

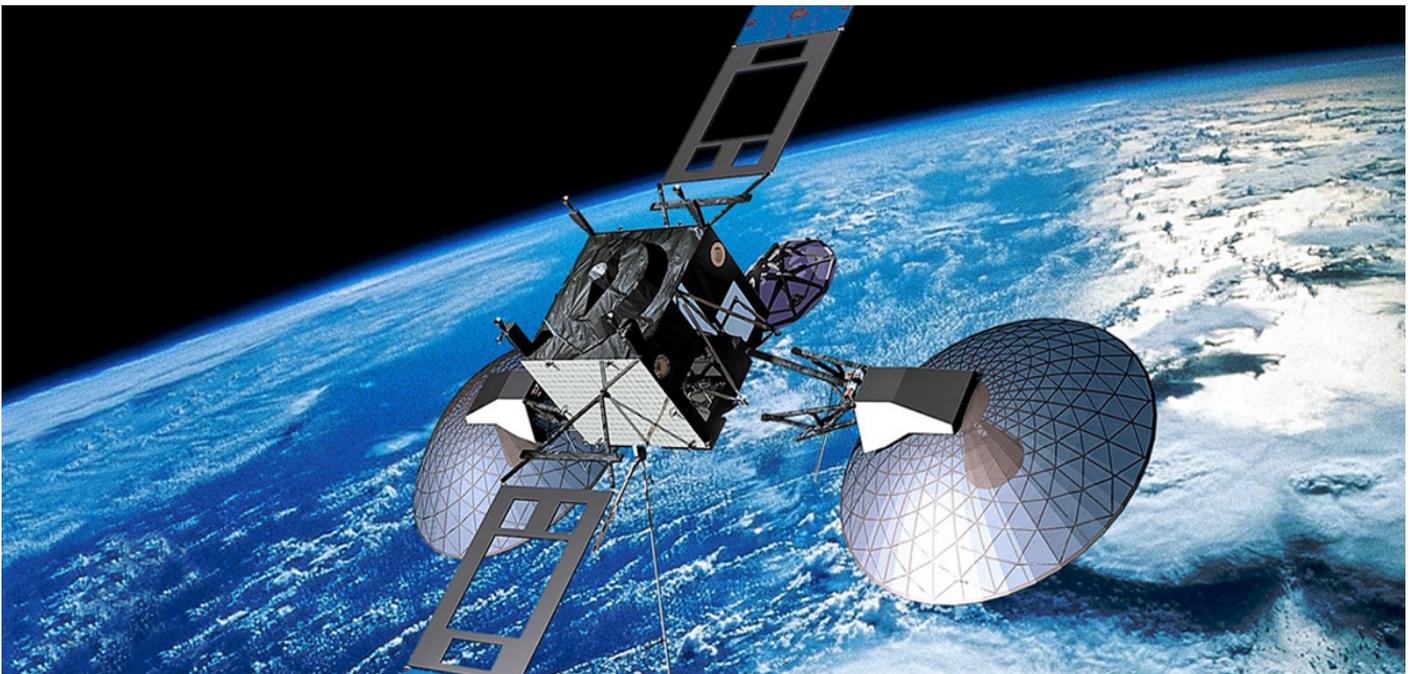


UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Instituto Universitario de Microelectrónica Aplicada

Doctorado en Tecnologías de Telecomunicación

Tesis Doctoral

APORTACIONES A LA GEOLOCALIZACIÓN EFICIENTE



Daniel Alcaraz Real-Arce

Las Palmas de Gran Canaria

Junio de 2017



UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Instituto Universitario de Microelectrónica Aplicada

Instituto: **Instituto Universitario de Microelectrónica Aplicada**

Programa de doctorado: **Doctorado en Tecnologías de Telecomunicación**

Título de la Tesis

“APORTACIONES A LA GEOLOCALIZACIÓN EFICIENTE”

Tesis Doctoral presentada por D. Daniel Alcaraz Real-Arce

Dirigida por el Dr. D. Carlos Javier Sosa González

Codirigida por el Dr. D. Juan Antonio Montiel Nelson

El Director,

El Codirector,

El Doctorando,

Las Palmas de Gran Canaria, a 9 de Junio de 2017



UNIVERSIDAD DE LAS PALMAS
DE GRAN CANARIA

Instituto Universitario de Microelectrónica Aplicada

Doctorado en Tecnologías de Telecomunicación

Tesis Doctoral

APORTACIONES A LA GEOLOCALIZACIÓN EFICIENTE

Daniel Alcaraz Real-Arce

Las Palmas de Gran Canaria

Junio de 2017

Le quiero dedicar esta Tesis Doctoral a las dos personas a las que más les va a alegrar su finalización:

a mis padres.

Gracias mis queridos padres, Ginés y Carmen, por haber estado siempre ahí, apoyándome, dándome ánimos, preguntando tímidamente por la tesis de cuando en cuando para que no la olvidara. Tengo infinitas cosas que agradecerles, y este documento es una más.

Gracias a mis tíos, porque sé que van a ser los siguientes de la lista en alegrarse por la finalización de esta etapa. Aleka, Isabel, Andrés, Paco, muchísimas gracias por todos esos: “¿...y la tesis como va?”. Entre todos consiguieron que mis ánimos y mi ilusión por terminar lo que un día empecé, no flaquearan. Esta tesis es también de ustedes.

Gracias al IUMA por abrirme sus puertas y darme la oportunidad de desarrollar el trabajo que ha dado lugar a esta Tesis. Gracias Antonio. Por tus recordatorios, tus consejos y tus ánimos.

Hay dos personas sin las que definitivamente este documento no sería posible. Muchísimas gracias a mis directores de Tesis: Javier y Juan Antonio. Gracias por confiar en mi. Por acogerme en la División como uno más. Por la ayuda. Por la paciencia. Por el compromiso.

Tengo que agradecer al resto de amig@s, cercanos y lejanos, que han estado ahí todo este tiempo, animando y apoyando hasta el final. Es un lujo contar con amigos de verdad que están ahí ante cualquier circunstancia. Ustedes saben quienes son.

A Judit, por sufrir de una manera tan maravillosa esta última e intensa etapa de Tesis. Por estar siempre a mi lado con una sonrisa, poniendo toda la tranquilidad y el cariño cuando más lo necesitaba.

A la memoria de todas esas personas que me miran desde arriba y que sé que hoy llorarían de felicidad.

Gracias a todos por haber contribuido, de una forma u otra, a la finalización de este documento.

Introducción

En un mundo globalizado, el uso y la evolución de las tecnologías de geolocalización van en aumento. Conocer la posición del usuario de un servicio, de un vehículo, o en definitiva de un objeto en la superficie de la tierra es una de las necesidades más frecuentes tanto de la industria, como de instituciones gubernamentales, y cada día más, una de las aplicaciones más atractivas para los nuevos modelos de negocio.

La tecnología actual permite llevar a cabo la localización de un receptor en la superficie de la tierra gracias al sistema de navegación global por satélite (GNSS). Este sistema se compone de distintas constelaciones de satélites sincronizados entre si que orbitan alrededor de la Tierra generando y transmitiendo diversas señales. Los receptores GNSS utilizan estas señales para determinar su posición a través de un importante número de operaciones que suponen un reto para la autonomía de los dispositivos portátiles.

Las señales que son generadas y transmitidas por los satélites a la Tierra, y recibidas por el receptor. De toda la información emitida, destacan los códigos que permiten identificar de forma unívoca a cada uno de los satélites. Estos son los llamados códigos pseudo aleatorios, o códigos PRN. Estos códigos pertenecen a una familia de secuencias binarias muy particulares, ya que cumplen una serie de propiedades que las hacen de enorme interés para esta aplicación. A lo largo de la presente tesis se analizarán en profundidad estas propiedades, pero a modo de resumen, estas se basan en que al comparar dos de estas secuencias, solamente si son idénticas y están alineadas se obtiene un valor máximo. Para el resto de casos este valor es mínimo.

De esta forma el proceso de detección de los satélites se reduce a la comparación/correlación de la secuencia recibida por el receptor, con todas las secuencias posibles que tiene almacenadas en su memoria. Esto supone un elevado número de operaciones que conlleva una elevada demanda computacional.

