filtrado.

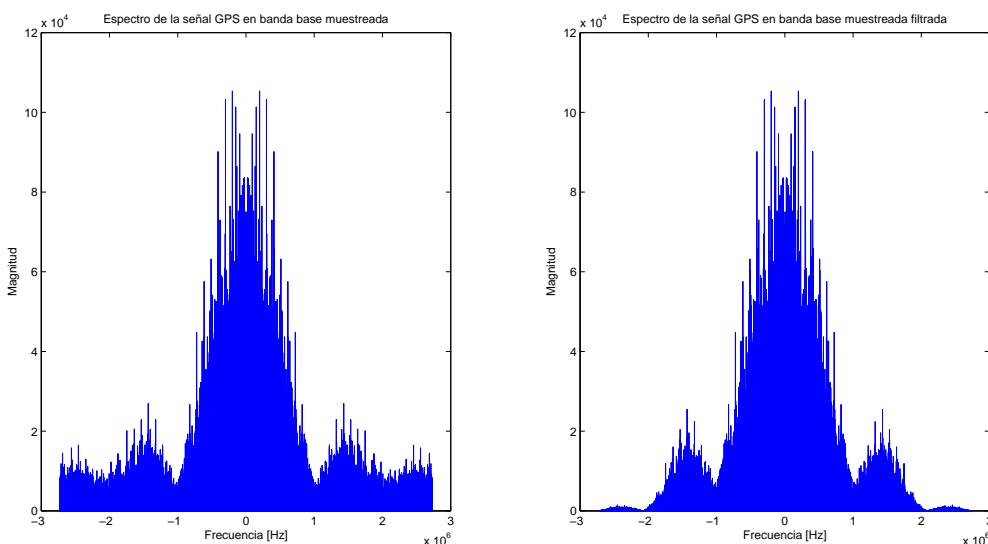


Fig. 6.16. Espectro de la señal GPS antes y después de aplicar el filtrado

6.2.2 Simulación de filtrado

Una vez definido el filtro, lo utilizaremos para realizar las simulaciones de esta sección. En la imagen izquierda de la Fig. 6.17 podemos apreciar la magnitud de las envolventes cuando la señal GPS no es filtrada, es decir, si utilizamos la señal GPS de la imagen de la izquierda de la Fig. 6.16. En este caso, observamos como la magnitud de las envolventes son las ideales, tal y como hemos explicado en el subapartado 5.4.1.7.

En cambio, en la imagen de la derecha de la Fig. 6.17 trabajamos con la señal GPS filtrada representada en la imagen derecha de la Fig. 6.16. El valor de la envolvente del código *prompt* es menor al ideal y, cuando se produce algún desajuste, como podemos ver en ciertos picos (imagen derecha Fig. 6.17), enseguida el tracking reajusta las envolventes al valor indicado. Como podemos observar, el filtrado no afecta demasiado al seguimiento de las envolventes que posteriormente se utilizarán para generar el nuevo código *prompt* del DLL, ya que las envolventes de los códigos *early* y *late* son iguales durante gran parte de la simulación.

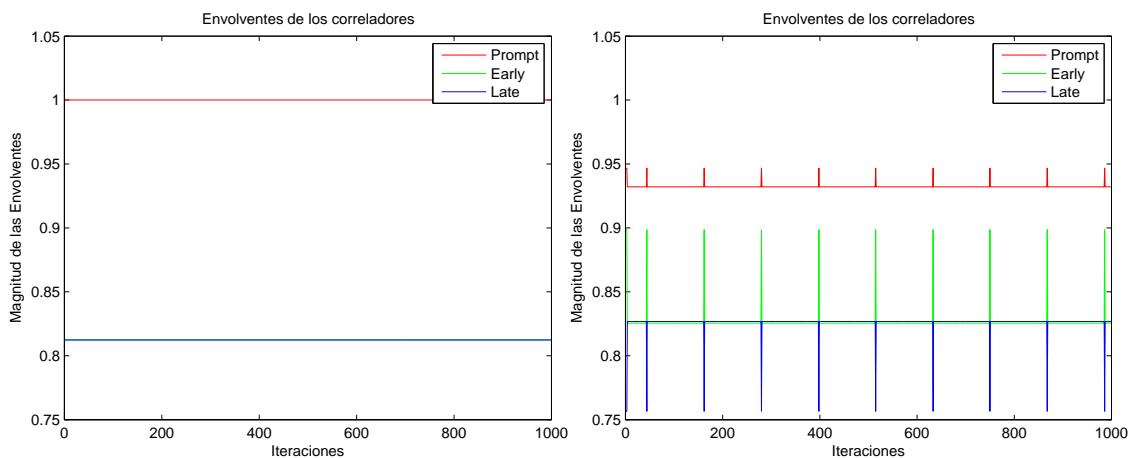


Fig. 6.17. Magnitud de las envolventes sin filtro (izquierda), y con filtro (derecha)

El filtro paso bajo elimina las altas frecuencias, asociadas a las variaciones rápidas de la señal. Por tanto, al eliminarlas, la señal evoluciona más suavemente, y este efecto se puede ver en la Fig. 6.18, donde con el efecto del filtrado el triángulo de correlaciones se vuelve más redondeado.

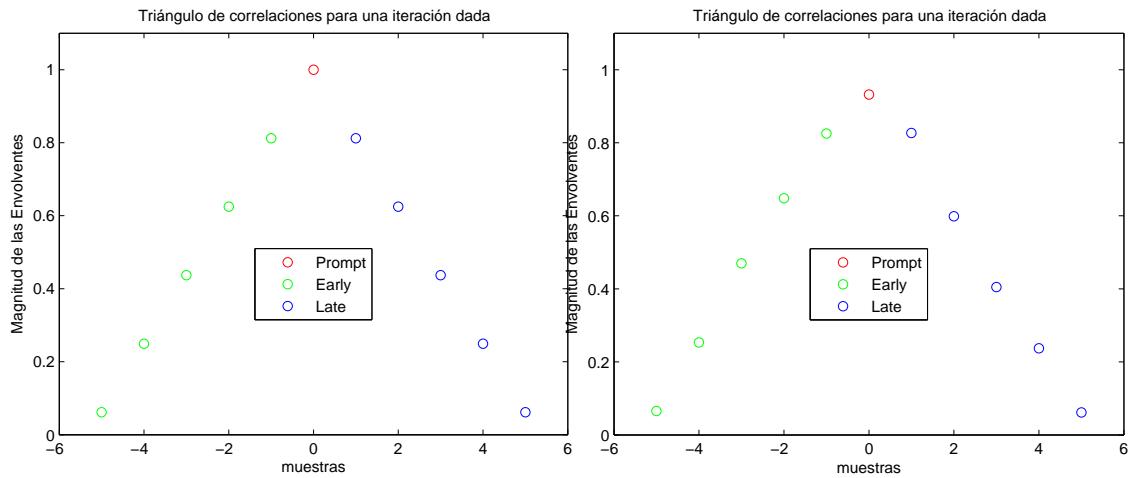


Fig. 6.18. Triángulo de correlaciones sin filtro (izquierda), y con filtro (derecha)

Los siguientes bloques de la arquitectura DLL son el discriminador y el filtro de lazo. Las dos imágenes superiores corresponden a la simulación de la señal GPS no filtrada, mientras que las dos inferiores corresponden a la señal GPS filtrada. Observamos cómo, filtrando, el DLL va ajustando a lo largo de las iteraciones el error de retraso de código, y este se va suavizando por el filtro de lazo (Fig. 6.19):

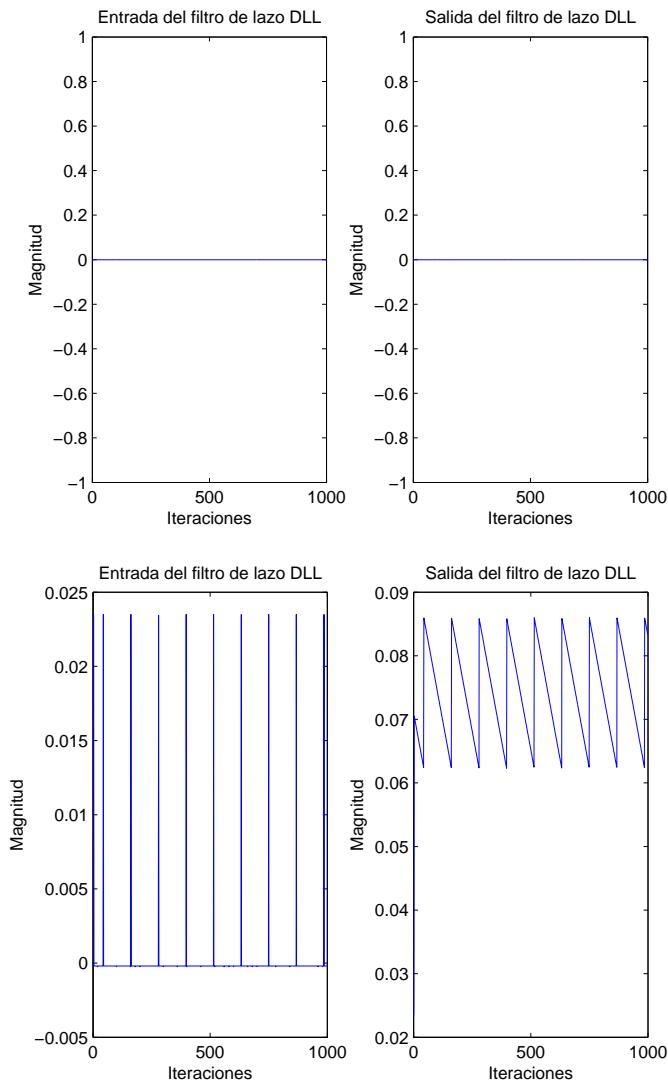


Fig. 6.19. Entrada y salida del filtro de lazo DLL sin filtro (superior) y con filtro (inferior)

A continuación analizaremos cuál es la respuesta del discriminador PLL. Recordemos que la entrada al discriminador es el número complejo $p = Ae^{j\varphi_{error}}$. El bucle de tracking consigue anular φ_{error} (Fig. 6.20), y el efecto del filtrado se aprecia en el módulo de p , haciendo que la envolvente del código *prompt* no sea igual a la unidad.

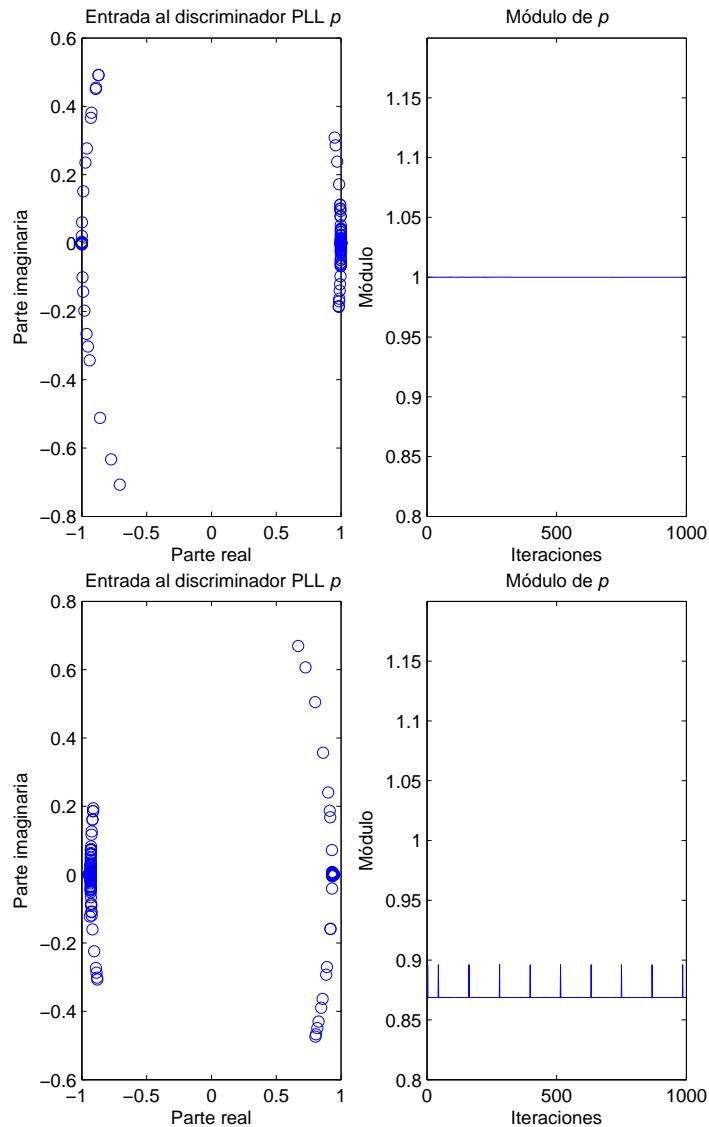


Fig. 6.20. Entrada al discriminador PLL sin filtro (superior), y con filtro (inferior)

El siguiente elemento dentro del PLL es el filtro de lazo. En la siguiente simulación utilizaremos un filtro de lazo de orden 1 que nos permita seguir la desviación Doppler introducida (Fig. 6.21). Observamos cómo no hay mucha diferencia entre las señales superiores (sin filtro), e inferiores (con filtro) de la Fig. 6.21.

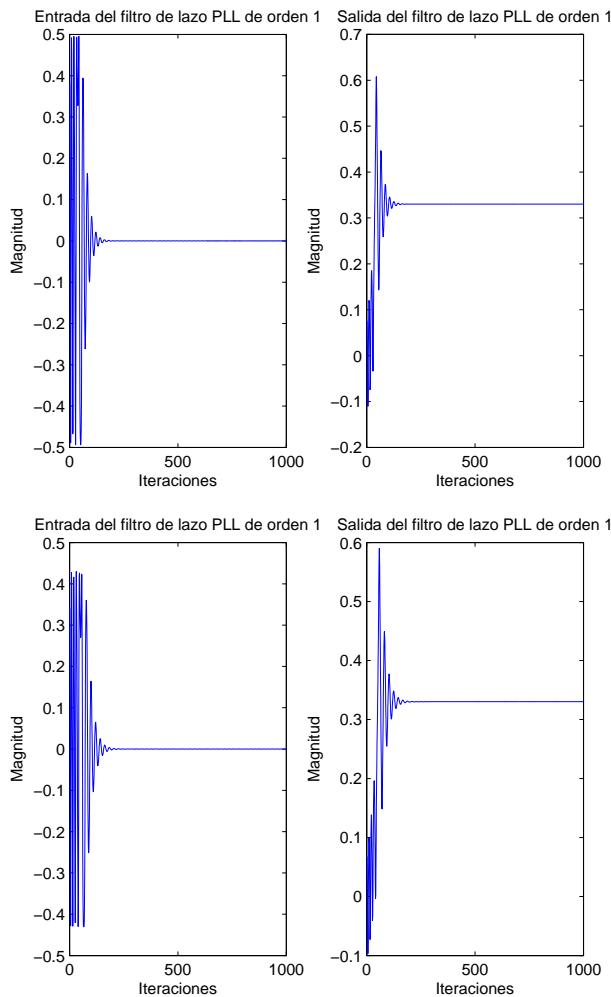


Fig. 6.21. Entrada y salida del filtro de lazo PLL de orden 1 sin filtro (superior) y con filtro (inferior)

Finalmente, analizaremos si el efecto del filtrado tiene consecuencias en el cálculo de la fase estimada, en este caso de 45° . Las siguientes dos imágenes están realizadas simulando la señal GPS sin desviación Doppler (Fig. 6.22):

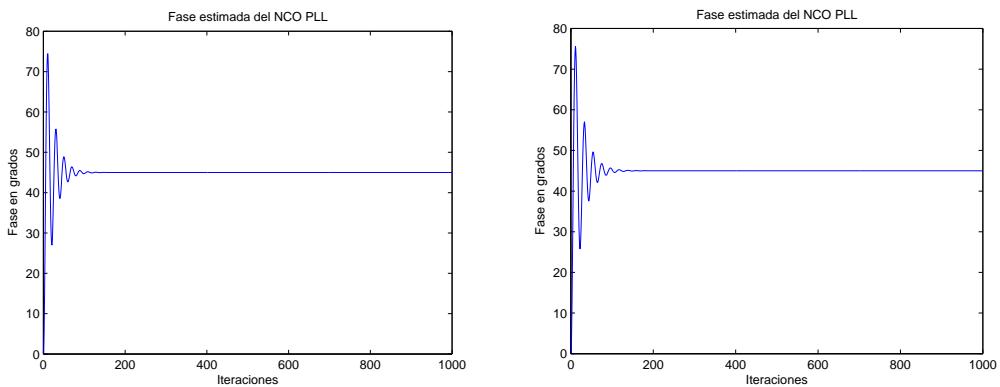


Fig. 6.22. Fase estimada sin desviación Doppler sin filtro (izquierda) y con filtro (derecha)