El ámbito de estudio de la presente Tesis Doctoral se centra en encontrar soluciones que permitan disponer al receptor de códigos de menor longitud, pero que conserven las propiedades de los códigos originales de manera que sea posible llevar a cabo la detección de satélites con un menor coste computacional, y por tanto con un ahorro de batería (y aumento de autonomía) de los dispositivos portátiles.

Con este objetivo, en el marco del proyecto “Pure GNSS (SolSubC20080 1000282)” subvencionado por la Agencia Canaria de Investigación, Innovación y Sociedad de la Información [1], se inició un estudio previo sobre metodologías capaces de generar nuevos códigos PRN utilizando una aproximación determinista. Inicialmente se tomaron longitudes múltiplo de los códigos de detección, y posteriormente se amplió este concepto a secuencias de longitud no múltiplo. Dentro de esta metodología se exploró para todas las longitudes propuestas la posibilidad de muestrear los bits de forma alternada o contigua, exponiendo los resultados obtenidos.

Con el fin de mejorar estos resultados resultados derivados de utilizar la aproximación determinista [2], se introduce el concepto de algoritmo genético, y se concretiza su aplicación al problema planteado inicialmente. De esta forma se proponen 5 funciones objetivo aplicadas a tres algoritmos de optimización, obteniéndose distintas soluciones que son evaluadas en términos de detección. A partir de aquí se introducen nuevas variables de optimización y se explora la idea de aumentar el número de generaciones del algoritmo, con el fin de mejorar los resultados de detección. Se varían las longitudes de los códigos y se evalúan distintos parámetros a través de los que se obtienen distintos ahorros computacionales en comparación con la metodología tradicional. Lo anterior ha sido publicado en conferencias y revistas especializadas dentro del primer y segundo cuartil [3-5].

A continuación se introduce el concepto de la atenuación, que es un parámetro con el que se enfrentaría un receptor hardware real y que nos ayudará a validar las metodologías propuestas. Además, con el fin de ampliar el caso de estudio, en esta ocasión se aplicará la metodología propuesta al caso de receptores GALILEO. Con el objetivo de mejorar los resultados obtenidos para GPS, se introduce una variación en el método de inicialización de individuos basado en la función mayoría. Esto permite partir de mejores individuos semilla que proporcionarán mejores códigos finales. Los resultados obtenidos para este último caso validan la metodología propuesta estableciendo un compromiso entre ahorro computacional y degradación de señal. Esto resulta útil en casos donde existe una buena visibilidad de satélites e interesa maximizar la autonomía de baterías por parte del receptor, o bien casos en los que es necesario obtener una mejor calidad de señal ya que esta llega degradada. Actualmente se encuentra en fase de revisión una aportación de revista donde se destaca la repetición de códigos en el sistema Galileo (no se ha incluido como anexo porque no está publicada).

Una vez se han obtenido los códigos de detección a través de esta última metodología, estos pueden ser aplicados a la práctica totalidad de los receptores GNSS actuales sin que sea necesaria ninguna modificación de hardware.

El trabajo desarrollado en esta Tesis Doctoral se ha estructurado en seis capítulos. El contenido de los mismos se describe a continuación:

- **Capítulo 1: Estado del Arte**

Este capítulo introduce los conceptos que explican qué es la geolocalización y la forma en que esta se lleva a cabo. Se establece el número de satélites que son necesarios para obtener la posición de un receptor. Además se describe el funcionamiento del sistema GPS presentando los distintos segmentos que lo componen, destacando que es en el segmento de usuario en el que se desarrolla esta Tesis Doctoral.

- **Capítulo 2: Metodología**

En este segundo capítulo se detalla el proceso de detección de satélites que se lleva a cabo en el receptor. Se describe la generación de secuencias de números pseudoaleatorios que identifican unívocamente a cada uno de los satélites, y cuales son las propiedades que los hacen tan interesantes. Se plantea el problema asociado al cálculo necesario para la detección de satélites visibles, y se define el ámbito de investigación de esta tesis.

- **Capítulo 3: Estudio previo con aproximación determinista (GPS)**

Este capítulo se centra en la obtención de una metodología determinista de códigos de identificación de menor longitud que proporcionen un ahorro computacional significativo a la hora de la detección de satélites con respecto a los códigos preestablecidos. A continuación se extiende esta idea para el caso real, en el que se recibe una secuencia continua en la que se repite un determinado código. Por último se presentan y evalúan los resultados obtenidos para distintas longitudes.

- **Capítulo 4: Aproximación basada en Algoritmos Genéticos**

En este capítulo se va un paso más allá en la metodología para generación de códigos de detección. Para ello se recurre a los algoritmos genéticos. En este sentido se llevan a cabo varios experimentos utilizando distintos motores de optimización y funciones objetivo. Se presentan los resultados obtenidos, y se introducen nuevas variables de optimización con el fin de mejorar la calidad de los resultados.

- **Capítulo 5: Metodología utilizando Algoritmos Genéticos aplicados a Galileo**

En este capítulo se introduce la atenuación a la metodología propuesta así como un nuevo método de inicialización de individuos aplicada en este caso a un receptor Galileo. De esta forma se pretende dotar a las metodologías propuestas de generalidad y validarlas de manera que sea posible utilizar estas metodologías en receptores GNSS comerciales sin que sea necesario llevar a cabo ninguna modificación hardware.

- **Capítulo 6: Conclusiones**

Se presentan en este último capítulo las conclusiones extraídas a partir del trabajo desarrollado en esta Tesis Doctoral.

- **Anexo I - Publicaciones**

Se incluyen en este anexo el conjunto de publicaciones elaboradas en el transcurso de esta Tesis Doctoral en el formato original de la publicación.

REFERENCIAS

[1]. "Plataforma basada en Ultra Receptor software para aplicaciones GNSS (SolSubC200801000282)", Agencia Canaria de Investigación, Innovación y Sociedad de la Información del Gobierno de Canarias, Instituto Universitario de Microelectrónica Aplicada, ULPGC, IDECNET S.A., Soluciones Avanzadas Canarias S.L., Desde: 15/10/2009 Hasta: 14/10/2011 N° total de meses: 24.

[2]. Alcaraz, D., Sosa, J., & Montiel-Nelson, J. A. (2012). A novel fast and efficient GPS detection method. DCIS 2012 conference Avignon (France) proceedings, 498-499. ISBN: 978-2-9517461-1-14.

[3]. Sosa, J., Bautista, T., Alcaraz, D., Nelson, J. A. M., & Alonso, S. G. (2013). Generation of New Detection Codes for GPS Satellites Using NSGA-II. In EUROGEN 2013 International Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems. Las Palmas de Gran Canaria (Spain).

[4]. Sosa, J., Alcaraz, D., Bautista, T., Montiel-Nelson, J. A., García-Alonso, S., Monzón-Verona, J. M., et al. (2013). Lite GPS pseudo-code generation using genetic algorithms. *Engineering Computations*, 30(8), 1170–1192.

[5]. Sosa, J., Bautista, T., Alcaraz, D., García-Alonso, S., & Montiel-Nelson, J. A. (2015). Generation of New Detection Codes for GPS Satellites Using NSGA-II. In D. Greiner, B. Galván, J. Périaux, N. Gauger, K. Giannakoglou, & G. Winter (Eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences* (pp. 511–520). *Computational Methods in Applied Sciences*, 36. Springer International Publishing. ISBN: 978-3-319-11541-2.

Índice del documento

Capítulo 1

Estado del Arte

1.1. Sistemas GNSS	1
1.2. Geolocalización	4
1.3. Descripción del sistema GPS	7
1.4. Conclusiones	12

Capítulo 2

Metodología

2.1. Introducción	13
2.2. El proceso de generación de códigos	14
2.3. Conclusiones	22

Capítulo 3

Estudio previo con aproximación determinista (GPS)

3.1. Introducción	23
3.2. Nuevo PRN de 341 bits	24
3.3. Resultados de la aproximación determinista para longitud 341.	26
3.4. Metodología determinista con secuencias de longitud no múltiplo	31
3.5. Conclusiones	39

Capítulo 4

Aproximación basada en Algoritmos Genéticos

4.1. Introducción	42
4.2. Nuevos códigos PRN	47
4.3. Resultados de los métodos presentados	52
4.4. Introducción de nuevas variables de optimización	68
4.5. Experimentos	70
4.6. Conclusiones	71

Capítulo 5

Metodología utilizando Algoritmos Genéticos aplicados a GALILEO

5.1. Galileo	74
5.2. Introducción de la nueva metodología	75
5.3. Experimentos	77
5.4. Conclusiones	85

Capítulo 6

Conclusiones

87

Bibliografía

89

Anexo I- Publicaciones

95

Índice de Figuras

Capítulo I

Estado del Arte

Figura 1.1. Localización de un receptor GNSS con un único satélite a una pseudo distancia 'A'.	4
Figura 1.2. Localización de un receptor GNSS con 2 satélites a pseudo distancias 'A' y 'B'.	5
Figura 1.3. Localización de un receptor GNSS con tres satélites a pseudo distancias A, B y C.	6
Figura 1.4. Utilización de un cuarto satélite como señal de reloj.	6
Figura 1.5. Segmentos del sistema GPS.	8
Figura 1.6. Segmento espacial GPS. Fuente gas.gov.com.	9
Figura 1.7. Estaciones del Segmento de Control de GPS. Fuente gps.gov.com.	10

Capítulo 2

Metodología

Figura 2.1. Bandas de frecuencias GNSS donde se localizan L1 y L2.	15
Figura 2.2. Generación de las señales GPS en los satélites.	16
Figura 2.3. Estructura de la señal L1.	16
Figura 2.4. Efecto de la modulación BPSK en la portadora L1.	17
Figura 2.5. Generador de códigos C/A.	18
Figura 2.6. Auto-correlación (izq.) y cros-correlación (dcha.) del PRN I.	19

Capítulo 3

Estudio previo con aproximación determinista (GPS)

Figura 3.1: Correlaciones entre el nPRN y los tramos "A" de los 32 satélites.	26
Figura 3.2: Correlaciones entre el nPRN y los tramos "B" de los 32 satélites.	27
Figura 3.3: Correlaciones entre el nPRN y los tramos "C" de los 32 satélites.	28
Figura 3.4. Correlaciones entre el nPRN y los tramos "A" de los 32 satélites.	29
Figura 3.5. Correlaciones entre el nPRN y los tramos "B" de los 32 satélites.	30
Figura 3.6. Correlaciones entre el nPRN y los tramos "C" de los 32 satélites.	30
Figura 3.7. Relación señal a ruido para el nPRN I de 512 bits muestreados a $\frac{1}{2}$ f.	33
Figura 3.8. Relación señal a ruido para el nPRN I de 512 bits muestreado a f.	34
Figura 3.9. Relación señal a ruido para el nPRN I de 256 bits muestreados a $\frac{1}{4}$ f.	35
Figura 3.10. Relación señal a ruido para el nPRN I de 256 bits muestreados a f.	36

Capítulo 4

Aproximación basada en Algoritmos Genéticos

Figura 4.1. Ejemplo de una función coste.	44
Figura 4.2. Esquema seguido por los algoritmos genéticos.	45

Figura 4.3. Espacio de búsqueda completo (cubo). nPRN óptimo para la solución 2a utilizando NSGA-II y el Omni-Optimizador y la solución I de GENEsYs.	62
Figura 4.4. Representación de la Frontera de Pareto para la búsqueda del nuevo código PRN para el satélite I utilizando NSGA-II y la función multiobjetivo I.	65
Figura 4.5. Correlación entre los datos que recibe el GPS y el nPRN propuesto para el satélite I.	67

Capítulo 5

Metodología utilizando Algoritmos Genéticos aplicados a GALILEO

Figura 5.1. Metodología propuesta para la obtención de los nuevos códigos Galileo E5a/b.	76
Figura 5.2. Ejemplo de la técnica de inicialización propuesta.	80
Figura 5.3. Atenuación de la señal que se encuentra al ejecutar el proceso de correlación una sola vez en términos de longitud del código.	84
Figura 5.4. Ganancia de señal vs longitud de código para el mismo esfuerzo computacional que los códigos PRN estándar.	85

Índice de Tablas

Capítulo 2

Metodología

Tabla 2.1. Asignación de fase en los códigos C/A.	20
---	----

Capítulo 3

Estudio previo con aproximación determinista (GPS)

Tabla 3.1. Tabla de verdad para el caso de 341 bits alternos.	25
Tabla 3.2. Valores de correlaciones con nPRN de 341 bits muestreados a 1/3 f.	28
Tabla 3.3. Valores de correlaciones con nPRN de 341 bits muestreados a f.	31
Tabla 3.4. Valores de detección para un nPRN de 512 bits.	34
Tabla 3.5. Valores de detección para un nPRN de 256 bits.	37
Tabla 3.6. Valores de detección para un nPRN de 128 bits	39

Capítulo 4

Aproximación basada en Algoritmos Genéticos

Tabla 4.1. Parámetros de configuración del algoritmo genético y detalles de ejecución.	54
Tabla 4.2. Picos de detección mínimos (valor mínimo de DPA, DPB y DPC) y ruido máximo (N) para el nPRN óptimo cuando las secuencias recibidas se encuentran alineadas utilizando GENEsYs y una función objetivo simple.	55
Tabla 4.3. Picos de detección (DP) mínimos y ruido máximo (N) para el nPRN cuando las señales de entrada están alineadas utilizando NSGA-II para todas las soluciones de funciones objetivas.	57
Tabla 4.4. Picos de detección mínimos (DP) y ruido máximo (N) utilizando Omni-Optimización.	58
Tabla 4.5. Comparativa de los picos de detección (DP) y ruido máximo (N) obtenidos utilizando NSGA-II, Omni-Optimizador y GENEsYs.	60
Tabla 4.6. Selección del mejor nPRN a partir de los tres motores de optimización y soluciones.	64
Tabla 4.7. Parámetros para el algoritmo NSGA-II utilizado.	70
Tabla 4.8. Experimentos con NSGA-II para el Satélite 1 y $\xi < 400$.	72

Capítulo 5

Metodología utilizando Algoritmos Genéticos aplicados a GALILEO

Tabla 5.1. Parámetros para el algoritmo NSGA-II.	78
Tabla 5.2. Experimentos de inicialización de individuos para Galileo.	81
Tabla 5.3. Experimentos de inicialización de individuos para satélites E5a Galileo.	83

Capítulo I

Estado del Arte

I.1. Sistemas GNSS	1
I.2. Geolocalización	4
I.3. Descripción del sistema GPS	7
I.4. Conclusiones	12

Acrónimos

DoD	Department of Defense
ESA	European Space Agency
FPGA	Field Programmable Gate Array
GA	Genetic Algorithm
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
ITU	International Telecommunications Union
LORAN	LOng-range Aid to Navigation
NAVSAT	Navy Navigation Satellite System
NAVSTAR	Navigation Satellite Timing And Ranning

I.1. Sistemas GNSS

Conocer la posición en la que se localiza un objeto determinado o una persona en la superficie de la tierra ha sido uno de los avances más importantes que se han producido en las últimas décadas [1]. De manera cada vez más frecuente, los gobiernos y las empresas privadas están utilizando

estos sistemas basados en navegación vía satélite para fines tanto comerciales como militares [2]. En un mundo globalizado, todo apunta a que el uso y la evolución de las tecnologías de geolocalización irá en aumento [3].

El sistema navegación global por satélite (GNSS) [4] a nivel mundial está basado en constelaciones de satélites sincronizados entre sí que siguen órbitas polares alrededor de la tierra transmitiendo diversas señales [5, 6]. Cada una de estas señales es generada a bordo de cada satélite, e incluyen una secuencia numérica unívoca de identificación de dicho satélite. Los receptores GNSS utilizan estas señales y las secuencias de identificación para determinar su posición. Actualmente este procedimiento es el empleado por los receptores GNSS que todos utilizamos diariamente en productos comerciales como GPS Garmin, GPS Tomtom o cualquier teléfono móvil con geolocalización, entre otros.