Analizando la Fig. 6.22, observamos como en ambas simulaciones el módulo de tracking converge a la fase indicada de manera similar. El receptor tiene el mismo comportamiento cuando la fase depende linealmente de la frecuencia Doppler (Fig. 6.23):

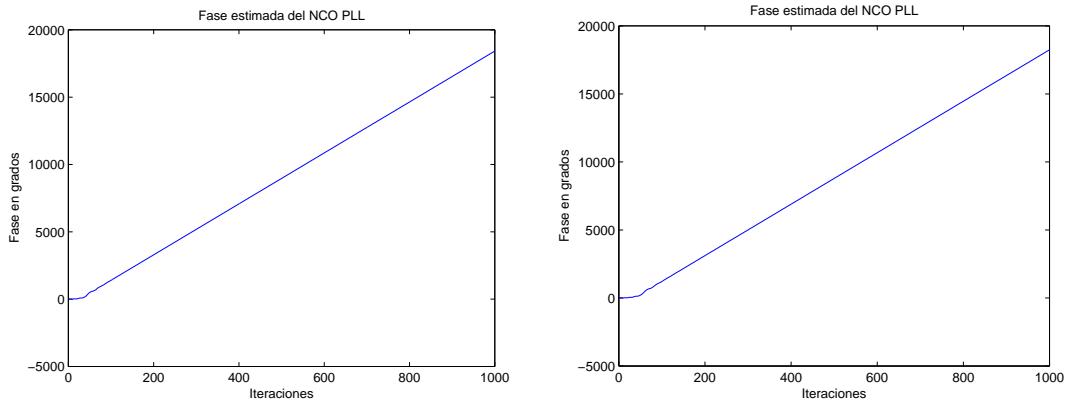


Fig. 6.23. Fase estimada con desviación Doppler sin filtro (izquierda) y con filtro (derecha)

Observamos cómo el filtrado no afecta demasiado a la estimación de la fase, ya que en las dos simulaciones anteriores con filtrado el módulo de tracking consigue estimar el valor de la fase correctamente. Y como no afecta a la estimación de la fase, tampoco lo hará en el cálculo del observable de fase en metros entre satélite y receptor (Fig. 6.24):

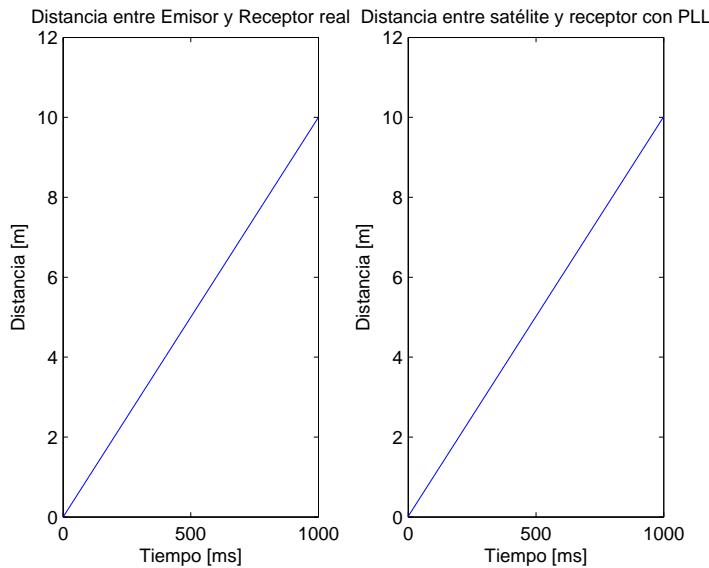


Fig. 6.24. Variación real de distancia (izquierda), variación filtrada con el PLL (derecha)

Por tanto, concluimos que el filtrado no afecta demasiado a la estimación de la fase y el retraso de la señal GPS, ya que hemos podido hacerles el seguimiento correctamente. El filtrado responde a la restricción de ancho de banda del receptor, y también se utiliza para eliminar posibles interferencias.

6.3 Multipath

La propagación multicamino²¹ consiste en que la señal GPS transmitida por el satélite llega al receptor a través de varios caminos, de manera que este recibe la señal directa, y varias versiones retardadas y atenuadas de la señal GPS (Fig. 6.25):

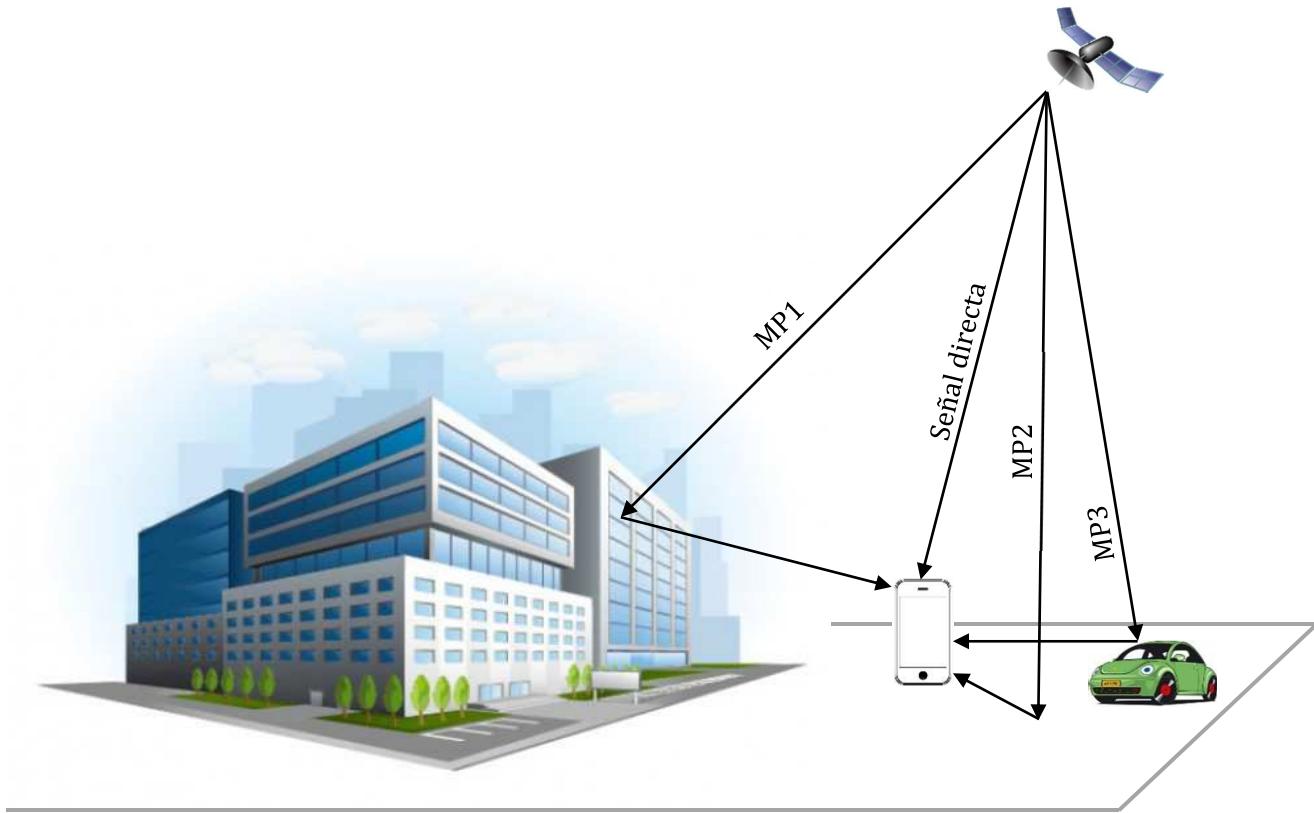


Fig. 6.25. Esquema de la propagación multicamino

En la Fig. 6.25, considerando como receptor el teléfono móvil, observamos cómo, además de la señal directa, el receptor recibe la señal GPS a través de varios caminos. En general, estas réplicas de la señal GPS llegan con un mayor retardo y atenuación, ya que recorren una mayor distancia hasta llegar al receptor, y las reflexiones provocan una pérdida de potencia de la señal.

Por tanto, la existencia de estas réplicas de la señal GPS provocan el denominado efecto multicamino, que hace alusión al error que causa la propagación multicamino en el seguimiento del retraso de código y desviación Doppler, parámetros que hemos

²¹ *Multipath* en inglés

analizado en el módulo de tracking. Las fuentes de error que causan el efecto multicamino son la amplitud, el retardo, la fase y la desviación Doppler.

El DLL calcula el retraso de código a partir de los valores del triángulo de correlación. Si la señal deseada se encuentra solapada con versiones retardadas de ella misma, se distorsionan los valores del triángulo, y con ello el valor de retraso de código. La fase medida por el PLL también se verá afectada por el efecto multicamino, ya que no será la fase de la señal directa, sino que estará afectada por los valores de la correlación de las reflexiones.

A continuación generaremos tres simulaciones en las que veremos cómo cambia el triángulo de correlaciones, que sirve para analizar el error de retraso de código, con una señal réplica. Si bien el modelo de dos rayos (señal directa y réplica) puede no ser muy representativo de escenarios reales, permite analizar el funcionamiento y comparar las diferentes técnicas que se usan para estudiar el *multipath*.

6.3.1 Simulación de retardo

La señal réplica es la señal GPS pero con una cierta atenuación, retardo, diferencia de fase y desviación Doppler sobre la original. En esta primera simulación hemos utilizado los siguientes parámetros para generar la señal réplica:

Amplitud = 1

Retardo = 3 muestras

Fase = Fase original

Doppler = Doppler original

Por tanto, la réplica tiene la misma amplitud que la señal GPS original y está retardada 3 muestras. Recordamos que un chip está formado por 5.33 muestras, con lo que este retardo corresponde aproximadamente a la mitad de un chip. Hemos mantenido la fase y la desviación Doppler de la señal original.

Si bien no es muy habitual que la señal réplica no presente atenuación sobre la señal directa, en esta primera simulación hemos querido generarlas con la misma amplitud para observar mejor el efecto multicamino en el triángulo de correlaciones.

Con estos parámetros, generamos el triángulo de correlaciones de ambas señales de manera superpuesta. Observamos cómo el triángulo centrado en la muestra 0 corresponde al de la señal directa, y el que está retrasado tres muestras corresponde a la señal réplica (Fig. 6.26):

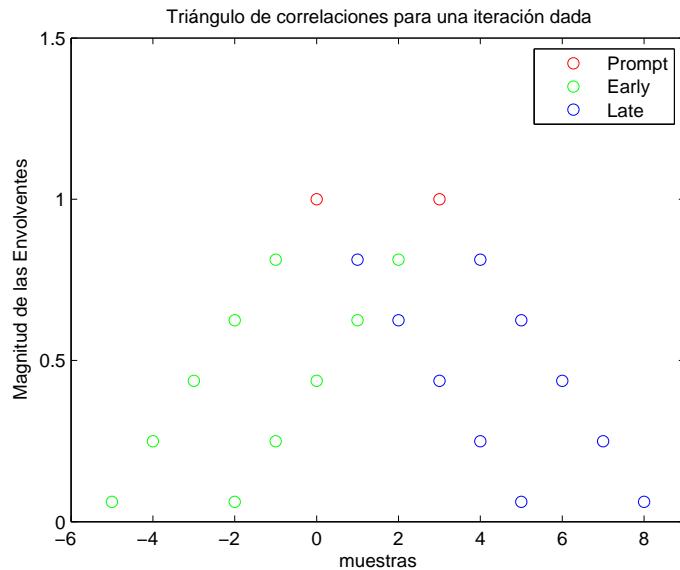


Fig. 6.26. Triángulo de correlaciones de la señal directa y de la señal réplica

Hemos generado el triángulo de correlaciones de manera superpuesta para analizar mejor el efecto multicamino, pero realmente el receptor no las recibe superpuestas, sino que recibe la suma de ambos triángulos (Fig. 6.27).

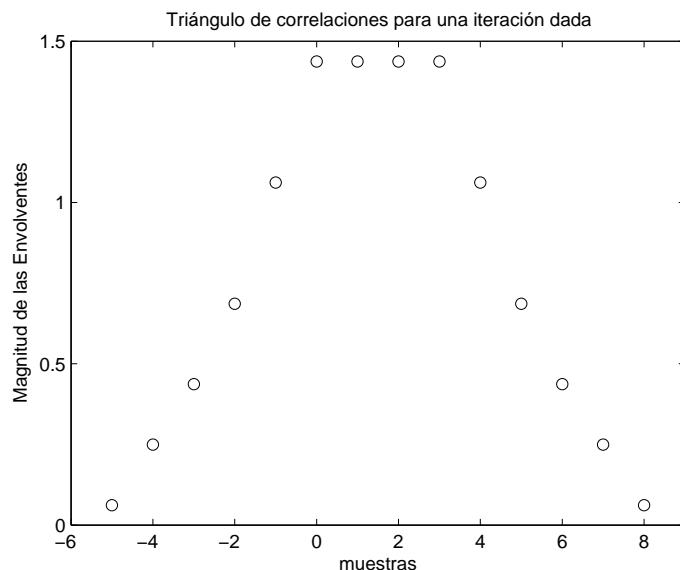


Fig. 6.27. Triángulo de correlaciones de la suma de señales

En la Fig. 6.27 observamos cómo los valores de las muestras $-5, -4, -3, 6, 7$ y 8 mantienen los valores de la señal directa, ya que los valores de correlación de la señal réplica son nulos. Los demás valores Fig. 6.27 son la suma de los valores de la Fig. 6.26 para cada par de muestras. Esto se puede apreciar en los valores centrales del triángulo, donde las pendientes de las rectas son iguales pero de signo contrario, y eso provoca una respuesta plana en las muestras $0, 1, 2$ y 3 .

Como podemos observar, se han desvirtuado los valores de las envolventes de los códigos *early* y *late*, y el discriminador DLL tendrá dificultades a la hora de estimar el retardo a aplicar en la siguiente iteración del bucle de tracking.