Los sistemas de navegación por satélite se vienen desarrollando en todo el mundo desde la década de los sesenta [7]. Desde la perspectiva puramente militar, conocer la posición de las tropas ha sido un punto clave en las guerras modernas. A raíz de la invención de la radio, se potenció la automatización de la determinación de la posición. En ese sentido el Departamento de Defensa (DoD) de los Estados Unidos de América desarrolló distintos sistemas que permitían obtener la posición de sus tropas mediante la triangulación de diversas señales generadas por estaciones terrestres, como pudo ser el sistema LOnge-range Aid to Navigation (LORAN) [8]. Sin embargo, las estaciones fijas en tierra tienen asociada una debilidad estratégica inherente a su condición fija, así como un inconveniente asociado a la disponibilidad geográfica de aliados donde situar dichas estaciones radio. Estos factores provocaron la necesidad por parte del Departamento de Defensa Americano de disponer de un sistema de geolocalización global que permitiera desplegar la cobertura de señal en aquellos puntos donde fuera necesario, reduciendo al máximo el peligro de ataque a las instalaciones de soporte a la geolocalización [9]. Un primer prototipo de este sistema fue el conocido como TRANSIT o NAVSAT [10]. Este fue el primer sistema de satélites operacional, y supuso un importante avance a la hora de diseñar el primer sistema GNSS denominado Sistema de Posicionamiento Global (GPS). GPS fue desarrollado durante la

década de los setenta como herramienta puramente militar, pero debido a lo exitoso de este avance, el uso del sistema se tornó poco a poco indispensable para un conjunto amplio de aplicaciones civiles.

Desde sus comienzos, el GPS ha constituido un área de investigación muy activa para los científicos [11, 12], que ha sido reforzado por el hecho de que millones de usuarios lo han empleado a lo largo de estos 40 años. Este hecho, junto a la rivalidad existente entre naciones y la gestión única que se realiza sobre el sistema completo por parte del Departamento de Defensa Americano, ha inducido a varias potencias mundiales a desarrollar su propio sistema de navegación global. Así, por ejemplo Rusia ha desarrollado un sistema de posicionamiento global llamado GLONASS [13], que fue iniciado en 1976. La baja cobertura ofertada inicialmente se vio posteriormente ampliada a una cobertura global en 1991, encontrándose desde 2005 en una fase de modernización. No obstante, aunque se barajaba una fecha de finalización para uso civil a finales de 2015, su conclusión es incierta debido distintos problemas técnicos así como a los problemas económicos por los que pasa la economía mundial y la rusa en particular.

A finales de los noventa, la Unión Europea propuso su propio sistema de navegación llamado GALILEO [9]. El retraso sufrido por la implantación de GALILEO llevó a la Agencia Espacial Europea (ESA) a lanzar en Diciembre de 2005 un primer satélite experimental con el único objeto de ocupar las frecuencias asignadas para no perder la concesión de la Unión Internacional de Telecomunicaciones (ITU). En 2008 fue lanzado un segundo satélite experimental con el fin de recabar datos científicos [13]. A estos dos satélites de pruebas en fase experimental se les han unido en los últimos años 10 equipos más, de un total de 24 que estarán operativos en 2020. Los planes de implantación de GALILEO sufren retrasos importantes pero se encuentra operativo proporcionando servicios básicos, aunque no los previstos, desde el último trimestre de 2016.

A finales de la primera década del siglo XXI, el gobierno chino se interesa por los sistemas GNSS proponiendo el sistema Beidou, también llamado COMPASS [14]. A pesar de que la opacidad del gobierno chino no permite disponer en la actualidad de las especificaciones oficiales de dicho sistema de

navegación, ni de la marcha del proyecto, se cree que el sistema constará de 35 satélites y empezará a ofrecer servicios de manera global en 2020 [15].

1.2. Geolocalización

El término geolocalización, se ha utilizado para denominar a la nueva revolución digital caracterizada por el desarrollo del acceso universal a coordenadas y localización geográfica en cualquier lugar, en cualquier momento y bajo coste [16]. La geolocalización permite a una persona, vehículo, equipamiento u objeto, obtener su posición (producir información geográfica y coordenadas, dicho en otras palabras, localizar), particularmente durante un proceso de movilidad, viaje o desplazamiento.

El principio en el que se basa la geolocalización consiste en realizar mediciones del tiempo de propagación de las señales entre un transmisor y un receptor para obtener la distancia que los separa. Una vez obtenido ese tiempo, para conocer la distancia, se ha de multiplicar por la velocidad de la luz. Sin embargo, puesto que la velocidad de la luz es constante, es usual que para ahorrar esfuerzo computacional de los receptores, se obvie el paso del tiempo a la distancia (multiplicación por la velocidad de la luz) y se trabaje directamente con los tiempos. En este contexto, a la medida de tiempo se la conoce como pseudo distancia.

Cuando el receptor obtiene la señal de un único satélite, calcula la pseudo distancia a la que se encuentra. El receptor se localiza en algún punto de la superficie de una esfera cuyo radio es 'A', tal como ilustra la figura 1.1.

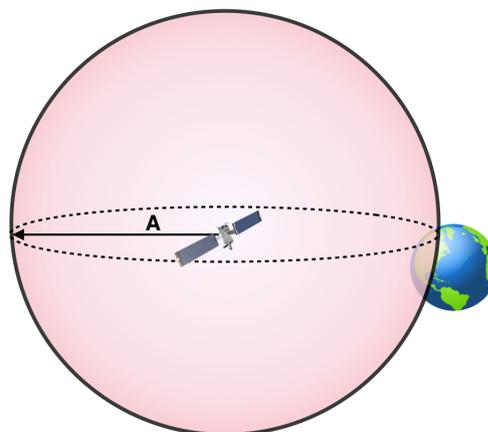


Figura 1.1. Localización de un receptor GNSS con un único satélite a una pseudo distancia 'A'.

Cuando el receptor observa un segundo satélite, se calcula la pseudo distancia a la que el receptor se encuentra de este nuevo satélite. Si identificamos 'B' como dicha pseudo distancia, al igual que en el caso anterior, el receptor se ha de encontrar en algún punto de la superficie de la esfera de radio 'B'. Puesto que el receptor ha de encontrarse en la superficie de ambas esferas, los únicos puntos de dichas esferas que cumplen la restricciones descritas son aquellos que se encuentran en su intersección, tal como se muestra en la franja verde de la figura 1.2.

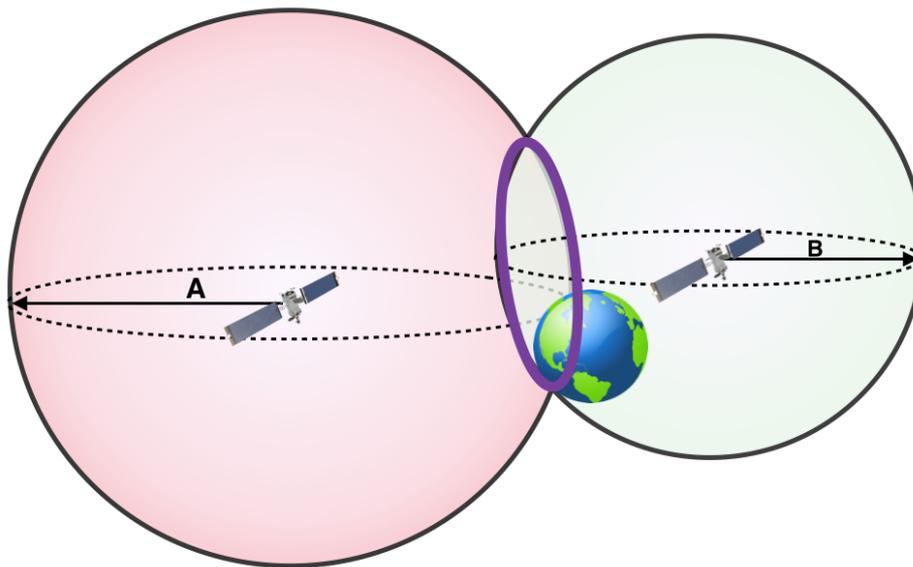


Figura 1.2. Localización de un receptor GNSS con 2 satélites a pseudo distancias 'A' y 'B'

Siguiendo el mismo procedimiento, si el receptor obtiene la señal de un tercer satélite, como se muestra en la figura 1.3, existirán tres superficies esféricas que se interseccionan únicamente en dos puntos. Uno de estos puntos es descartado como posición puesto que estará fuera de la Tierra (P_E en la figura 1.3), por lo que se obtiene un solo punto en el que se localizará al receptor (P_T en la figura 1.3).

Mediante este método es posible determinar la posición del receptor de forma unívoca. Sin embargo, mientras que los relojes que incorporan los satélites son extremadamente precisos, esta característica en el reloj del receptor está muy limitada debido a que se prima el criterio de bajo coste en su implementación. Por este motivo se requiere la señal de un cuarto satélite que será utilizada como referencia de reloj tal como indica la figura 1.4.

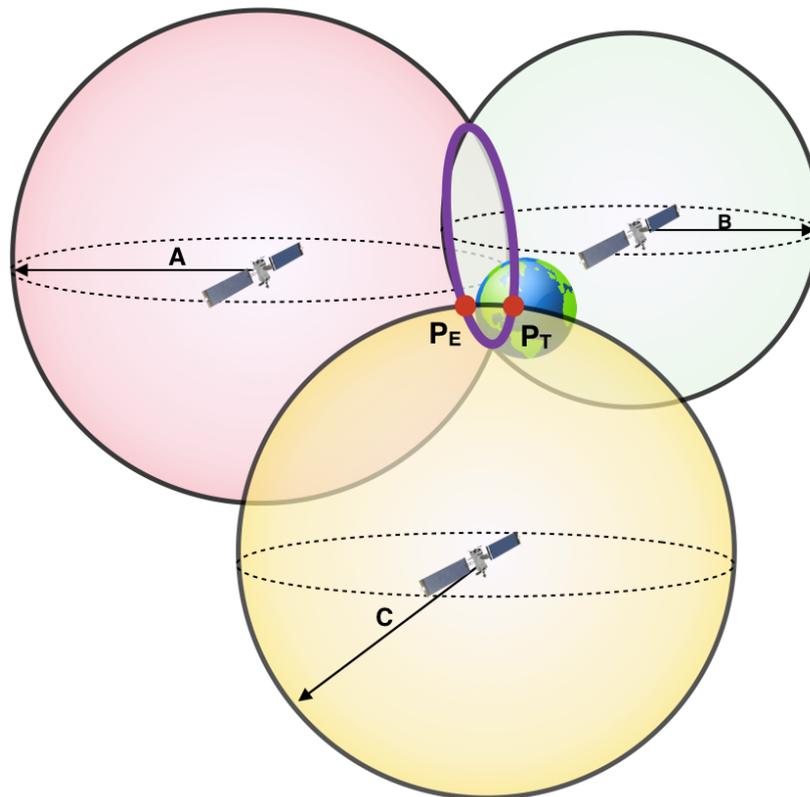


Figura 1.3. Localización de un receptor GNSS con tres satélites a pseudo distancias A, B y C.

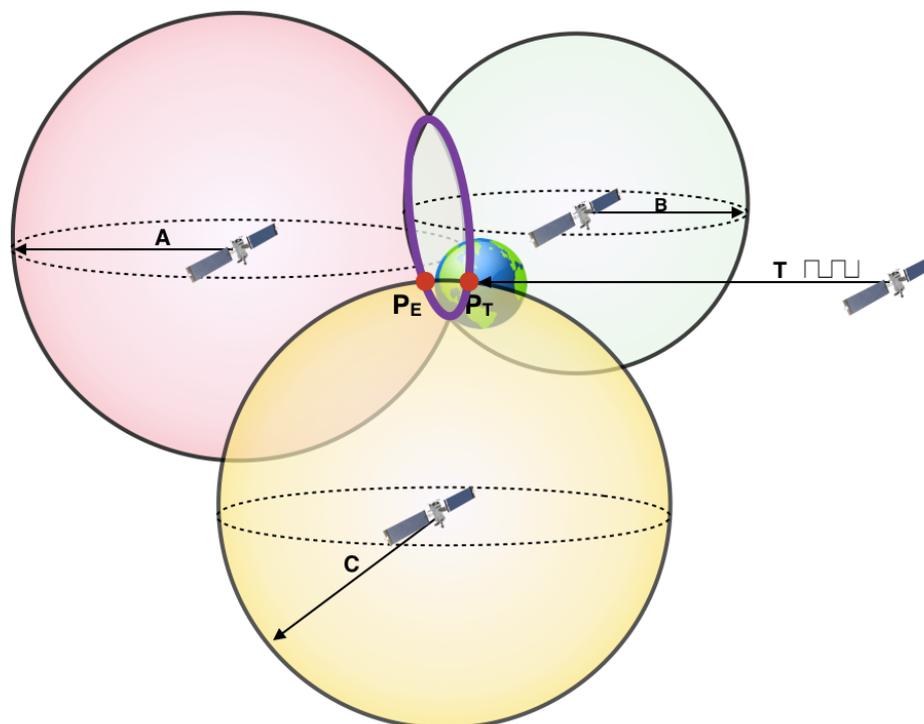


Figura 1.4. Utilización de un cuarto satélite como señal de reloj.

En este escenario, a modo de resumen y en base a lo descrito en los párrafos anteriores, para determinar la posición un receptor GNSS es necesaria la recepción de señal de cuatro satélites. Uno de ellos será tomado como referencia de reloj, y los otros tres se emplearán para medir pseudo distancias y obtener la posición del receptor. Cabe destacar que, la recepción de más de cuatro satélites permite mejorar la precisión de la localización. Pero no hay que olvidar que la inclusión de más pseudo distancias (más superficies esféricas) en el cálculo requiere un mayor esfuerzo computacional.

1.3. Descripción del sistema GPS

En 1973 el Departamento de Defensa de los Estados Unidos comenzó el desarrollo del sistema de posicionamiento global (GPS) [17]. En 1995 hicieron públicos los datos de la frecuencia L1 y declararon la operatividad total del sistema. El propósito principal de las aplicaciones de GPS, es determinar la localización con una precisión pre-establecida. Inicialmente, el objetivo principal de la especificación y desarrollo de este sistema global de navegación por satélite (GNSS) fue formar parte de un equipamiento militar para asistir a la localización de tropas. Actualmente, su uso en aplicaciones civiles se ha visto enormemente extendida [2], y el número de aplicaciones civiles que requieren soporte GPS crece cada día.

La tecnología desplegada con el sistema GPS tiene tal grado de madurez que se ha hecho impensable su ausencia en ciertos ámbitos científicos como pueden ser la geodesia o la cartografía [1], donde es imprescindible realizar medidas muy precisas. Por otra parte, cada día surgen nuevas aplicaciones para las que el sistema originalmente no fue ideado. Ejemplo de ello son las que estudian las condiciones atmosféricas con el objeto de obtener medidas medioambientales indirectas, con aplicaciones específicas [18] como la identificación de polvo en suspensión en Asia con fines meteorológicos [19], evaluar la distorsión que existe en una imagen debido al nivel de humedad existente en la atmósfera [20] en el campo de la astronomía, o simplemente emplear las señales GPS como una fuente precisa de reloj [21] entre otras.

El sistema GPS está constituido por tres segmentos, tal y como muestra la figura 1.5. Por una parte se encuentra el segmento espacial, constituido por

los satélites que orbitan alrededor de la tierra. Por otra, el segmento de control constituido por las estaciones terrestres, que se encargan de monitorizar y controlar la constelación de satélites desde tierra. Por último, el segmento usuario constituido por los receptores de GPS que detectan y reciben la señal de los satélites para determinar la posición [1].