6.3.2 Simulación de retardo, fase y amplitud

En la segunda simulación atenuaremos y cambiaremos la fase de la señal réplica. Estos son los valores de los parámetros que hemos utilizado para generar la señal réplica:

Amplitud = 0.5

Retardo = 3 muestras

Fase = 180° sobre la fase original

Doppler = Doppler original

Los anteriores valores provocarán que las señales estén en contrafase, y los valores de correlación de la señal réplica sean negativos (Fig. 6.28):

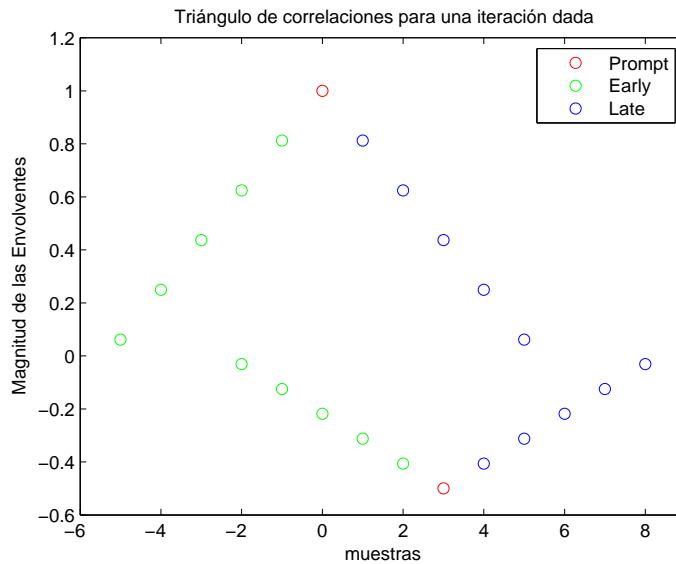


Fig. 6.28. Triángulo de correlaciones de la señal directa y de la señal réplica

De esta manera, el triángulo de correlaciones real que recibe el receptor es el siguiente (Fig. 6.29):

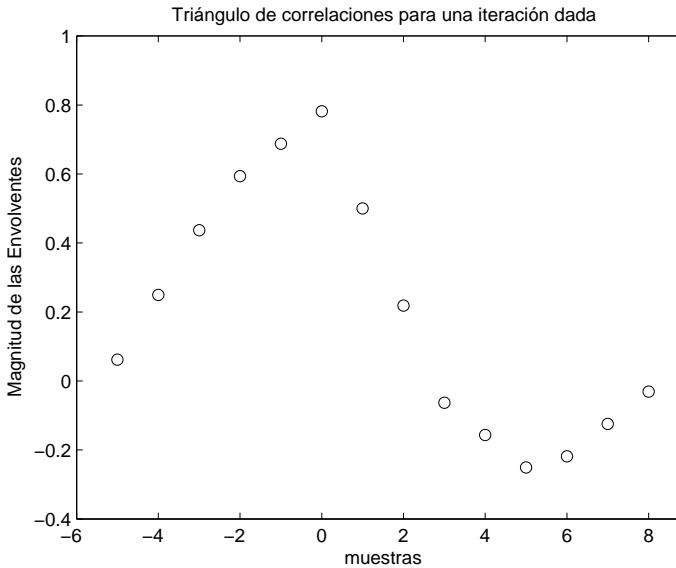


Fig. 6.29. Triángulo de correlaciones de la suma de señales

En esta segunda simulación observamos cómo, si no hubiera *multipath*, la salida del discriminador DLL sería nula, con lo que el código *prompt* estaría perfectamente alineado. La llegada de la señal réplica engaña al discriminador, ya que la envolvente *early* es mayor que la *late* y provoca un error en el remuestreo final del DLL.

6.3.3 Simulación de retardo y amplitud

Esta tercera simulación pretende ilustrar el efecto que tiene un retardo de más de un chip de la señal réplica sobre la señal directa. Como hemos comentado a lo largo del proyecto, un chip está formado por 5.33 muestras, por tanto, introduciremos un retardo de 6 muestras:

Amplitud = 0.5

Retardo = 6 muestras

Fase = Fase original

Doppler = Doppler original

Con estos parámetros, generamos el triángulo de correlaciones de ambas señales de manera superpuesta (Fig. 6.30):

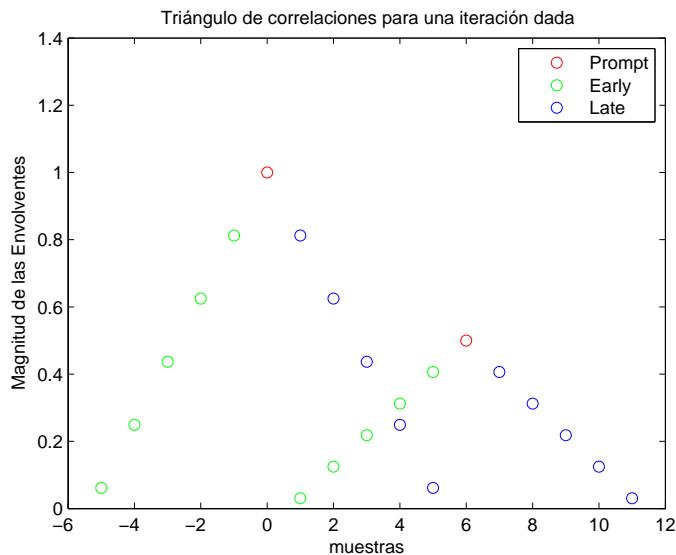


Fig. 6.30. Triángulo de correlaciones de la señal directa y de la señal réplica

De esta manera, el triángulo de correlaciones real que recibe el receptor es el siguiente (Fig. 6.31):

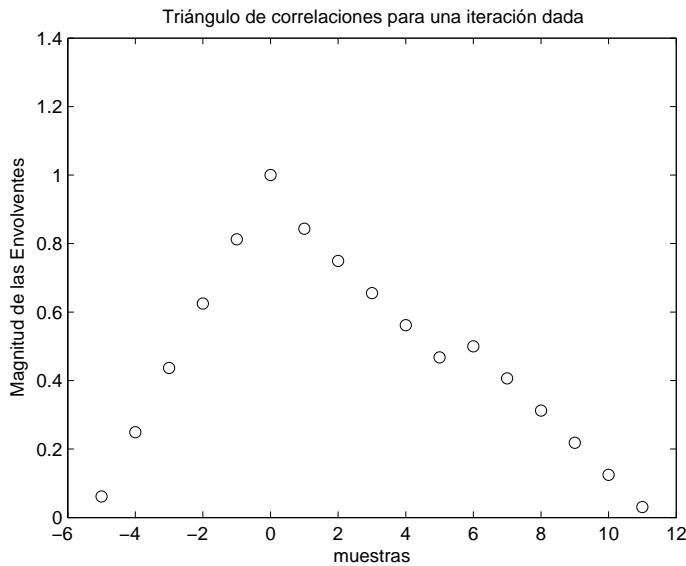


Fig. 6.31. Triángulo de correlaciones de la suma de señales

Si los retardos relativos de las diferentes componentes son comparables o mayores al tiempo de chip, el multicamino es una fuente de diversidad que permite mejorar las prestaciones del sistema. En este caso, el *multipath* solo afecta a la envolvente del código *late*, que aumenta de valor.

Hemos querido hacer esta tercera simulación para notar que la Fig. 6.31 se encuentra en el umbral entre el fenómeno del *multipath* como fuente error, y como fuente de diversidad. Si hubiéramos generado la señal réplica con una muestra más de retardo, los valores de las tres envolventes principales serían los ideales y el receptor podría analizar la señal directa y tratar la réplica para generar un sistema más robusto. Estas técnicas de diversidad no se han desarrollado en el proyecto, y solo se apunta su existencia para evidenciar la problemática.

6.4 Ruido

Los cálculos que realiza el receptor para calcular su posición están afectados por fluctuaciones aleatorias que contiene la señal GPS. Estas fluctuaciones suelen englobarse en el término genérico de ruido del receptor. Llamaremos ruido a la radiación de RF captada por la antena diferente a la de la señal de interés, al ruido generado en los componentes que forman el receptor (amplificadores, osciladores, filtros, etc.), y al ruido de cuantificación de la conversión analógico-digital.

El ruido tiene fundamentalmente origen térmico, pero también existen otros tipos de ruido, como por ejemplo el ruido *flicker*²² presente en los elementos activos como los osciladores. Como en todo tipo de sistemas de comunicaciones, el ruido emmascara la señal útil provocando la aparición de un *jitter*²³ en las medidas de los parámetros de retraso de código y desviación Doppler que realiza el receptor GPS.

El DLL no puede identificar con total precisión las transiciones de los chips, por lo que la medida de retraso de código no es perfecta y oscila alrededor del valor ideal, mientras que las medidas proporcionadas por el PLL también contienen un *jitter* de fase.

6.4.1 Modelo de ruido

En este apartado presentaremos el modelo de ruido AWGN²⁴ que vamos a utilizar en las siguientes simulaciones. La señal GPS que entra al módulo de adquisición está formada por un flujo constante de muestras, complejas y en banda base. Por tanto, estas dos últimas propiedades son las que también deberá tener el ruido: debe ser complejo y estar modulado en banda base. Además tendrá una varianza σ^2 y lo consideraremos de media nula.

De esta manera, el modelo de ruido es el siguiente:

$$\text{Ruido} = \sqrt{\frac{\sigma^2}{2}} \cdot \text{randn}(1, \text{Longitud señal GPS}) + j \sqrt{\frac{\sigma^2}{2}} \cdot \text{randn}(1, \text{Longitud señal GPS})$$

²² El ruido *flicker*, también llamado ruido rosa, es un tipo de ruido caracterizado por una densidad espectral inversamente proporcional a la frecuencia.

²³ Nos referimos al concepto *jitter* para explicar las fluctuaciones, sobre su valor ideal, que experimentan los dos parámetros que debe seguir el receptor GPS.

²⁴ AWGN son las siglas en inglés de *Additive White Gaussian Noise*.

Como hasta ahora, consideraremos que la potencia de la señal GPS P_S es fija e igual a la unidad. Las simulaciones de este apartado se generarán en base a cuatro valores distintos de potencia de ruido P_N , representada por su varianza σ^2 . Por tanto, referenciaremos los cuatro casos de ruido a partir de la SNR que experimenta el receptor:

Recordando la expresión de la potencia del ruido AWGN:

$$P_N = \sigma^2$$

Y su relación lineal con la SNR :

$$SNR = \frac{P_S}{P_N}$$

Calcularemos la relación señal a ruido para $\sigma^2 = 0, 10, 100$ y 500 expresada en dB:

$$SNR_1 = 10 \cdot \log_{10} \left(\frac{P_S}{P_N} \right) = 10 \cdot \log_{10} \left(\frac{1}{0} \right) = \infty$$

$$SNR_2 = 10 \cdot \log_{10} \left(\frac{P_S}{P_N} \right) = 10 \cdot \log_{10} \left(\frac{1}{10} \right) = -10 \text{ dB}$$

$$SNR_3 = 10 \cdot \log_{10} \left(\frac{P_S}{P_N} \right) = 10 \cdot \log_{10} \left(\frac{1}{100} \right) = -20 \text{ dB}$$

$$SNR_4 = 10 \cdot \log_{10} \left(\frac{P_S}{P_N} \right) = 10 \cdot \log_{10} \left(\frac{1}{500} \right) \approx -27 \text{ dB}$$

Por tanto, introduciremos el ruido en el sistema de la siguiente manera: primero implementamos la señal GPS, y posteriormente generamos el ruido de manera independiente. Una vez generadas ambas señales, se suman muestra a muestra. Habiendo introducido brevemente el ruido, su modelo y los valores utilizados, realizaremos varias simulaciones para observar los efectos que éste produce sobre los cálculos de los parámetros del receptor GPS.

6.4.2 Simulación de matriz de correlación

En esta primera simulación variaremos los valores de varianza para observar los cambios en los valores de la matriz de correlación de la adquisición. Trabajaremos con la señal GPS simulada, y con la *adquisición serial* correspondiente a dos períodos de la señal GPS y un periodo de la señal réplica. La siguiente figura (Fig. 6.32) representa la matriz de correlación para los valores de *SNR* anteriormente especificados:

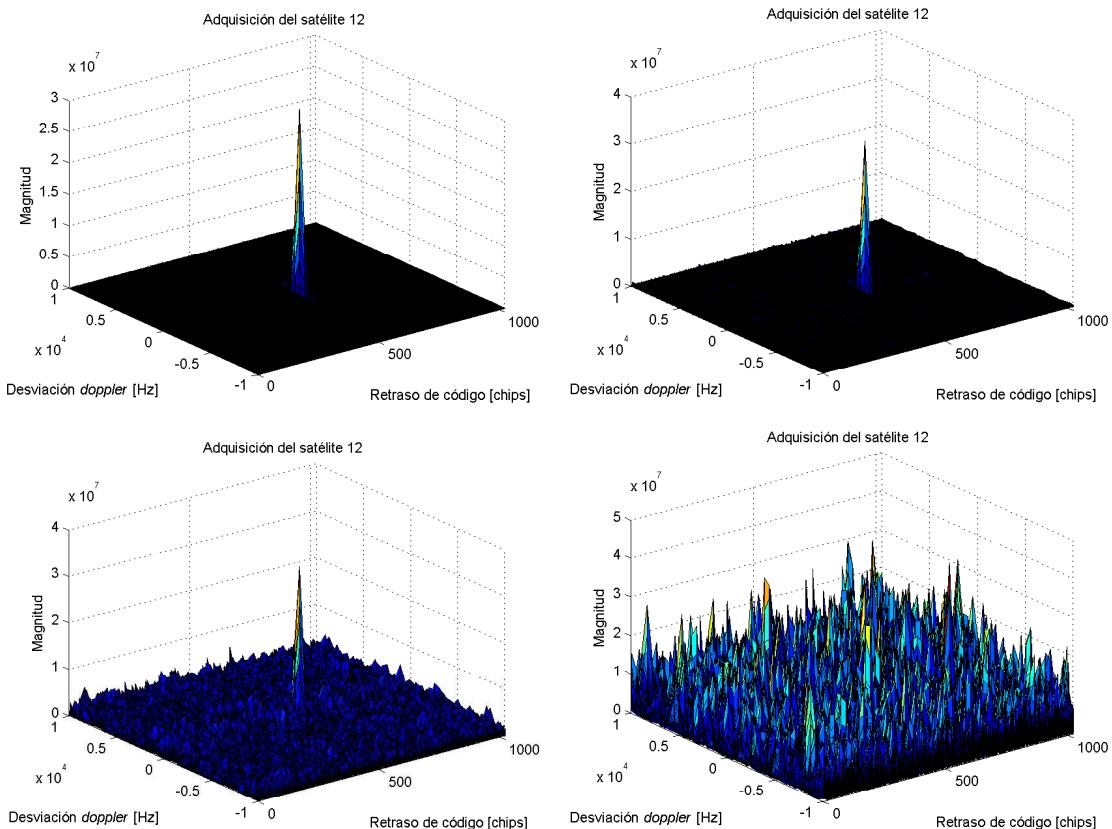


Fig. 6.32. Matriz de adquisición. De izquierda a derecha y de arriba a abajo, con valores de *SNR* de ∞ , -10 , -20 , y -27 dB

Observamos cómo a medida que aumentamos la potencia de ruido, aumenta también la correlación para los valores no deseados de retraso de código y desviación Doppler. Por tanto, si seguimos aumentando la varianza, corremos el riesgo de provocar que el máximo no sea el deseado, pero aún y así supere el umbral de detección de la adquisición y el receptor entregue al módulo de tracking una mala estimación de los parámetros. Esto provocará que el módulo de tracking tarde más en converger, y hará del receptor un sistema menos eficiente.

6.4.3 Simulación de envolventes

En la segunda simulación variaremos la potencia de ruido para analizar su efecto sobre el triángulo de correlaciones. En la siguiente figura (Fig. 6.33), observamos cómo la introducción de ruido provoca la aparición del *jiiter* mencionado, que podemos ver en la variación de los valores de correlación de las envolventes con una $SNR = -10$ dB.