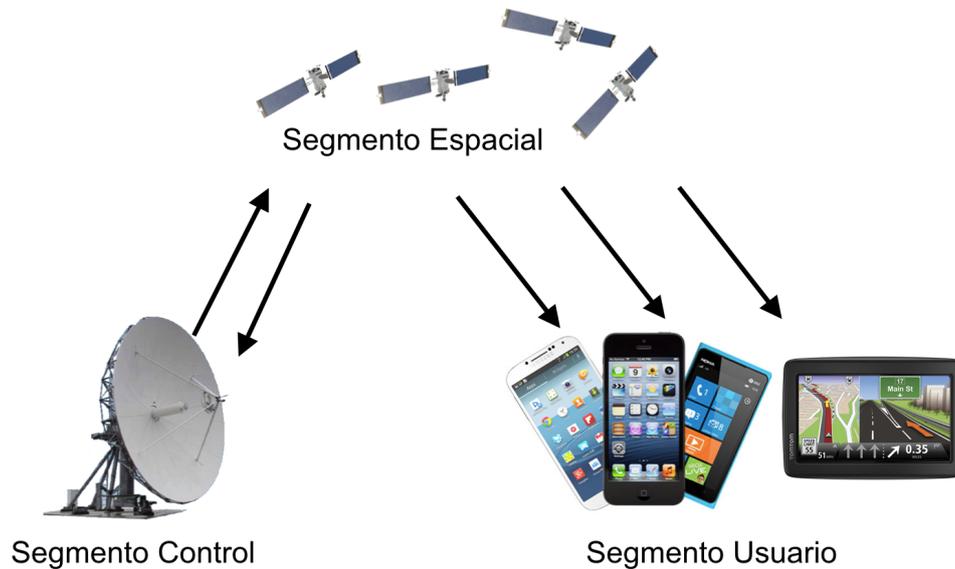


Figura 1.5. Segmentos del sistema GPS.

El segmento espacial consta de un total de 24 satélites activos (ver figura 1.6) de los 31 que hay en órbita. Estos satélites se encuentran situados en 6 planos orbitales igualmente distribuidos a 20.200 km de altitud con una inclinación de 55° grados con respecto al plano ecuatorial. Los satélites están sincronizados a nivel de reloj con sus vecinos en órbita, de forma que la constelación forma una red global sincronizada. Entre los satélites existe comunicación a través del envío de comandos de red. Esta red está diseñada de forma que no requiera ninguna orden para mantener la sincronización y red orbital durante al menos 180 días. Cada satélite orbita la tierra dos veces cada 24 horas.

Cada uno de los 6 planos orbitales estaba inicialmente distribuido en cuatro franjas ocupadas por satélites. Esta ordenación en 24 franjas fue ampliada en 2011 de forma que seis satélites fueron reposicionados y tres añadidos, expandiéndose 3 de las 24 franjas. La configuración inicial aseguraba que los usuarios son capaces de visualizar al menos cuatro satélites

desde casi cada punto en el planeta. Con la nueva configuración de 27 franjas, se ha aumentado la cobertura así como la precisión.

A fecha de Marzo de 2017 existen 31 satélites operacionales en la constelación GPS. De estos, 27 conforman el núcleo básico de la constelación, y el resto pueden ser utilizados para aumentar el rendimiento, pero no se utilizan en el funcionamiento normal de GPS. Además existen varios satélites orbitando que ya no se encuentran en funcionamiento, que se mantienen en órbita por si en alguna ocasión surge la necesidad de reactivarlos.

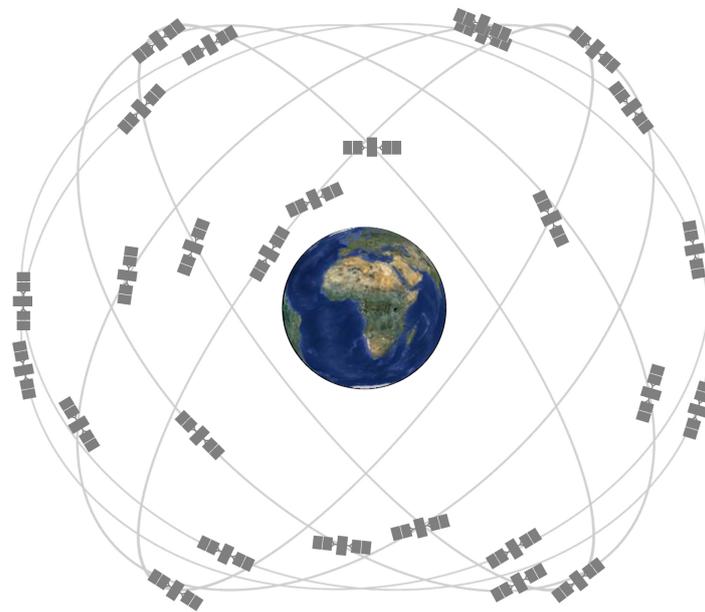


Figura 1.6. Segmento espacial GPS. Fuente gas.gov.com.

El segmento de control consiste en una red de estaciones terrestres que monitorizan la constelación de satélites. Estas estaciones monitorizan las transmisiones y llevan a cabo análisis. Como se muestra en la figura 1.7, a fecha de abril de 2016, el segmento de control operacional incluye una estación de control maestra, una estación de control alternativa, 11 antenas de control y 15 localizaciones de monitorización. Por medidas de seguridad, solamente la estación terrena de Schriever AFB en Colorado es capaz de enviar órdenes a la red de satélites. Por motivos de redundancia, existe la posibilidad de que entre en funcionamiento la estación Vandenberg AFB de California en modo control y no monitorización.

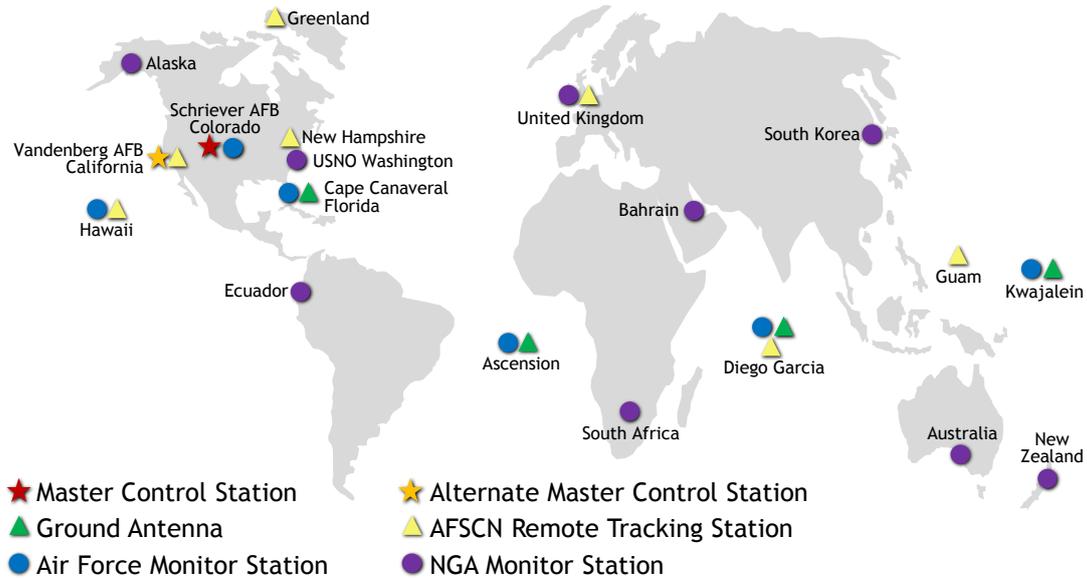


Figura 1.7. Estaciones del Segmento de Control de GPS. Fuente gps.gov.com.

La estación principal de control se encuentra en Colorado y se encarga de generar y transmitir a los satélites los mensajes de navegación asegurándose de la precisión y correcto funcionamiento de la constelación de satélites. Recibe información de navegación de las estaciones de monitorización y la utiliza para calcular las localizaciones precisas de los satélites GPS en el espacio, y transmitir los datos a los satélites.

Las estaciones de monitorización realizan el seguimiento de la constelación de satélites GPS según pasan sobre ellas. Estas estaciones recaban datos atmosféricos, medidas de rango y portadora, así como señales de navegación. Existen en total 15 estaciones de monitorización localizadas alrededor del mundo.

Las antenas terrestres se encargan de comunicar con los satélites GPS. Se utilizan para recibir datos de navegación y recabar datos de telemetría. Existen cuatro antenas terrestres GPS localizadas en las estaciones de monitorización. Además el segmento de control se conecta con siete estaciones de seguimiento remoto alrededor del mundo, aumentando la visibilidad, flexibilidad, y robustez para telemetría y seguimiento.

El segmento de usuario se basa en los receptores de GPS. Desde los que se encuentran integrados en los modernos teléfonos inteligentes, hasta los que se utilizan en los vehículos [22]. Los receptores de GPS que se utilizan ya de

manera habitual y cotidiana deben cumplir varios requisitos. Uno de los más importantes está ligado a su naturaleza, ya que han de ser portátiles. Por este motivo es importante que minimicen su consumo de energía con el fin de aumentar su autonomía. Es en este segmento, el del usuario, en el que se desarrolla la presente Tesis Doctoral.

Como se ha explicado, un receptor GPS establece su posición utilizando un algoritmo de tres pasos. Primero, el receptor identifica y sigue a los satélites visibles [4]. A continuación, el receptor mide los retardos de las señales de los satélites. Por último, el receptor calcula su posición cuando existen al menos cuatro satélites identificados. Los tres pasos descritos se ejecutan siguiendo un proceso iterativo que es repetido de forma perpetua [23]. Tras cada repetición, se obtendrá una localización. Si bien los retardos entre las señales de los satélites GPS se obtienen mediante simples restas, descubrir y seguir los satélites visibles son tareas que requieren un enorme número de operaciones [24].

Con respecto al propósito inicial del desarrollo de GPS, que era obtener información muy precisa sobre la posición del usuario de la aplicación, se ha producido un enorme progreso derivado del gran avance que se ha experimentado en lo referente a la comprensión del comportamiento de las señales del sistema [25]. Esto ha sido en su mayor parte gracias a un modelado preciso del comportamiento multicamino/multitrayectoria de estas señales al verse afectados por colisiones, es decir las reflexiones de la señal a lo largo de su trayectoria [26]. Este conocimiento ha sido incorporado a las aplicaciones GPS actuales, con el fin de lograr una mejor precisión [27]. Las novedosas aplicaciones que han aparecido, han experimentado una gran demanda, sobre todo debido al incremento en el uso de las últimas generaciones de teléfonos móviles, con el aumento de sus capacidades de procesamiento [28]. En estos dispositivos, el GPS es una aplicación adicional que debe operar de manera independiente, pero a la vez debe competir en cuanto a consumo de energía con el resto de aplicaciones [29]. Puesto que la mayoría de los dispositivos móviles poseen una fuente finita de energía, hoy en día es muy importante que el esfuerzo computacional demandado por las aplicaciones GPS mantenga su consumo al mínimo.

El trabajo presentado en esta Tesis Doctoral aporta diversas técnicas y metodologías que, aplicadas a los sistemas actuales, permiten reducir la energía consumida por las aplicaciones de GPS.

I.4. Conclusiones

A través de este capítulo se han introducido los conceptos que explican qué es la geolocalización y la forma en que esta se lleva a cabo. Se ha concluido que son necesarios 3 satélites para triangular la posición de un receptor GPS, más un cuarto que actuará como señal de reloj. Además se han introducido el funcionamiento del sistema GPS presentado los distintos segmentos que lo componen destacando que es en el segmento de usuario en el que se desarrolla la presente Tesis.

Capítulo 2

Metodología

2.1. Introducción	13
2.2. El proceso de generación de códigos	14
2.3. Conclusiones	22

Acrónimos

AN	Auto-correlation Noise
BPSK	Binary Phase Shift Keying
C/A	Coarse Acquisition Code
CN	Cross-correlation Noise
CPU	Central Processing Unit
DP	Detection Peak
FFT	Fast Fourier Transform
GPS	Global Positioning System
IFFT	Inverse Fast Fourier Transform
P(Y)	Encryption Precision Code
PRN	Pseudo Random Number
UHF	Ultra High Frequency

2.1. Introducción

Como se ha descrito en el capítulo anterior, para poder averiguar su localización, el receptor GPS necesita recibir señales de al menos 4 satélites. Estas señales son generadas en cada uno de los satélites y transmitidas a

tierra continuamente. El proceso que se lleva a cabo en el receptor para averiguar de qué satélites se trata requiere un importante número de cálculos. En este segundo capítulo se recogen los aspectos más relevantes del mismo.

2.2. El proceso de generación de códigos

Las señales GPS generadas por los satélites son transmitidas a la Tierra en dos radiofrecuencias en UHF que cubre la banda de frecuencias entre los 500 MHz y los 3 GHz. Estas frecuencias, que se representan en la figura 2.1, se denominan L1 y L2 y derivan de una frecuencia común, $f_0 = 10,23$ MHz:

$$f_{L1} = 154f_0 = 1575,42 \text{ MHz,}$$

$$f_{L2} = 120f_0 = 1227,60 \text{ MHz}$$

La portadora de ambas señales transporta información de navegación relativa a las órbitas de los satélites, secuencias de propagación y factores de corrección de errores entre otros. La información transportada en la portadora se codifica mediante una o incluso dos secuencias binarias únicas. La primera se denomina código de adquisición ordinario (C/A, Coarse Acquisition Code), y la segunda se conoce como código de cifrado de precisión (P(Y), Encryption Precision Code). El código C/A es una secuencia de 1.023 bits repetido cada milisegundo obteniendo una tasa de bit de 1.023 MHz. El código P es un código más largo (unos $2,35 \cdot 10^4$ bits) con una tasa de bit de 10,23 MHz, que es repetido cada semana empezando al principio de la semana GPS emplazado en la media noche de sábado al domingo. Además el código C/A sólo está presente en la portadora L1, mientras que el código P(Y) se encuentra en L1 y L2.

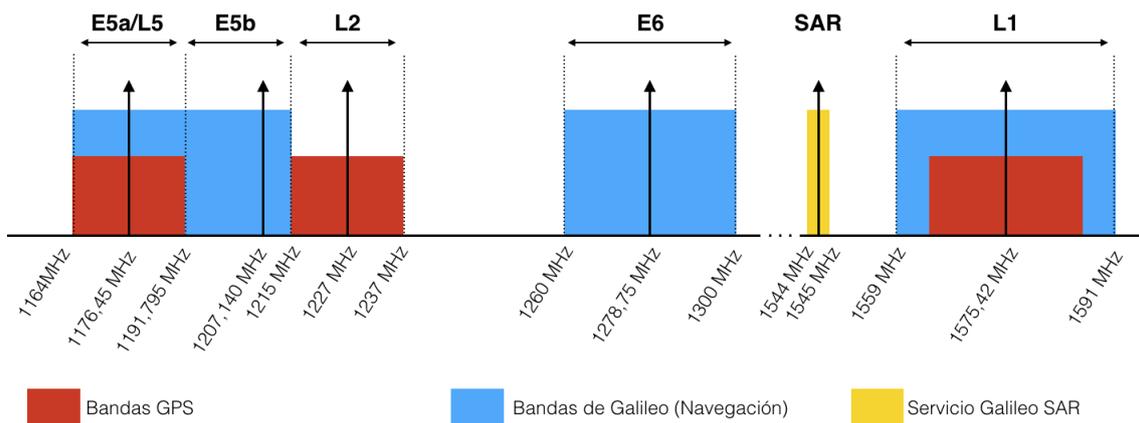


Figura 2.1. Bandas de frecuencias GNSS donde se localizan L1 y L2.