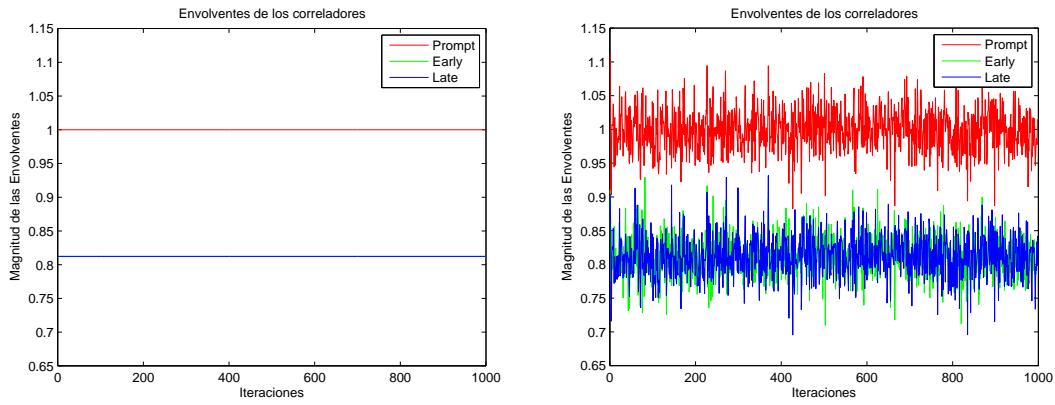


Fig. 6.33. Magnitud de las envolventes sin ruido (izquierda), y para una $SNR = -10$ dB (derecha)

Esto provoca que, si nos centramos en una iteración y representamos el triángulo de correlaciones (ver subapartado 5.4.1.7), las envolventes de los diferentes códigos cambien también de valor y ya no sean las ideales (Fig. 6.34):

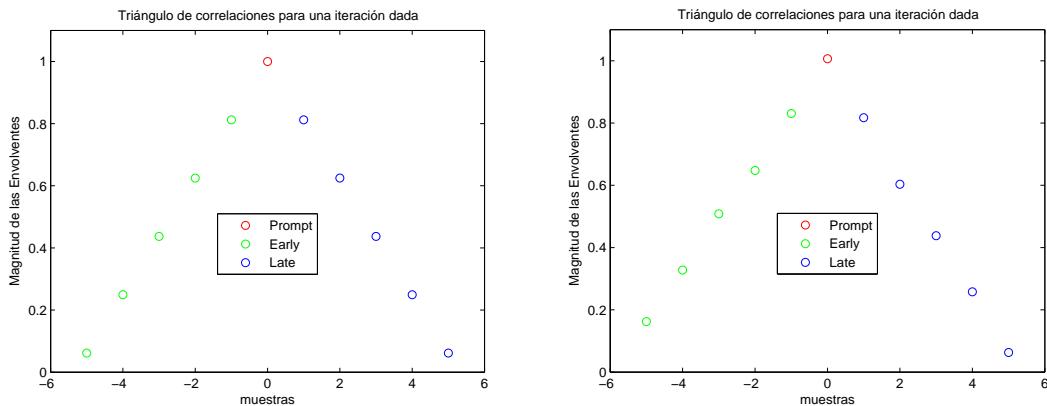


Fig. 6.34. Triángulo de correlaciones sin ruido (izquierda), y para una $SNR = -10$ dB (derecha)

Aunque se aprecie poco (Fig. 6.34), observamos como el primer y el último valor azul, correspondientes a desplazamientos de 1 y 5 muestras sobre el código *prompt*, tienen un valor menor al ideal. Si aumentamos la potencia de ruido hasta alcanzar los valores

de SNR de -20 y -27 dB observamos como el valor del *jitter* se hace más patente en los valores del triángulo de correlaciones (Fig. 6.35):

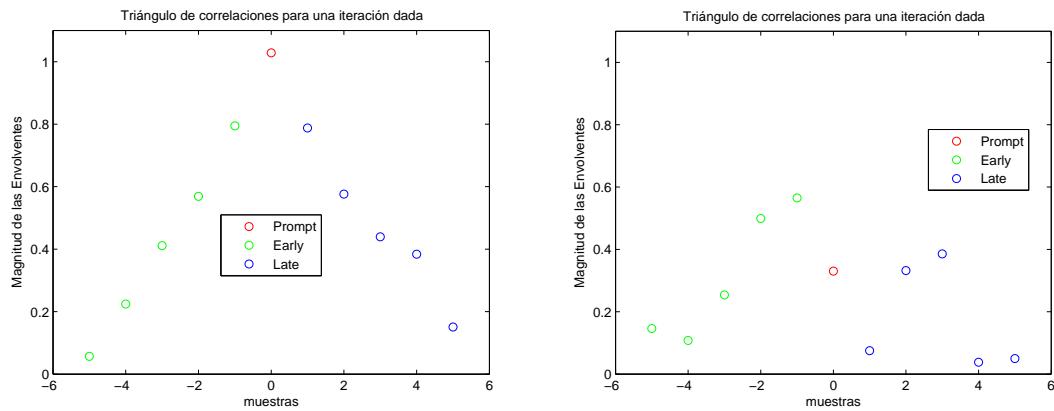


Fig. 6.35. Triángulo de correlaciones para una $SNR = -20$ dB (izquierda), y para una $SNR = -27$ dB (derecha)

Con una $SNR = -20$ dB, el comportamiento de las envolventes todavía es razonable, pero una $SNR = -27$ dB desvirtúa totalmente el sistema.

6.4.4 Simulación de fase estimada

En esta tercera y última simulación del ruido variaremos entre los cuatro valores anteriores de potencia de ruido, y obtendremos la fase que mide el PLL. Únicamente estamos simulando el ruido, así que no introduciremos desviación Doppler a la fase, con lo que la fase que debería estimar el PLL será la entrante de 45° (Fig. 6.36):

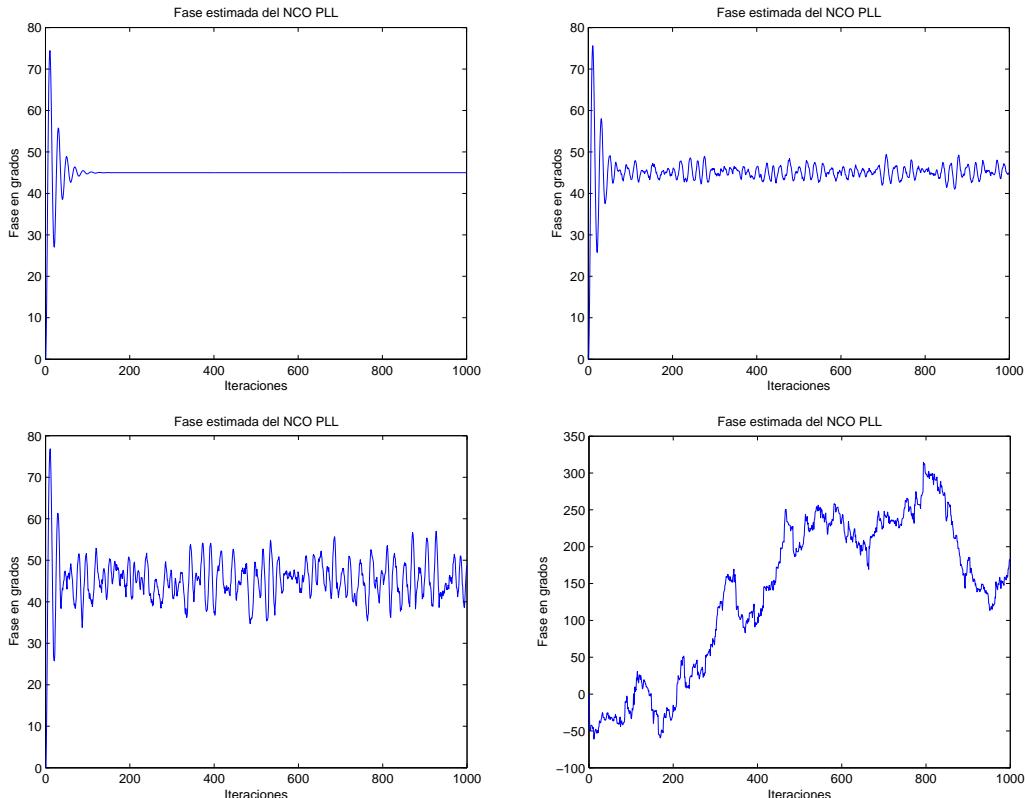


Fig. 6.36. Estimación de fase del PLL. De izquierda a derecha y de arriba a abajo, con valores de SNR de $\infty, -10, -20$, y -27 dB

Observamos cómo, sin ruido, la fase estimada que devuelve el PLL converge en poco más de 100 iteraciones al valor ideal especificado. Cuando la SNR alcanza los valores -10 dB y -20 dB, el PLL no converge pero se intuye un rango de valores en la estimación, y a medida que aumentamos la potencia de ruido, el *jitter* también aumenta. En las anteriores simulaciones hemos visto que para una $SNR = -27$ dB el sistema quedaba desvirtuado, y este fenómeno se puede apreciar en la medida de la fase de la última imagen de la Fig. 6.36, donde ni el PLL converge ni se acerca al valor real.

Escalando por los valores adecuados, la estimación de la fase devolverá el observable de fase en metros, tal y como hemos visto en el apartado 6.1.5.

6.5 Efecto combinado

Una vez estudiados los anteriores efectos por separado, en esta última sección del capítulo los combinaremos para analizar cómo se comporta un Sistema GPS en condiciones reales, es decir, cuando sus propios componentes electrónicos generan ruido, existe *multipath*, todos los componentes de la señal GPS están afectados por una desviación Doppler, y el receptor tiene un ancho de banda limitado debido al filtrado.

Para simular el efecto multicamino, generaremos la señal directa y la réplica, ambas en banda base. Antes de modularlas a la frecuencia residual Doppler, las sumaremos muestra a muestra y les añadiremos ruido una vez sumadas. Más tarde, aplicaremos el filtrado para finalmente introducir la frecuencia Doppler e iniciar el proceso de adquisición.

A continuación realizaremos diversas simulaciones, haciendo hincapié no tanto en los valores utilizados para generarlas, cuyo estudio se ha hecho en las anteriores secciones del capítulo, sino en el conocimiento que podemos extraer de ellas.

En esta primera simulación hemos generado una señal réplica con un retraso de código diferente al de la señal directa para ver si realmente la adquisición generaba otro pico de correlación de menor magnitud que el original (Fig. 6.37).

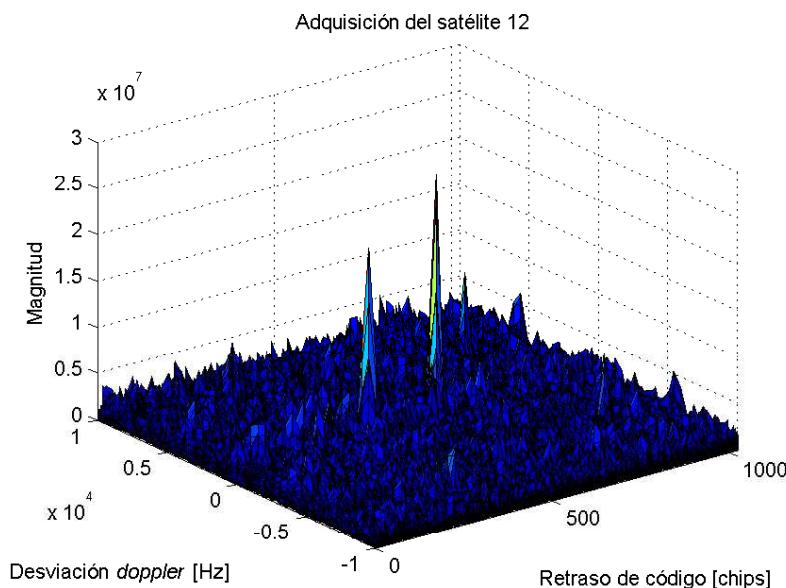


Fig. 6.37. Matriz de correlación para simular el efecto multicamino

Observamos cómo el comportamiento es el esperado y, además, el receptor consigue adquirir los parámetros de retraso de código y desviación Doppler correctamente, ya que el pico de correlación de mayor magnitud corresponde al de la señal directa, y ambos parámetros serán entregados al tracking para que realice su seguimiento.

El módulo de tracking tendrá más problemas debido al *jitter* y al efecto multicamino aún presente en la señal GPS. Si bien los parámetros recibidos de retraso de código y desviación Doppler son los correctos para el instante inicial, su seguimiento se verá condicionado por los anteriores efectos mencionados.

Cuando la señal GPS era ideal y no considerábamos ningún efecto adicional, al módulo de tracking le era más fácil trabajar, ya que los valores de las envolventes eran los ideales y constantes a lo largo del tiempo (Fig. 5.27). Este hecho provocaba que la salida del discriminador DLL se anulara y, de esta manera, el código *prompt* estaba permanentemente alineado con el código C/A de la señal GPS entrante. Ahora esto ya no es así, y los valores de las envolventes se ven modificados en cada iteración para intentar ajustarse a la señal GPS (Fig. 6.38).

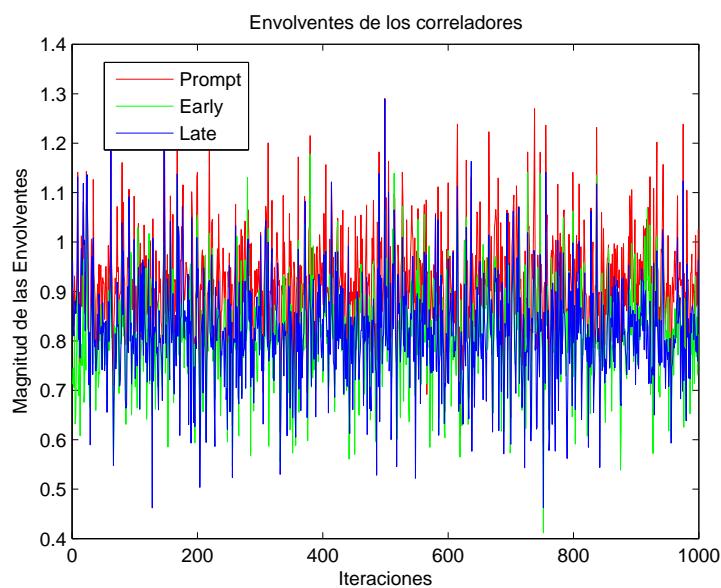


Fig. 6.38. Magnitud de las envolventes de los correladores

Aunque los valores de las envolventes experimentan un cierto *jitter*, se puede observar que la envolvente del código *prompt* está razonablemente por encima de la de los códigos desplazados una muestra a izquierda y derecha (Fig. 6.38), tal y como podemos observar en el triángulo de correlaciones del DLL para una iteración dada (Fig. 6.39).

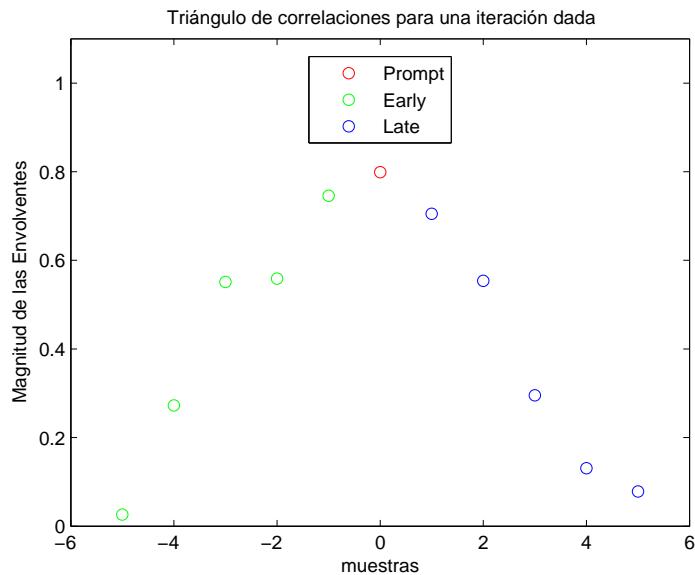


Fig. 6.39. Triángulo de correlación de las envolventes

Como hemos comentado, la variabilidad de las envolventes provoca que el discriminador DLL tenga que reajustarse continuamente (imagen izquierda Fig. 6.40), con lo que los valores del filtro de lazo también se ajustan acorde a su entrada (imagen derecha Fig. 6.40).

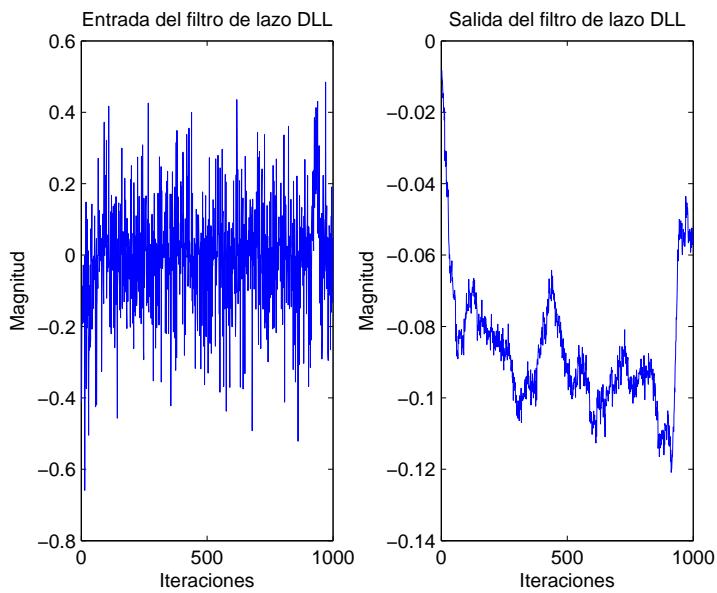


Fig. 6.40. Salida del discriminador y salida del filtro de lazo DLL

Los valores del filtro de lazo serán acumulados para generar el NCO_{DLL} , y este retardo será el que iteración a iteración se aplicará sobre el remuestreo del código *prompt*. Por su parte, la entrada al discriminador PLL, como hemos visto en la Fig. 6.38, no tendrá un

módulo unidad constante, y la fase de la señal GPS será mucho más difícil de estimar, ya que el error de fase (Fig. 6.41) oscila y no converge claramente a cero debido al *jitter* existente en la simulación.

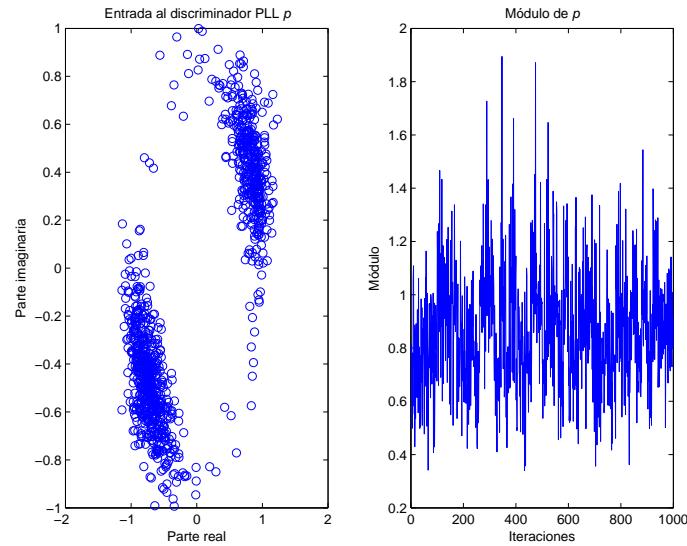


Fig. 6.41. Entrada al discriminador PLL

La salida del discriminador es entregada al filtro de lazo para que proceda con su suavizado y, finalmente, estos valores son los encargados de generar la fase NCO_{PLL} (Fig. 6.42), que el generador de exponencial aplicará para estimar la fase de la señal GPS:

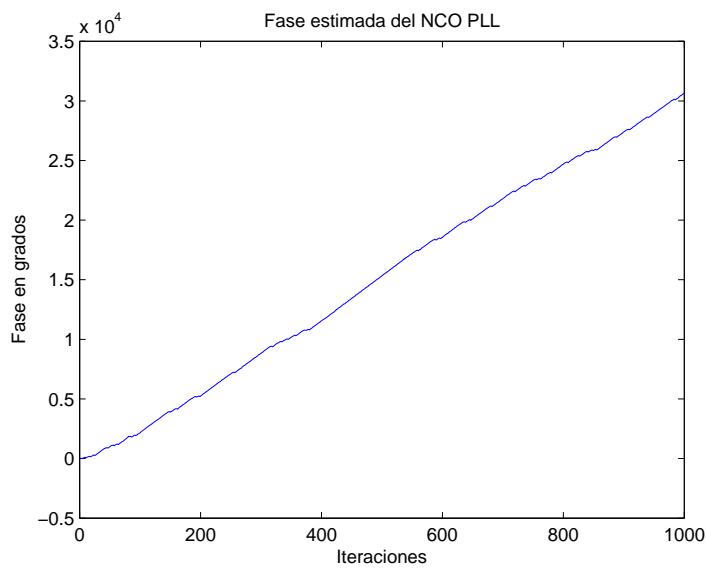


Fig. 6.42. Fase estimada por el NCO_{PLL}

Tanto la señal directa como la réplica han sido generadas con una desviación Doppler producida por una velocidad constante de 20 m/s. Por tanto, la desviación Doppler real viene determinada por la siguiente expresión:

$$f_{dop_{REAL}} = f_{L1} \frac{\nu}{c} = 1575.42 \cdot 10^6 \frac{20}{299792458} = 105.1 \text{ Hz}$$

Observando los valores de la figura Fig. 6.42, podemos calcular la frecuencia Doppler estimada:

$$f_{dop_{ESTIMADA}} = \frac{1}{2\pi} \cdot \frac{\text{NCO}_{PLL}(1000)_{rad} - \text{NCO}_{PLL}(0)_{rad}}{1 - 0}$$

$$f_{dop_{ESTIMADA}} = \frac{1}{2\pi} \cdot \frac{\frac{\pi}{180} (30530.8 - 0)}{1 - 0} = 84.8 \text{ Hz}$$

Observamos que los valores real y estimado de la desviación Doppler distan demasiado, y esto se refleja a la hora de estimar el observable de fase en metros entre satélite y receptor (Fig. 6.43), en la que en un segundo de integración se comete un error de cuatro metros. Por tanto, tenemos un 20% de error en la medida de la distancia.

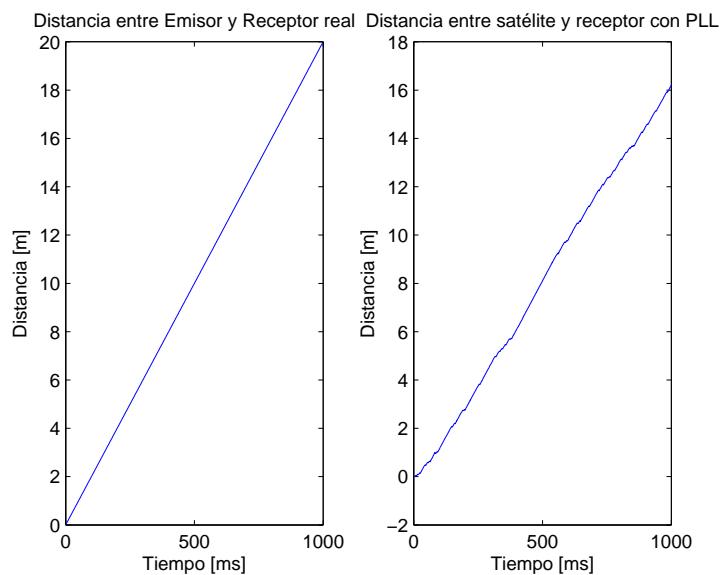


Fig. 6.43. Variación real de distancia (izquierda), variación con el PLL (derecha)

Aquí concluimos el último de los capítulos que analiza el funcionamiento del Sistema GPS y del receptor. Hemos presentado la manera de estimar y refinar los parámetros retraso de código y desviación Doppler de la señal GPS, y los hemos utilizado para analizar cómo se comporta el receptor GPS de manera ideal y bajo efectos reales. El siguiente capítulo presentará las conclusiones finales y las áreas de mejora y de desarrollo futuro. Finalmente, el proyecto acabará con un apartado de Anexo y de Bibliografía, además de un breve resumen del proyecto realizado.

7. Conclusiones

El último capítulo numerado del documento presentará las conclusiones a las que se han llegado después de realizar el proyecto, se analizará si se han cumplido los objetivos propuestos al comienzo, se trazarán posibles líneas de investigación futura, y se mostrarán posibles áreas de mejora dentro del diseño del receptor.

Como comentamos al inicio, el principal objetivo del proyecto era entender el funcionamiento básico de un receptor GPS. Podemos decir que hemos cumplido con el objetivo porque hemos implementado una versión básica de señal GPS, hemos simulado en ella efectos reales, y hemos analizado las señales y funciones que tiene cada uno de los módulos del receptor. Si bien nos hemos centrado en las etapas de adquisición y tracking, hemos sentado las bases de los módulos de *front-end* y de navegación, entendiendo la función de cada uno y cómo realizaría sus funciones en caso de que en un futuro decidamos abordarlos en más profundidad.

El objetivo principal se ha podido conseguir porque hemos adquirido el retraso de código y la desviación Doppler en la etapa de adquisición, hemos realizado su seguimiento en el módulo de tracking, y es precisamente este seguimiento el que nos ha permitido calcular el pseudorrango y el observable de fase entre satélite y receptor, parámetros que se envían al módulo de navegación para que sea él el que estime la posición del propio receptor.

Existen múltiples líneas de investigación a partir de la realización de este proyecto: desde estudiar cómo tratar el efecto multicamino en el remuestreo del código *prompt* del DLL, hasta profundizar y resolver las ecuaciones de navegación para posicionar un receptor GPS. Si bien se han presentado algunos efectos reales que sufre la señal GPS, no se han estudiado con todo lujo de detalles, y ese sería un aspecto importante para ampliar el conocimiento sobre el receptor.

Dentro de las áreas de mejora en el diseño del receptor destacaría la teoría que hay detrás de las constantes de los filtros de lazo. En este proyecto se han aplicado a modo de prueba y error y, aunque se ha realizado una breve lectura del tema, no han sido exactamente las mismas para cada simulación. Otro aspecto algo más específico es que, a igualdad de valores en la matriz de correlación de adquisición cuando existe una señal réplica igual a la directa, los máximos que devuelve el algoritmo de adquisición son los de posición más baja: es decir, para los valores de retraso de código y desviación Doppler menores. Un último punto a mejorar es conocer las técnicas que controlan los efectos que sufre la señal GPS y poder hacer un mejor cálculo del observable de fase.

Personalmente, considero que ha sido un primer contacto positivo con las comunicaciones por satélite, un punto de partida para poder desarrollar proyectos más específicos y de mayor envergadura en el futuro.

Anexo Código MATLAB®

```

function main_ideal

% Esta es la función en la que generaremos la señal GPS en banda base
% muestreada y filtrada, y que servirá de referencia para llamar a las
% funciones de generación de código C/A, adquisición y tracking. En esta
% función solo tendremos en cuenta la frecuencia Doppler en la portadora,
% no así en el código C/A y el mensaje de navegación.

Nchips=1023;
fi=0;
fm=5.456e6; %frecuencia de muestro muestras/s
tm=1/fm; %tiempo de muestreo
tInc=1e-3; tInc segundos de Incoming
tCA=2; % tCA milisegundos del Código CA (PRN)
tSAT=1e3; % tCA milisegundos del Código CA (PRN)
fchip=1.023e6;
tchip=1/fchip; %s/chip

muestrasInc=round(fm*tInc); % muestrasInc muestras de Incoming
muestrasChip=fm*tchip; % Cada chip de Incoming tiene muestrasChip
muestras
muestras1msCAlocal=round(Nchips*1*muestrasChip) %muestras de 1ms del
CAlocal
muestrasCAlocal=round(Nchips*tCA*muestrasChip) %muestras del CAlocal

%-----
% Generación de la Frecuencia Doppler.
fL1=154*10.23e6; %Portadora
c=299792458; %velocidad de la luz en [m/s]
v=0; %velocidad asociada a la frecuencia Doppler [m/s]
doppler=v*fL1/c; %frecuencia Doppler [Hz]

% Opción 1. %Frecuencia Doppler constante en todas las muestras
fdop(1:Nchips*tSAT*muestrasChip)=doppler;

% Opción 2. %Frecuencia Doppler Variable. La primera mitad tiene
% Doppler > 0 y la otra mitad tienen Doppler < 0.
fdop(1:Nchips*tSAT*muestrasChip/2)=doppler;
fdop((Nchips*tSAT*muestrasChip/2)+1:Nchips*tSAT*muestrasChip)=-doppler;
%-----

%-----
% Generación de la señal GPS muestreada en banda base
PRNSAT=12; % Código PRN asociado al satélite
CASAT=cacode(PRNSAT); % Código C/A de 1023 chips y 1023 muestras
var=0; % varianza de ruido

a=(0:muestrasChip*Nchips*tSAT-1)*(1/muestrasChip);
b=floor(a);
CASAT=CASAT(mod(b,Nchips)+1); % Código C/A remuestreado del satélite.
Long1msCA=Nchips*muestrasChip;

```

```

Long20msCA=20*Long1msCA;

for i=1:tSAT/20
q=(i-1)*Long20msCA;
p=q+1:1:q+Long20msCA;

CA20ms=CASAT(p);

x(i)=randi([0,1],1,1);
bit20ms=floor(x(i) +(x(i)-x(i))*rand(1,Long20msCA));
bit20ms=2*bit20ms-1;

IncomingBB(p)=CA20ms.*bit20ms;
ruido=sqrt(var/2).*randn(1,Long20msCA)+1j*sqrt(var/2).*randn(1,Long20msCA);
IncomingBB(p)=IncomingBB(p)+ruido;
end

codephaseSAT=3000; % Retraso de código impuesto para la simulación
IncomingBB=circshift(IncomingBB,[0 codephaseSAT]); % Señal GPS en BB
fase=pi/4; % Fase de la señal IncomingBB en BB
%-----

% Espectro de la señal GPS en banda base
Fourier1=fftshift(fft(IncomingBB));
Modulol=abs(Fourier1);
f=(0:muestras1msCAlocal*tSAT-1)-muestras1msCAlocal*tSAT/2;
plot(f,Modulol,'b'); % Espectro de la señal GPS en banda base

% Filtro paso bajo
w=(2e6)/(fm/2);
[B,A]=cheby2(3,20,w,'low');
IncomingBB=filter(B,A,IncomingBB); % Señal GPS filtrada

% Espectro de la señal GPS filtrada en banda base
Fourier2=fftshift(fft(IncomingBB));
Modulo2=abs(Fourier2);
plot(f,Modulo2,'b'); % Espectro de la señal GPS filtrada en banda base

% Modulación a la frecuencia Doppler residual
n=0:Nchips*tSAT*muestrasChip-1;
Incoming=IncomingBB.*exp(+1j*(2*pi*(tm*fdop.*n)+ fase));
IncomingtrackPB=Incoming;

%-----
% Llamada de funciones en el receptor para procesar la señal GPS
PRN=12; % Código PRN asociado que intentará adquirir el receptor
CA=cacode(PRN);
graficas(Long20msCA,Long1msCA,muestras1msCAlocal,CA,muestrasChip,Nchips);
[CAlocal codephase freq carrier n a
b]=adq_1Inc2CA(Nchips,fi,fm,tm,fchip,tchip,tInc,tCA,muestrasInc,muestrasChip,muestras1msCAlocal,muestrasCAlocal,Incoming,PRN,CA);
[CAlocal codephase freq carrier n a
b]=adq_2Inc1CA(Nchips,fi,fm,tm,fchip,tchip,tInc,tCA,muestrasInc,muestrasChip,muestras1msCAlocal,muestrasCAlocal,Incoming,PRN,CA);

```

```

track_ideal2(tSAT,Nchips,fase,fi,fdop,c,v,fm,tm,fchip,tchip,tInc,tCA,m
uestasInc,muestrasChip,muestras1msCALocal,muestrasCALocal,Incomingtra
ckPB,IncomingBB,n,PRN,CALocal,codephase,freq,CA,a,b,muestraref,fL1);
%-----

function
graficas(Long20msCA,Long1msCA,muestras1msCALocal,CA,muestrasChip,Nchip
s)

% Esta función genera las gráficas del capítulo 4: señal temporal y
% espectro del mensaje de navegación, así como las correlaciones del
% código C/A y su espectro.

% Entradas:
% Long20msCA = Longitud de 20 periodos de CALocal
% Long1msCA = Longitud de 1 periodo de CALocal
% muestras1msCALocal = #[muestras] en un periodo de CALocal
% CA = 1023 muestras del código C/A asociado al satélite PRN
% muestrasChip = #[muestras] en cada chip
% Nchips = 1023 [chips]

%-----
% Mensaje de navegación

t=2000; %ms

for i=0:99
x(i+1)=randi([0,1],1,1); % x=[0,1]
bit20ms=floor(x(i+1) +(x(i+1)-x(i+1))*rand(1,Long20msCA)); % Si x=0 ->
bit20ms=[0 0 .... 0]. Si x=1 -> bit20ms=[1 1 .... 1].
bit20ms=2*bit20ms-1; %Para pasar de [0,1] a [-1,1]
nav(Long20msCA*i+1:Long20msCA*(i+1))=bit20ms;
end

aa=0:1/Long1msCA:2000-1/Long1msCA;
figure (1)
subplot(2,2,1)
plot(aa,nav) %Señal temporal del mensaje de navegación
axis([0 200 -1.5 1.5])
title('Mensaje de navegación')
xlabel('Tiempo [ms]')
ylabel('Bit de navegación')

Fourier1=fftshift(fft(nav));
Modulol1=abs(Fourier1);
f=(-muestras1msCALocal*1000:muestras1msCALocal*1000-1)*1/2;
subplot(2,2,2)
plot(f,Modulol1,'b'); %Espectro del mensaje de navegación
axis([-500 500 0 3e6])
title('Espectro del mensaje de navegación REAL')
xlabel('Frecuencia [Hz]')
ylabel('Magnitud')
%-----

```

```
%-----
% Código C/A
rl=conv(CA,fliplr(CA));
freq_axis=-1+1/1023:1/1023:1-1/1023;
subplot (2,2,1)
plot(freq_axis,rl);
title('Autocorrelación del código C/A del satélite 12')
xlabel('Retardo normalizado')
ylabel('Magnitud')

a1=(0:muestrasChip*Nchips*1-1)*(1/muestrasChip);
b1=floor(a1);
CAlocal=CA(mod(b1,Nchips)+1);

Fourier1=fftshift(fft(CAlocal));
Modulol=abs(Fourier1);
f=(-Nchips*muestrasChip/2:Nchips*muestrasChip/2-1);
subplot (2,2,2)
plot(f,Fourier1, 'b');
title('Espectro del Código C/A ')
xlabel('Frecuencia [Hz]')
ylabel('Magnitud')
axis([-500 500 0 500])
-----
```

```
function CA=cacode(PRN)

% Esta función devolverá en la variable CA, el código C/A de 1023 chips
% y 1023 muestras en BPSK asociado al código PRN que representa el
% satélite.

% Entradas:
% PRN = Código asociado al satélite
% Salidas:
% CA = 1023 muestras del código C/A asociado al satélite PRN

tap=[2 6;
      3 7;
      4 8;
      5 9;
      1 9;
      2 10;
      1 8;
      2 9;
      3 10;
      2 3;
      3 4;
      5 6;
      6 7;
      7 8;
      8 9;
      9 10;
      1 4;
      2 5;
      3 6;
      4 7;
```

```

5 8;
6 9;
1 3;
4 6;
5 7;
6 8;
7 9;
8 10;
1 6;
2 7;
3 8;
4 9
5 10
4 10
1 7
2 8
4 10];

Nchips=1023; % Longitud código C/A
Ndes=10; %bits desplazamiento

v1=ones(1,Ndes);
v2=ones(1,Ndes);

for i=1:Nchips

    G1(i)=v1(Ndes);
    bit1_v1=mod(v1(3)+v1(Ndes),2);
    v1(2:Ndes)=v1(1:Ndes-1);
    v1(1)=bit1_v1;

    G2(i)=mod(v2(tap(PRN))+v2(tap(37+PRN)),2);
    bit1_v2=mod(v2(2)+v2(3)+v2(6)+v2(8)+v2(9)+v2(Ndes),2);
    v2(2:Ndes)=v2(1:Ndes-1);
    v2(1)=bit1_v2;

    CA(i)=mod(G1(i)+G2(i),2);
end

CA=2*CA-1;
end

function [CALocal codephase freq carrier n a
b]=adq_1Inc2CA(Nchips,fi,fm,tm,fchip,tchip,tInc,tCA,muestrasInc,muestr
asChip,muestras1msCALocal,muestrasCALocal,Incoming,PRN,CA)

% Esta es la primera de las dos funciones de adquisición serial. Trabaja
% con 1 periodo de la señal GPS y con 2 del código C/A del receptor, la
% señal réplica.

% Entradas:
% Nchips = 1023 [chips]
% fi = Frecuencia intermedia [Hz]
% fm = Frecuencia de muestreo [Hz]

```

```
% tm = Tiempo de muestreo [s]
% fchip = Frecuencia de chip [Hz]
% tchip = Tiempo de chip [s]
% tInc = [s] de la señal GPS
% tCA = [ms] del código C/A
% muestrasInc = #[muestras] de la señal GPS
% muestrasChip = #[muestras] en cada chip
% muestras1msCAlocal = #[muestras] en un periodo de CAlocal
% muestrasCAlocal = #[muestras] de CAlocal
% Incoming = Señal GPS modulada
% PRN = Código asociado al satélite
% CA = 1023 muestras del código C/A asociado al satélite PRN

% Salidas:
% CAlocal = Código C/A remuestreado
% codephase = Retraso de código adquirido
% freq = Frecuencia Doppler adquirida
% carrier = Señal GPS con la frecuencia Doppler extraída
% n = Índice de muestras
% a = Vector de muestras que representa los chips
% b = Parte entera del vector anterior

a=(0:muestrasChip*Nchips*tCA-1)*(1/muestrasChip);
b=floor(a);
CAlocal=CA(mod(b,Nchips)+1);

Incoming=Incoming(1:muestrasInc);

n=0:muestrasInc-1;
offset=0;

for fi=fi-1e4:5e2:fi+1e4
    carrier=Incoming.*exp(-j*2*pi*f*tm*n);
    matcor(offset+1,:)=conv(carrier,fliplr(CAlocal));
    offset=offset+1;
end

matcor=(abs(matcor).^2);

%Búsqueda de los máximos de la matriz
[num idx]=max(matcor(:));
[x y]=ind2sub(size(matcor),idx);
freq=(fi-1e4)+(x-1)*5e2
codephase=y-muestras1msCAlocal

% Representación de la matriz de correlación
filas=(fi-1e4:5e2:fi+1e4);
q=(muestrasInc+muestrasCAlocal-1)/Nchips;
columnas=0:1/q:Nchips-1/q;
surf(columnas,filas,matcor)
xlim([0 1022])
ylim([fi-1e4 fi+1e4])
title('Adquisición del satélite 12')
xlabel('Retraso de código en chips')
ylabel('Desviación {\itdoppler} [Hz]')
zlabel('Magnitud')
end
```

```

function [CAlocal codephase freq carrier n a
b]=adq_2Inc1CA(Nchips,fi,fm,tm,fchip,tchip,tInc,tCA,muestrasInc,muestras
asChip,muestras1msCAlocal,muestrasCAlocal,Incoming,PRN,CA)

% Esta es la segunda de las dos funciones de adquisición serial. Trabaja
% con 2 periodos de la señal GPS y con 1 del código C/A del receptor,
% la señal réplica.

% Entradas:
% Nchips = 1023 [chips]
% fi = Frecuencia intermedia [Hz]
% fm = Frecuencia de muestreo [Hz]
% tm = Tiempo de muestreo [s]
% fchip = Frecuencia de chip [Hz]
% tchip = Tiempo de chip [s]
% tInc = [s] de la señal GPS
% tCA = [ms] del código C/A
% muestrasInc = #[muestras] de la señal GPS
% muestrasChip = #[muestras] en cada chip
% muestras1msCAlocal = #[muestras] en un periodo de CAlocal
% muestrasCAlocal = #[muestras] de CAlocal
% Incoming = Señal GPS modulada
% PRN = Código asociado al satélite
% CA = 1023 muestras del código C/A asociado al satélite PRN

% Salidas:
% CAlocal = Código C/A remuestreado
% codephase = Retraso de código adquirido
% freq = Frecuencia Doppler adquirida
% carrier = Señal GPS con la frecuencia Doppler extraída
% n = Índice de muestras
% a = Vector de muestras que representa los chips
% b = Parte entera del vector anterior

a=(0:muestrasChip*Nchips*1-1)*(1/muestrasChip);
b=floor(a);
CAlocal=CA(mod(b,Nchips)+1);

% fichero=fopen('GNS3v3_c3.bin','r'); %Señal GPS real
% Incoming=fread(fichero,10912,'schar').';

Incoming=Incoming(1:2*muestras1msCAlocal);

n=0:2*muestras1msCAlocal-1;
offset=0;
for f=fi-1e4:5e2:fi+1e4
    carrier=Incoming.*exp(-j*2*pi*f*tm*n);
    matcor(offset+1,:)=conv(carrier,fliplr(CAlocal));
    offset=offset+1;
end

matcor2=matcor(:,muestras1msCAlocal:2*muestras1msCAlocal);
matcor2=(abs(matcor2).^2);

%Búsqueda de los máximos de la matriz
[num idx]=max(matcor2(:));
[x y]=ind2sub(size(matcor2),idx);
freq=(fi-1e4)+(x-1)*5e2
codephase=y-1

```

```
% Representación de la matriz de correlación
filas=(fi-1e4:5e2:fi+1e4); q=(muestras1msCAlocal+1)/Nchips;
columnas=0:1/q:Nchips-1/q;
surf(columnas,filas,matcor2)
xlim([0 1022])
ylim([fi-1e4 fi+1e4])
title('Adquisición del satélite 12')
xlabel('Retraso de código [chips]')
ylabel('Desviación {\itdoppler} [Hz]')
zlabel('Magnitud')
end

function
track_ideal2(tSAT,Nchips,fase,fi,fdop,c,v,fm,tm,fchip,tchip,tInc,tCA,m
uestrasInc,muestrasChip,muestras1msCAlocal,muestrasCAlocal,Incomingtra
ckPB,IncomingBB,n,PRN,CAlocal,codephase,freq,CA,a,b,muestraref,fL1)

% Esta función realiza la etapa de tracking del receptor GPS

% Entradas:
% tSAT = [ms] de la señal GPS
% Nchips = 1023 [chips]
% fase = fase en [grados] de la señal GPS en BB
% fi = Frecuencia intermedia [Hz]
% fdop = vector de muestras que simula la frecuencia Doppler
% c = velocidad de la luz en [m/s]
% v = velocidad relativa entre satélite y receptor [m/s]
% fm = Frecuencia de muestreo [Hz]
% tm = Tiempo de muestreo [s]
% fchip = Frecuencia de chip [Hz]
% tchip = Tiempo de chip [s]
% tInc = [s] de la señal GPS
% tCA = [ms] del código C/A
% muestrasInc = #[muestras] de la señal GPS
% muestrasChip = #[muestras] en cada chip
% muestras1msCAlocal = #[muestras] en un periodo de CAlocal
% muestrasCAlocal = #[muestras] de CAlocal
% IncomingtrackPB = Señal GPS entrante al módulo de tracking
% IncomingBB = Señal GPS en BB
% n = Índice de muestras
% PRN = Código asociado al satélite
% CAlocal = Código C/A remuestreado
% codephase = Retraso de código adquirido
% freq = Frecuencia Doppler adquirida [Hz]
% CA = 1023 muestras del código C/A asociado al satélite PRN
% a = Vector de muestras que representa los chips
% b = Parte entera del vector anterior
% muestraref = Muestra de referencia para verificar el tracking
% fL1 = Frecuencia portadora [Hz]

k=1e-1;
k1=k;
k2=k;

m=0:Nchips*tSAT*muestrasChip-1;
IncomingDem=IncomingtrackPB;
```

```

iteraciones=tSAT;
IncomingDem=IncomingDem(codephase+1:end);

CAlocal=CAlocal(1:muestras1msCAlocal);

fasetrack=0;

xx=0;
xxx=0;
filPLL1=0;
filPLL2=0;

prompt=CAlocal;
early=circshift(prompt,[0 -round(muestrasChip/4)]);
late=circshift(prompt,[0 +round(muestrasChip/4)]);

early2=early;
early3=early;
early4=early;
early5=early;

late2=late;
late3=late;
late4=late;
late5=late;

NCODLL=0;

for i=1:iteraciones-1
g=(i-1)*muestras1msCAlocal;
l=g+1:1:g+muestras1msCAlocal;

IncomingDem5456=IncomingDem(1);
NCO=exp(-j*2*pi*(freq-fi)*tm*1);

input=IncomingDem5456.*NCO.*exp(-j*fazetrack(i));

pentrada(i)=sum(input.*prompt)/muestras1msCAlocal;
eentrada(i)=sum(input.*early)/muestras1msCAlocal;
lentrada(i)=sum(input.*late)/muestras1msCAlocal;

%Envolventes
pp(i)=abs(pentrada(i)).^2;
ee(i)=abs(eentrada(i)).^2;
ll(i)=abs(lentrada(i)).^2;

%Discriminador Coherente DLL
disDLL(i)=ee(i)-ll(i);

%Filtro DLL de orden 0
%filDLL1(i)=k*0.1*disDLL(i);

%Filtro DLL de orden 1
xxx=xxx+disDLL(i);
filDLL2(i)=k1*0.1*disDLL(i)+k2*0.1*xxx;

%NCO DLL.
NCODLL(i+1)=NCODLL(i)+filDLL2(i);

```

```
%Generador de secuencia. Remuestreo.
z1=(0:muestrasLmsCAlocal-1)*(1/muestrasChip)+NCODLL(i);
x1=floor(z1); %La parte entera

prompt=CA(mod(x1,Nchips)+1);

e1=circshift(x1,[0 -1]);
t1=circshift(x1,[0 +1]);

early=CA(mod(e1,Nchips)+1);
late=CA(mod(t1,Nchips)+1);

%-----
%Triángulo de correlaciones

e2=circshift(x1,[0 -2]);
t2=circshift(x1,[0 +2]);

early2=CA(mod(e2,Nchips)+1);
late2=CA(mod(t2,Nchips)+1);

e3=circshift(x1,[0 -3]);
t3=circshift(x1,[0 +3]);

early3=CA(mod(e3,Nchips)+1);
late3=CA(mod(t3,Nchips)+1);

e4=circshift(x1,[0 -4]);
t4=circshift(x1,[0 +4]);

early4=CA(mod(e4,Nchips)+1);
late4=CA(mod(t4,Nchips)+1);

e5=circshift(x1,[0 -5]);
t5=circshift(x1,[0 +5]);

early5=CA(mod(e5,Nchips)+1);
late5=CA(mod(t5,Nchips)+1);

eentrada2(i)=sum(input.*early2)/muestrasLmsCAlocal; %1x1
lentrada2(i)=sum(input.*late2)/muestrasLmsCAlocal; %1x1

eentrada3(i)=sum(input.*early3)/muestrasLmsCAlocal; %1x1
lentrada3(i)=sum(input.*late3)/muestrasLmsCAlocal; %1x1

eentrada4(i)=sum(input.*early4)/muestrasLmsCAlocal; %1x1
lentrada4(i)=sum(input.*late4)/muestrasLmsCAlocal; %1x1

eentrada5(i)=sum(input.*early5)/muestrasLmsCAlocal; %1x1
lentrada5(i)=sum(input.*late5)/muestrasLmsCAlocal; %1x1

ee2(i)=abs(eentrada2(i)).^2;
ll2(i)=abs(lentrada2(i)).^2;

ee3(i)=abs(eentrada3(i)).^2;
ll3(i)=abs(lentrada3(i)).^2;
```

```

ee4(i)=abs(eentrada4(i)).^2;
l14(i)=abs(lentrada4(i)).^2;

ee5(i)=abs(eentrada5(i)).^2;
l15(i)=abs(lentrada5(i)).^2;

%Fin de Triángulo
%-----

%1. Discriminador PLL.
disPLL(i)=real(pentrada(i))*imag(pentrada(i)); %

%2. Filtro PLL.
filPLL1(i)=k*disPLL(i);

xx=xx+disPLL(i);
filPLL2(i)=k1*disPLL(i)+k2*xx;

fasetrack(i+1)=fasetrack(i)+filPLL2(i);
end

figure (4)
subplot(4,3,1)
plot(sqrt(pp), 'r')
hold on
plot(sqrt(ee), 'g')
hold on
plot(sqrt(ll), 'b')
hold on
legend('Prompt','Early','Late', 'Location', 'best')
title('Envolventes de los correladores')
xlabel('Iteraciones')
ylabel('Magnitud de las Envolventes')

subplot(4,3,2)
plot(pentrada, 'ko')
title('Entrada al discriminador PLL \itp')
xlabel('Parte real')
ylabel('Parte imaginaria')

subplot(4,3,3)
plot(disDLL, 'k')
title('Error del discriminador DLL')
xlabel('Iteraciones')
ylabel('Magnitud')

subplot(4,3,4)
plot(disDLL, 'b')
title('Error del discriminador PLL. Entrada al filtro PLL de orden 0.
')
xlabel('Iteraciones')
ylabel('Magnitud')

subplot(4,3,5)
plot(filDLL2, 'b')
title('Salida al filtro PLL de orden 0')
xlabel('Iteraciones')

```

```

ylabel('Magnitud')

subplot(4,3,6)
NCODLL(1:iteraciones-1);
plot(NCODLL,'r')
title('Retardo acumulado del error por iteración')
xlabel('Iteraciones')
ylabel('Magnitud')

subplot(4,3,7)
plot(disPLL,'b')
title('Entrada filtro PLL de orden 1. Error del discriminador PLL')
xlabel('Iteraciones')
ylabel('Magnitud')

subplot(4,3,8)
plot(filPLL2,'k')
title('Salida filtro PLL de orden 1')
xlabel('Iteraciones')
ylabel('Magnitud')

subplot(4,3,9)
plot(fasetrack*180/pi,'b')
title('Fase estimada. NCO')
xlabel('Iteraciones')
ylabel('Fase en grados')

subplot(4,3,10)
distanciaPLL=(fasetrack/(2*pi))*c/fL1;
plot(distanciaPLL,'b');
title('Distancia entre Emisor y Receptor con PLL')
xlabel('Tiempo [ms]')
ylabel('Distancia [m]')

subplot(4,3,11)
tiempo=NCODLL/fchip; distanciaDLL=c*tiempo;
plot(distanciaDLL,'b');
title('Distancia entre Emisor y Receptor con DLL')
xlabel('Tiempo [ms]')
ylabel('Distancia [m]')

subplot(4,3,12)
it=1:tSAT;
kk=1;
for ii=0:muestras1msCAlocal:Nchips*tSAT*muestrasChip-
muestras1msCAlocal
vel(kk)=fdop(ii+1)*c/fL1;
kk=kk+1;
end

distanciaDatos=cumsum(vel.*1e-3); %suma acumulativa por posición
plot(distanciaDatos,'b');
title('Distancia entre Emisor y Receptor con mis datos')
xlabel('Tiempo [ms]')
ylabel('Distancia [m]')

```

```

figure (5)
subplot(1,2,1)
ejeP=(2:10:92)
ejeE=ejeP-1;
ejeL=ejeP+1;
plot(ejeP,pp(50:59), 'ro')
hold on
plot(ejeE,ee(50:59), 'go')
hold on
plot(ejeL,ll(50:59), 'bo')
title('Triángulo de correlaciones para 10 iteraciones consecutivas')
legend('Prompt','Early','Late', 'Location', 'best')

subplot(1,2,2)
iteracion=95;
ejePP=6;
ejeEE=1:5;
ejeLL=7:11;
ValoresEarly=[ee5(iteracion) ee4(iteracion) ee3(iteracion)
ee2(iteracion) ee(iteracion)];
ValoresLate=[ll(iteracion) ll2(iteracion) ll3(iteracion)
ll4(iteracion) ll5(iteracion)];
plot(ejePP,sqrt(pp(iteracion)), 'ro')
hold on
plot(ejeEE,sqrt(ValoresEarly), 'go')
hold on
plot(ejeLL,sqrt(ValoresLate), 'bo')
title('Triángulo de correlaciones para una iteración dada')
legend('Prompt','Early','Late', 'Location', 'best')

end

```

```

function main_fchip

% Esta es la función en la que generaremos la señal GPS en banda base
% muestreada y filtrada, y que servirá de referencia para llamar a las
% funciones de generación de código C/A, adquisición y tracking. En esta
% función tenemos en cuenta la frecuencia Doppler en toda la señal GPS.

Nchips=1023;
fi=0;
fm=5.456e6;
tm=1/fm;
tInc=1e-3;
tCA=2;
tSAT=1001;

fL1=154*10.23e6;
c=299792458;
v=0;
doppler=v*fL1/c;
fchip=1.023e6*(1+doppler/fL1);
tchip=1/fchip;

muestrasChip=fm*tchip;

```

```

muestrasInc=round(fm*tInc);
muestrasImsCALocal=round(Nchips*1*muestrasChip)
muestrasCALocal=round(Nchips*tCA*muestrasChip)

% Opción 1. %Frecuencia Doppler Constante en todas las muestras
fdop(1:Nchips*tSAT*muestrasChip)=doppler;

% Opción 2. %Frecuencia Doppler Variable. La primera mitad tiene
% Doppler > 0 y la otra mitad tienen Doppler < 0.
fdop(1:Nchips*tSAT*muestrasChip/2)=doppler;
fdop((Nchips*tSAT*muestrasChip/2)+1:Nchips*tSAT*muestrasChip)=-doppler;

% Opción 3. %Frecuencia Doppler Variable.
fdop(1:Nchips*tSAT*muestrasChip/4)=2*doppler;
fdop((Nchips*tSAT*muestrasChip/4)+1:Nchips*tSAT*muestrasChip/(4/3))=-doppler;
fdop((Nchips*tSAT*muestrasChip/(4/3))+1:Nchips*tSAT*muestrasChip)=doppler;

PRNSAT=12;
CASAT=cacode(PRNSAT); %1023

a=(0:muestrasChip*Nchips*tSAT-1)*(1/muestrasChip);
b=floor(a); %La parte entera
CASAT=CASAT(mod(b,Nchips)+1);

k=1;

for ii=1:length(a)
    if(a(ii)>=k*Nchips)
        cnt=ii-1;
        vectormuestrasporperiodoCAacum(k)=cnt;
        k=k+1;
    end
end
%En cada posición del vectormuestrasporperiodoCAacum está el número de
muestras acumuladas que hay en cada periodo del Código C/A

vectormuestrasporperiodoCA(1)=vectormuestrasporperiodoCAacum(1);
for kk=2:tSAT-1
    vectormuestrasporperiodoCA(kk)=vectormuestrasporperiodoCAacum(kk)-vectormuestrasporperiodoCAacum(kk-1);
end
%En cada posición del vectormuestrasporperiodoCA está el número de
muestras que hay en cada periodo del Código C/A

m=1;
resta=vectormuestrasporperiodoCAacum(20);
for qq=20:20:tSAT-1

CASAT20=CASAT(m:vectormuestrasporperiodoCAacum(qq));

x(qq/qq)=randi([0,1],1,1); % x=[0,1]

if qq>20

```

```

resta=vectormuestrasporperiodoCAacum(qq)-
vectormuestrasporperiodoCAacum(qq-20);
end

bit20CA=floor(x(qq/qq)+(x(qq/qq)-x(qq/qq))*rand(1,resta));
bit20CA=2*bit20CA-1;

IncomingBB(m:vectormuestrasporperiodoCAacum(qq))=CASAT20.*bit20CA;

m=vectormuestrasporperiodoCAacum(qq)+1;
end

var=500;
ruido=sqrt(var/2).*randn(1,length(IncomingBB))+1j*sqrt(var/2).*randn(1
,length(IncomingBB));
IncomingBB=IncomingBB+ruido;

codephaseSAT=3000;
IncomingBB=circshift(IncomingBB,[0 codephaseSAT]);
fase=45;

% Espectro de la señal GPS en banda base
Fourier1=fftshift(fft(IncomingBB));
Modulol=abs(Fourier1);
f=(0:muestras1msCALocal*tSAT-1)-muestras1msCALocal*tSAT/2;
plot(f,Modulol,'b'); % Espectro de la señal GPS en banda base

% Filtro paso bajo
w=(2e6)/(fm/2);
[B,A]=cheby2(3,20,w,'low');
IncomingBB=filter(B,A,IncomingBB); % Señal GPS filtrada

% Espectro de la señal GPS filtrada en banda base
Fourier2=fftshift(fft(IncomingBB));
Modulo2=abs(Fourier2);
plot(f,Modulo2,'b'); % Espectro de la señal GPS filtrada en banda base

n=0:length(IncomingBB)-1;
fdop=fdop(1:length(IncomingBB));
Incoming=IncomingBB.*exp(+1j*(2*pi*(tm*fdop.*n)+ fase));

%-----
% Multipath: Genero una IncomingM retrasada, desfasada, con una
% amplitud menor y diferente Doppler. Finalmente, hacemos la suma
% de la señal directa y la réplica generada para simular el multipath.

retardoM=5.333/2;
faseM=pi/4;
AmpM=0.9;
vM=20;
dopplerM=vM*fL1/c;

%Opción 1. %Frecuencia Doppler Constante en todas las muestras
fdopM(1:Nchips*tSAT*muestrasChip)=dopplerM;

```

```
%Opción 2. %Frecuencia Doppler Variable. La primera mitad tiene
Doppler > 0 y la otra mitad tienen Doppler < 0.
fdopM(1:Nchips*tSAT*muestrasChip/2)=dopplerM;
fdopM((Nchips*tSAT*muestrasChip/2)+1:Nchips*tSAT*muestrasChip)=-dopplerM;

PRNSATM=12;
CASATM=cacode(PRNSAT); varM=10;

aM=(0:muestrasChip*Nchips*tSAT-1)*(1/muestrasChip)+retardoM;
bM=floor(aM);
CASATM=CASATM(mod(bM,Nchips)+1);

kM=1;

for iiM=1:length(aM)
    if(aM(iiM)>=kM*Nchips)
        cntM=iiM-1;
        vectormuestrasporperiodoCAacumM(kM)=cntM;
        kM=kM+1;
    end
end
%En cada posición del vectormuestrasporperiodoCAacum está el número de
muestras acumuladas que hay en cada periodo del Código C/A

vectormuestrasporperiodoCAM(1)=vectormuestrasporperiodoCAacumM(1);
for kkM=2:tSAT-1
vectormuestrasporperiodoCAM(kkM)=vectormuestrasporperiodoCAacumM(kkM)-
vectormuestrasporperiodoCAacumM(kkM-1);
end
%En cada posición del vectormuestrasporperiodoCA está el número de
muestras que hay en cada periodo del Código C/A

mM=1;
restaM=vectormuestrasporperiodoCAacumM(20);
for qqM=20:20:tSAT-1

CASAT20M=CASATM(mM:vectormuestrasporperiodoCAacumM(qqM));

xM(qqM/qqM)=randi([0,1],1,1); % x=[0,1]

if qqM>20
restaM=vectormuestrasporperiodoCAacumM(qqM)-
vectormuestrasporperiodoCAacumM(qqM-20);
end

bit20CAM=floor(xM(qqM/qqM) +(x(qqM/qqM)-x(qqM/qqM))*rand(1,restaM)); %
bit20CAM=2*bit20CAM-1;

IncomingBBM(mM:vectormuestrasporperiodoCAacumM(qqM))=CASAT20M.*bit20CAM;

mM=vectormuestrasporperiodoCAacumM(qqM)+1;
end
```

```

nM=0:length(IncomingBBM)-1;
fdopM=fdopM(1:length(IncomingBBM));

codephaseSATM=2000;
IncomingBBM=circshift(IncomingBBM,[0 2000]);

IncomingM=AmpM.*IncomingBBM.*exp(+1j*(2*pi*(tm*fdopM.*nM)+ faseM));
Incoming=Incoming(1:length(IncomingM))+IncomingM;

var=100;
ruido=sqrt(var/2).*randn(1,length(Incoming))+1j*sqrt(var/2).*randn(1,length(Incoming));
Incoming=Incoming+ruido;
%-----



%-----%
% Llamada de funciones en el receptor para procesar la señal GPS
PRN=12;
CA=cacode(PRN);
[CALocal codephase freq carrier n a
b]=adq_1Inc2CA(Nchips,fi,fm,tm,fchip,tchip,tInc,tCA,muestrasInc,muestras
asChip,muestras1msCALocal,muestrasCALocal,Incoming,PRN,CA);
[CALocal codephase freq carrier n a
b]=adq_2Inc1CA(Nchips,fi,fm,tm,fchip,tchip,tInc,tCA,muestrasInc,muestras
asChip,muestras1msCALocal,muestrasCALocal,Incoming,PRN,CA);

track_fchip2(tSAT,Nchips,fase,fi,fdop,c,v,fm,tm,fchip,tchip,tInc,tCA,m
uestrasInc,muestrasChip,muestras1msCALocal,muestrasCALocal,Incomingtra
ckPB,IncomingBB,n,PRN,CALocal,codephase,freq,CA,a,b,muestraref,fL1,vec
tormuestrasporperiodoCAacum,vectormuestrasporperiodoCA,CASAT);
%-----



function
track_fchip2(tSAT,Nchips,fase,fi,fdop,c,v,fm,tm,fchip,tchip,tInc,tCA,m
uestrasInc,muestrasChip,muestras1msCALocal,muestrasCALocal,Incomingtra
ckPB,IncomingBB,n,PRN,CALocal,codephase,freq,CA,a,b,muestraref,fL1,vec
tormuestrasporperiodoCAacum,vectormuestrasporperiodoCA,CASAT)

% Esta función realiza parte de la etapa de tracking del receptor GPS,
% dejando la otra parte a la función bucletracking2.m, que es llamada
% desde esta misma función

% Entradas:
% tSAT = [ms] de la señal GPS
% Nchips = 1023 [chips]
% fase = fase en [grados] de la señal GPS en BB
% fi = Frecuencia intermedia [Hz]
% fdop = vector de muestras que simula la frecuencia Doppler
% c = velocidad de la luz en [m/s]
% v = velocidad relativa entre satélite y receptor [m/s]
% fm = Frecuencia de muestreo [Hz]
% tm = Tiempo de muestreo [s]
% fchip = Frecuencia de chip [Hz]
% tchip = Tiempo de chip [s]
% tInc = [s] de la señal GPS
% tCA = [ms] del código C/A

```

```
% muestrasInc = #[muestras] de la señal GPS
% muestrasChip = #[muestras] en cada chip
% muestras1msCAlocal = #[muestras] en un periodo de CAlocal
% muestrasCAlocal = #[muestras] de CAlocal
% IncomingtrackPB = Señal GPS entrante al módulo de tracking
% IncomingBB = Señal GPS en BB
% n = Índice de muestras
% PRN = Código asociado al satélite
% CAlocal = Código C/A remuestreado
% codephase = Retraso de código adquirido
% freq = Frecuencia Doppler adquirida [Hz]
% CA = 1023 muestras del código C/A asociado al satélite PRN
% a = Vector de muestras que representa los chips
% b = Parte entera del vector anterior
% muestraref = Muestra de referencia para verificar el tracking
% fL1 = Frecuencia portadora [Hz]
% vectormuestrasporperiodoCAacum = Vector que guarda las muestras por
% periodo del código C/A remuestreado de manera acumulada
% vectormuestrasporperiodoCA = Vector que guarda las muestras por
% periodo del código C/A remuestreado
% CASAT = Código C/A de la señal GPS

k=1e-1;
k1=10*k;
k2=10*k;

m=0:Nchips*tSAT*muestrasChip-1;
IncomingDem=IncomingtrackPB;
iteraciones=tSAT;
IncomingDem=IncomingDem(codephase+1:end);
CAlocaltrack=[CAlocal CAlocal(1:4)];

fasettrack=0;
xx=0;

prompt=CAlocaltrack;

early=circshift(prompt,[0 -round(muestrasChip/4)]);
late=circshift(prompt,[0 +round(muestrasChip/4)]);

early2=early;
early3=early;
early4=early;
early5=early;

late2=late;
late3=late;
late4=late;
late5=late;

acum=0;
xxx=0;
NCODLL=0;

aaa=(0:muestrasChip*Nchips*2-1)*(1/muestrasChip);

for i=1:iteraciones-2

    for ii=1:length(prompt)
```

```

if(aaa(ii)>=Nchips)

long=ii-1;
g=(i-1)*long;
lll=g+1:1:g+long;

[pp,ee,ee2,ee3,ee4,ee5,ll,112,113,114,115,disDLL,disPLL,filPLL1,filPLL
2,pentrada,fasetrack,NCODLL,filDLL1,filDLL2]=bucle_tracking2(tSAT,Nchi
ps,fase,fi,fdop,c,v,fm,tm,fchip,tchip,tInc,tCA,muestrasInc,muestrasChi
p,muestras1msCAlocal,muestrasCAlocal,IncomingtrackPB,IncomingBB,n,PRN,
CAlocal,codephase,freq,CA,a,b,muestraref,fL1,vectormuestrasporperiodoC
Aacum,vectormuestrasporperiodoCA,CASAT,prompt,early,late,k,k1,k2,Incom
ingDem,fasetrack,xx,early2,early3,early4,early5,late2,late3,late4,late
5,ii,i,CAlocaltrack,long,lll,xxx,NCODLL,acum);

break
end

end

figure (4)
subplot(4,3,1)
plot(pp,'r')
hold on
plot(ee,'g')
hold on
plot(ll,'b')
hold on
legend('Prompt','Early','Late', 'Location', 'best')
title('Envolventes de los correladores')
xlabel('Iteraciones')
ylabel('Magnitud de las Envolventes')

subplot(4,3,2)
plot(pentrada,'ko')
title('Entrada al discriminador PLL \itp')
xlabel('Parte real')
ylabel('Parte imaginaria')

subplot(4,3,3)
plot(disDLL,'k')
title('Error del discriminador DLL')
xlabel('Iteraciones')
ylabel('Magnitud')

subplot(4,3,4)
plot(disDLL,'b')
title('Entrada filtro PLL de orden 1. Error del discriminador PLL')
xlabel('Iteraciones')
ylabel('Magnitud')

subplot(4,3,5)
plot(filDLL2,'b')
title('Entrada filtro PLL de orden 1. Error del discriminador PLL')

```

```

xlabel('Iteraciones')
ylabel('Magnitud')

subplot(4,3,6)
NCODLL(1:iteraciones-2);
plot(NCODLL,'r')
title('Retardo acumulado del error por iteración')
xlabel('Iteraciones')
ylabel('Magnitud')


subplot(4,3,7)
plot(disPLL,'b')
title('Entrada filtro PLL de orden 1. Error del discriminador PLL')
xlabel('Iteraciones')
ylabel('Magnitud')


subplot(4,3,8)
plot(filPLL1,'k')
title('Salida filtro PLL de orden 1')
xlabel('Iteraciones')
ylabel('Magnitud')


subplot(4,3,9)
plot(fasetrack*180/pi,'b')
title('Fase estimada. NCO')
xlabel('Iteraciones')
ylabel('Fase en grados')


subplot(4,3,10)
distanciaPLL=(fasetrack/(2*pi))*c/fL1;
plot(distanciaPLL,'b');
title('Distancia entre Emisor y Receptor con PLL')
xlabel('Tiempo [ms]')
ylabel('Distancia [m]')


subplot(4,3,11)
tiempo=NCODLL/fchip; %Pasamos las muestras a unidades de tiempo
distanciaDLL=c*tiempo; % y conseguimos la distancia en metros
plot(distanciaDLL,'b');
title('Distancia entre Emisor y Receptor con DLL')
xlabel('Tiempo [ms]')
ylabel('Distancia [m]')


subplot(4,3,12)
it=1:tSAT;
kk=1;
for ii=0:muestras1msCAlocal:Nchips*tSAT*muestrasChip-
muestras1msCAlocal
vel(kk)=fdop(ii+1)*c/fL1;
kk=kk+1;
end

distanciaDatos=cumsum(vel.*1e-3); %suma acumulativa por posición
plot(distanciaDatos,'b');

```

```

title('Distancia entre Emisor y Receptor con mis datos')
xlabel('Tiempo [ms]')
ylabel('Distancia [m]')

figure (5)
subplot(1,2,1)
ejeP=(2:10:92)
ejeE=ejeP-1;
ejeL=ejeP+1;
plot(ejeP,pp(50:59), 'ro')
hold on
plot(ejeE,ee(50:59), 'go')
hold on
plot(ejeL,ll(50:59), 'bo')
title('Triángulo de correlaciones para 10 iteraciones consecutivas')
legend('Prompt','Early','Late', 'Location', 'best')

subplot(1,2,2)
iteracion=95;
ejePP=6;
ejeEE=1:5;
ejeLL=7:11;
ValoresEarly=[ee5(iteracion) ee4(iteracion) ee3(iteracion)
ee2(iteracion) ee(iteracion)];
ValoresLate=[ll(iteracion) ll2(iteracion) ll3(iteracion)
ll4(iteracion) ll5(iteracion)];
plot(ejePP,sqrt(pp(iteracion)), 'ro')
hold on
plot(ejeEE,sqrt(ValoresEarly), 'go')
hold on
plot(ejeLL,sqrt(ValoresLate), 'bo')
title('Triángulo de correlaciones para una iteración dada')
legend('Prompt','Early','Late', 'Location', 'best')

end

function
[pp,ee,ee2,ee3,ee4,ee5,ll,ll2,ll3,ll4,ll5,disDLL,retardo,disPLL,filPLL
1,filPLL2,pentrada,fasetrack,NCODLL,fildLL1,fildLL2]=bucle_tracking2(t
SAT,Nchips,fase,fi,fdop,c,v,fm,tm,fchip,tchip,tInc,tCA,muestrasInc,mue
strasChip,muestras1msCAlocal,muestrasCAlocal,IncomingtrackPB,IncomingB
B,n,PRN,CAlocal,codephase,freq,CA,a,b,muestraref,fL1,vectormuestraspor
periodoCAacum,vectormuestrasporperiodoCA,CASAT,prompt,early,late,k,k1,
k2,IncomingDem,ret,fasetrack,fasetrack2,xx,early2,early3,early4,early5
,late2,late3,late4,late5,ii,i,CAlocaltrack,long,lli,xxx,NCODLL,acum)

% Esta función realiza el bucle de tracking cuando la frecuencia Doppler
% afecta a toda la señal GPS

% Entradas:
% tSAT = [ms] de la señal GPS
% Nchips = 1023 [chips]
% fase = Fase en [grados] de la señal GPS en BB
% fi = Frecuencia intermedia [Hz]

```

```
% fdop = vector de muestras que simula la frecuencia Doppler
% c = velocidad de la luz en [m/s]
% v = velocidad relativa entre satélite y receptor [m/s]
% fm = Frecuencia de muestreo [Hz]
% tm = Tiempo de muestreo [s]
% fchip = Frecuencia de chip [Hz]
% tchip = Tiempo de chip [s]
% tInc = [s] de la señal GPS
% tCA = [ms] del código C/A
% muestrasInc = #[muestras] de la señal GPS
% muestrasChip = #[muestras] en cada chip
% muestras1msCAlocal = #[muestras] en un periodo de CAlocal
% muestrasCAlocal = #[muestras] de CAlocal
% IncomingtrackPB = Señal GPS entrante al módulo de tracking
% IncomingBB = Señal GPS en BB
% n = Índice de muestras
% PRN = Código asociado al satélite
% CAlocal = Código C/A remuestreado
% codephase = Retraso de código adquirido
% freq = Frecuencia Doppler adquirida [Hz]
% CA = 1023 muestras del código C/A asociado al satélite PRN
% a = Vector de muestras que representa los chips
% b = Parte entera del vector anterior
% muestraref = Muestra de referencia para verificar el tracking
% fL1 = Frecuencia portadora [Hz]
% vectormuestrasporperiodoCAacum = Vector que guarda las muestras por
% periodo del código C/A remuestreado de manera acumulada
% vectormuestrasporperiodoCA = Vector que guarda las muestras por
% periodo del código C/A remuestreado
% CASAT = Código C/A de la señal GPS
% prompt = código prompt
% early = código early
% late = código late
% k = constante de los filtros de lazo
% k1 = constante de los filtros de lazo
% k2 = constante de los filtros de lazo
% IncomingDem = Señal GPS demodulada
% fasetrack = fase de la réplica generada por el PLL
% xx = constante usada en fil1PLL2
% early2 = código early desplazado 1 muestra
% early3 = código early desplazado 2 muestras
% early4 = código early desplazado 3 muestras
% early5 = código early desplazado 4 muestras
% late2 = código late desplazado 1 muestra
% late3 = código late desplazado 2 muestras
% late4 = código late desplazado 3 muestras
% late5 = código late desplazado 4 muestras
% ii = índice de chips
% i = índice de iteraciones
% CAlocaltrack = código C/A usado en el tracking
% long = # muestras por chip
% l11 = índice de muestras para el tracking
% xxx = constante para filDLL2
% NCODLL = Remuestreo en chips del DLL
% acum = constant para filDLL1

%Salidas:
% pp = Envolvente al cuadrado del código prompt
% ee = Envolvente al cuadrado del código early
% ee2 = Envolvente al cuadrado del código early2
% ee3 = Envolvente al cuadrado del código early3
```

```
% ee4 = Envolvente al cuadrado del código early4
% ee5 = Envolvente al cuadrado del código early5
% ll = Envolvente al cuadrado del código late
% ll2 = Envolvente al cuadrado del código late2
% ll3 = Envolvente al cuadrado del código late3
% ll4 = Envolvente al cuadrado del código late4
% ll5 = Envolvente al cuadrado del código late5
% disDLL = Discriminador DLL
% disPLL = Discriminador PLL
% filPLL1 = Filtro PLL de orden 0
% filPLL2 = Filtro PLL de orden 1
% pentrada = Entrada al discriminador PLL
% fasetrack= fase de la réplica generada por el PLL
% NCODLL = Remuestreo en chips del DLL
% filDLL1 = Filtro DLL de orden 0
% filDLL2 = Filtro DLL de orden 1

global prompttrack,
global earlytrack;
global latetrack;
global pp
global ee
global ll
global ee2
global ll2
global ee3
global ll3
global ee4
global ll4
global ee5
global ll5
global disDLL
global disPLL
global retardo
global ret
global filPLL1
global filPLL2
global fasetrack
global pentrada
global filDLL1
global filDLL2
global NCODLL

l=1:long;
if i<2
prompttrack(i,:)=prompt(l);
earlytrack(i,:)=early(l);
latetrack(i,:)=late(l);

else
prompttrack(i,:)=prompttrack(i-1,:);
earlytrack(i,:)=earlytrack(i-1,:);
latetrack(i,:)=latetrack(i-1,:);
end

IncomingDem5456=IncomingDem(ll1);
NCO=exp(-j*2*pi*(freq-fi)*tm*l);

input=IncomingDem5456.*NCO.*exp(-j*fazetrack(i));
```

```

pentrada(i)=sum(input.*prompttrack(i,:))/muestras1msCAlocal;
eentrada(i)=sum(input.*earlytrack(i,:))/muestras1msCAlocal;
lentrada(i)=sum(input.*latetrack(i,:))/muestras1msCAlocal;

pp(i)=abs(pentrada(i)).^2;
ee(i)=abs(eentrada(i)).^2;
ll(i)=abs(lentrada(i)).^2;

disDLL(i)=ee(i)-ll(i);

filDLL1(i)=k*0.1*disDLL(i);

% %Filtro DLL de orden 1
% xxx=xxx+disDLL(i);
% filDLL2(i)=k1*0.01*disDLL(i)+k2*0.01*xxx;

acum=acum+filDLL1(i);
NCODLL(i)=acum;

z1=(0:muestras1msCAlocal-1)*(1/muestrasChip)+NCODLL(i);
x1=floor(z1); %La parte entera

prompttrack(i,:)=CA(mod(x1,Nchips)+1);

e1=circshift(x1,[0 -1]);
t1=circshift(x1,[0 +1]);
earlytrack(i,:)=CA(mod(e1,Nchips)+1); %5456 muestras de {1,-1}
latetrack(i,:)=CA(mod(t1,Nchips)+1); %5456 muestras de {1,-1}

%-----
%Triángulo

e2=circshift(x1,[0 -2]);
t2=circshift(x1,[0 +2]);

early2=CA(mod(e2,Nchips)+1);
late2=CA(mod(t2,Nchips)+1);

e3=circshift(x1,[0 -3]);
t3=circshift(x1,[0 +3]);

early3=CA(mod(e3,Nchips)+1);
late3=CA(mod(t3,Nchips)+1);

e4=circshift(x1,[0 -4]);
t4=circshift(x1,[0 +4]);

early4=CA(mod(e4,Nchips)+1);
late4=CA(mod(t4,Nchips)+1);
e5=circshift(x1,[0 -5]);
t5=circshift(x1,[0 +5]);

early5=CA(mod(e5,Nchips)+1);
late5=CA(mod(t5,Nchips)+1);

eentrada2(i)=sum(input.*early2)/muestras1msCAlocal; %1x1
lentrada2(i)=sum(input.*late2)/muestras1msCAlocal; %1x1

```

```
eentrada3(i)=sum(input.*early3)/muestras1msCAlocal; %1x1
lentrada3(i)=sum(input.*late3)/muestras1msCAlocal; %1x1

eentrada4(i)=sum(input.*early4)/muestras1msCAlocal; %1x1
lentrada4(i)=sum(input.*late4)/muestras1msCAlocal; %1x1

eentrada5(i)=sum(input.*early5)/muestras1msCAlocal; %1x1
lentrada5(i)=sum(input.*late5)/muestras1msCAlocal; %1x1

ee2(i)=abs(eentrada2(i)).^2;
ll2(i)=abs(lentrada2(i)).^2;

ee3(i)=abs(eentrada3(i)).^2;
ll3(i)=abs(lentrada3(i)).^2;

ee4(i)=abs(eentrada4(i)).^2;
ll4(i)=abs(lentrada4(i)).^2;

ee5(i)=abs(eentrada5(i)).^2;
ll5(i)=abs(lentrada5(i)).^2;
%-----  
disPLL(i)=real(pentrada(i))*imag(pentrada(i));
filPLL1(i)=k*disPLL(i);
xx=xx+disPLL(i);
filPLL2(i)=k1*disPLL(i)+k2*xx;
fasetrack(i+1)=fasetrack(i)+filPLL2(i);
```

Bibliografía

- [1] J.Bao, Y. Tsui, *Fundamentals of Global Positioning System Receivers. A Software Approach*. Wiley, 2000.
- [2] K. Borre, D.M. Akos, N. Bertelsen, P. Rinder, S. H. Jensen. *A Software-Defined GPS and Galileo Receiver: A Single-Frequency Approach*. Birkhäuser, 2007.
- [3] F. van Diggelen. *A-GPS: Assisted GPS, GNSS, and SBAS*. Artech House, 2009.
- [4] F. Johansson, R. Mollaei, J. Thor, J. Uusitalo. *GPS Satellite Signal Acquisition and Tracking*. 1998.
- [5] E. D. Kaplan, C.J. Hegarty. *Understanding GPS. Principles and Applications*. Artech House, 2006.
- [6] José A. López-Salcedo, Gonzalo Seco, José López Vicario. *Sistemas de radionavegación*. Universitat Oberta de Catalunya, 2012.
- [7] P. Misra, P. Enge. *Global Positioning System: Signals Measurements, and Performance*. Ganga-Jamuna Press, 2001.
- [8] D. Plausinaitis. *GPS Signal Acquisition. GPS Signals and Receiver Technology MM11*. Danish GPS Center. Aalborg University.
- [9] El-Rabbany, *Introduction to GPS. The Global Positioning System*. Artech House, 2002.
- [10] E. Simona. *Introduction to Global Navigation Satellite Systems*. Tampere University of Technology.
- [11] J. B. Thomas. *Functional Description of Signal Processing in the Rogue GPS Receiver*. JPL Publication 88-15. June 1, 1988. NASA.
- [12] W. Zhuang, J. Tranquilla. *Digital baseband processor for the GPS receiver (parts I and II)*. Aerospace and Electronic Systems, IEEE Transactions.
- [13] W. Zhuang, J. Tranquilla. *Modeling and analysis for the GPS pseudo-range observable*. Aerospace and Electronic Systems, IEEE Transactions.

Resum:

Aquest projecte és un anàlisi bàsic de les etapes d'adquisició i tracking d'un receptor GPS. Per fer-ho, hem simulat el senyal GPS que transmet un satèl·lit i els diferents efectes que experimenta fins arribar al receptor. Un cop és aquí, el senyal GPS s'ha processat per extreure i refinat els seus paràmetres de retard de codi i desviació Doppler, i a partir d'aquest punt estimar la distància que separa el satèl·lit del receptor. Per ampliar l'estudi, hem afegit efectes reals com soroll, multipath, filtrat i desviació Doppler, que ens han permès entendre millor el funcionament del receptor.

Resumen:

Este proyecto es un análisis básico de las etapas de adquisición y tracking de un receptor GPS. Para ello, hemos simulado la señal GPS que transmite un satélite y los diferentes efectos que experimenta hasta que llega al receptor. Una vez aquí, la señal GPS se ha procesado para extraer y refinarse sus parámetros de retraso de código y desviación Doppler, y a partir de ahí estimar la distancia que separa al satélite del receptor. Para ampliar el estudio, hemos añadido efectos reales como ruido, multipath, filtrado y desviación Doppler, que nos han permitido entender mejor el funcionamiento del receptor.

Summary:

This project is a basic analysis of the acquisition and tracking stages of GPS receiver. For this purpose, we simulate the GPS signal transmitted by the satellite and the different effects it suffers until it reaches to the receiver. Once here, the GPS signal is processed to extract and track parameters such as codephase and Doppler frequency and, from thereon, the receiver could estimate its distance from the satellite. In order to further explore the subject, we add real effects such as noise, multipath, filtering and Doppler deviation, which have enabled us to better understand the operation of the GPS receiver.