El proceso de generación de las señales en los satélites se representa en la figura 2.2. A la izquierda del todo, la señal de reloj principal de 10,23 MHz, es suministrada a los bloques que obtendrán las portadoras L1 y L2 multiplicando por 154 y 120 respectivamente. Debajo de estos, se encuentra un limitador destinado a estabilizar la señal de reloj antes de ser suministrada a los generadores de los códigos P(Y) y C/A. Abajo del todo se representa el generador de datos que proporciona los datos de navegación. Los generadores de códigos están sincronizados con los generadores de datos a través de la señal X_1 suministrada por el generador de código P(Y). Después de la generación de códigos, estos son combinados con los datos de navegación mediante sumadores en módulo 2. A continuación los datos combinados se transfieren a los dos moduladores para la frecuencia L1. Aquí las señales se modulan en la portadora mediante modulación BPSK introduciendo un desfase de 90° entre ambos códigos. Después de que la parte correspondiente a P(Y) se atenúe 3dB, las dos señales L1 se suman para formar una sola señal L1.

En la figura 2.3 se muestran las tres señales que van a modular la portadora L1. El código C/A se repite cada ms, mientras que un bit de navegación dura 20 ms. Por lo que por cada bit de navegación, la señal contiene 20 códigos C/A completos.

En la figura 2.4 se pueden ver el código C/A (señal identificada como C), los datos de navegación (señal identificada como D), la señal sumada en modulo 2 ($C \text{ xor } D$), y la portadora. La señal final está modulada en BPSK con un desfase de 180° cada vez que se produce un cambio de bit.

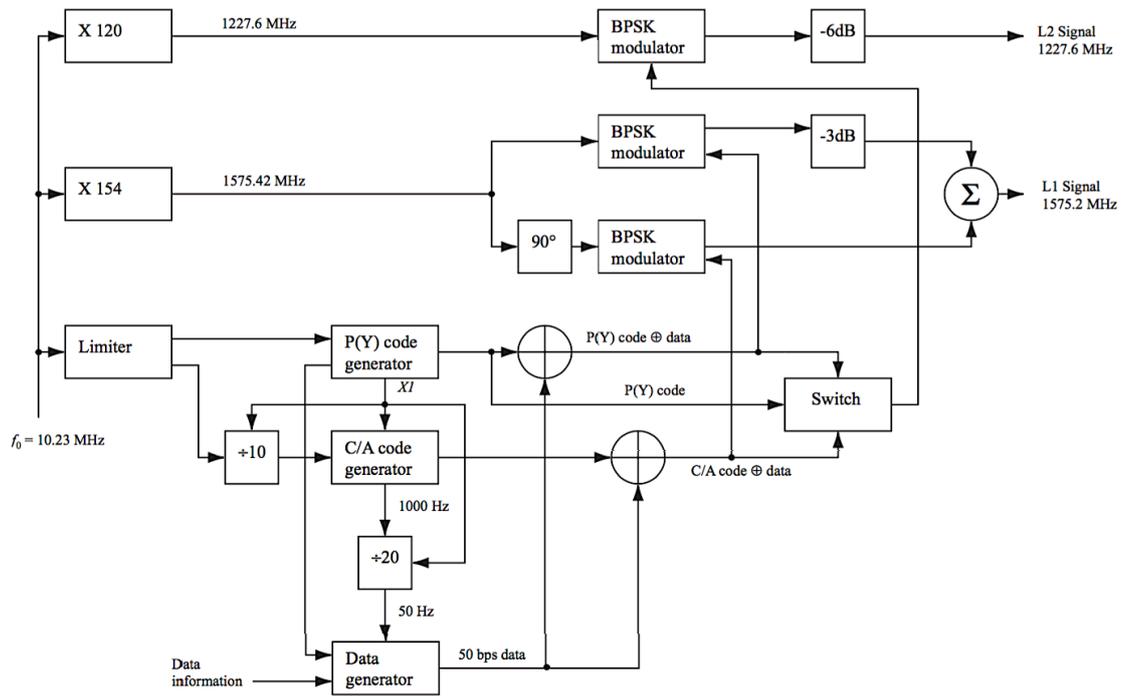


Figura 2.2. Generación de las señales GPS en los satélites.

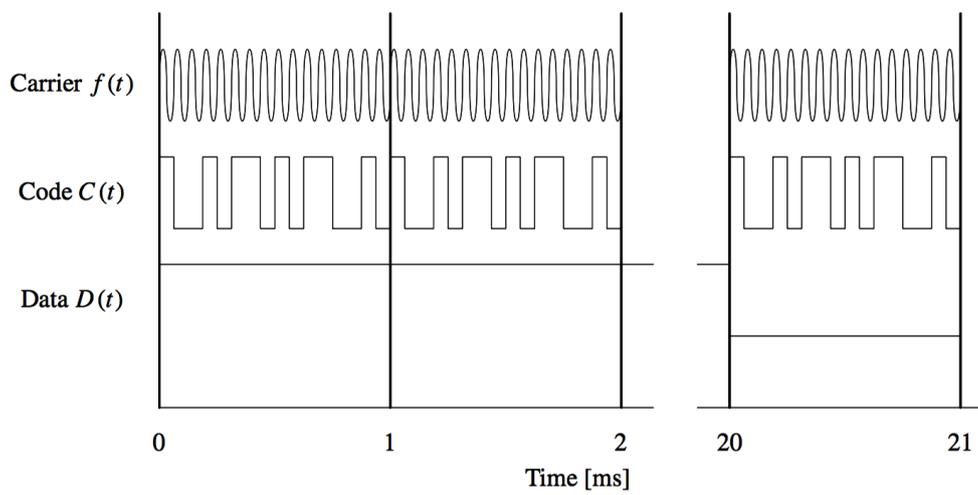


Figura 2.3. Estructura de la señal L1.

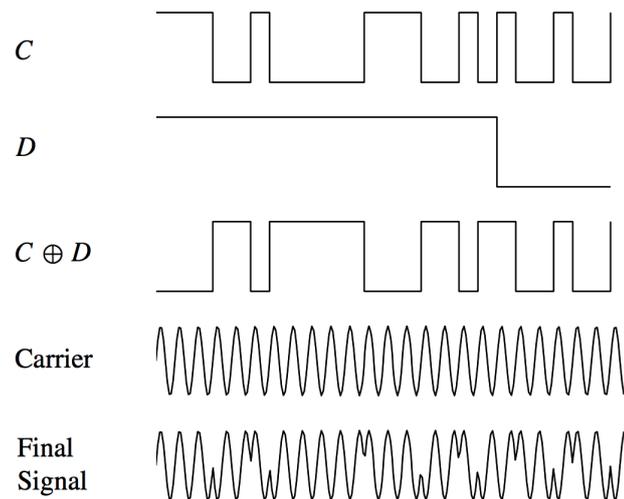


Figura 2.4. Efecto de la modulación BPSK en la portadora L1.

Las secuencias utilizadas como códigos C/A en GPS, pertenecen a una familia de secuencias muy particulares denominadas *Gold Codes* en honor de Robert Gold que las describió en 1967 [30]. Estas secuencias también son llamadas secuencias de ruido pseudo-aleatorias (*pseudo-random noise sequences*) o simplemente secuencias PRN. Las secuencias son generadas como se indica en la figura 2.5. El generador de códigos C/A contiene dos registros de desplazamiento etiquetados como G_1 y G_2 . Estos registros constan de 10 celdas a partir de las cuales se generan 2 secuencias de longitud 1.023. Estas dos secuencias se suman en modulo 2 para generar códigos C/A de longitud 1.023. Cada periodo 1.023, los registros de desplazamiento se resetean con el valor 1, haciendo que el código empiece de nuevo. El registro G_1 siempre tiene una configuración del tipo:

$$f(x) = 1 + x^3 + x^{10}$$

Ello implica que los bits de peso 3 y 10 son realimentados a la entrada. De la misma forma, el registro G_2 tiene una configuración:

$$f(x) = 1 + x^2 + x^3 + x^6 + x^8 + x^9 + x^{10}$$

Para generar los códigos C/A de los distintos satélites, las salidas de los dos registros de desplazamiento se combinan de una forma especial: del registro G_1 siempre se toma su salida, pero el registro G_2 escogen dos bits de distinto peso e introducen en un sumador en módulo 2. La selección de bits para el sumador se conoce como la selección de fase. En la tabla 2.1 puede

verse la combinación de la selección de fase para cada código C/A. También se muestran los primeros 10 bits de cada código en representación octal.

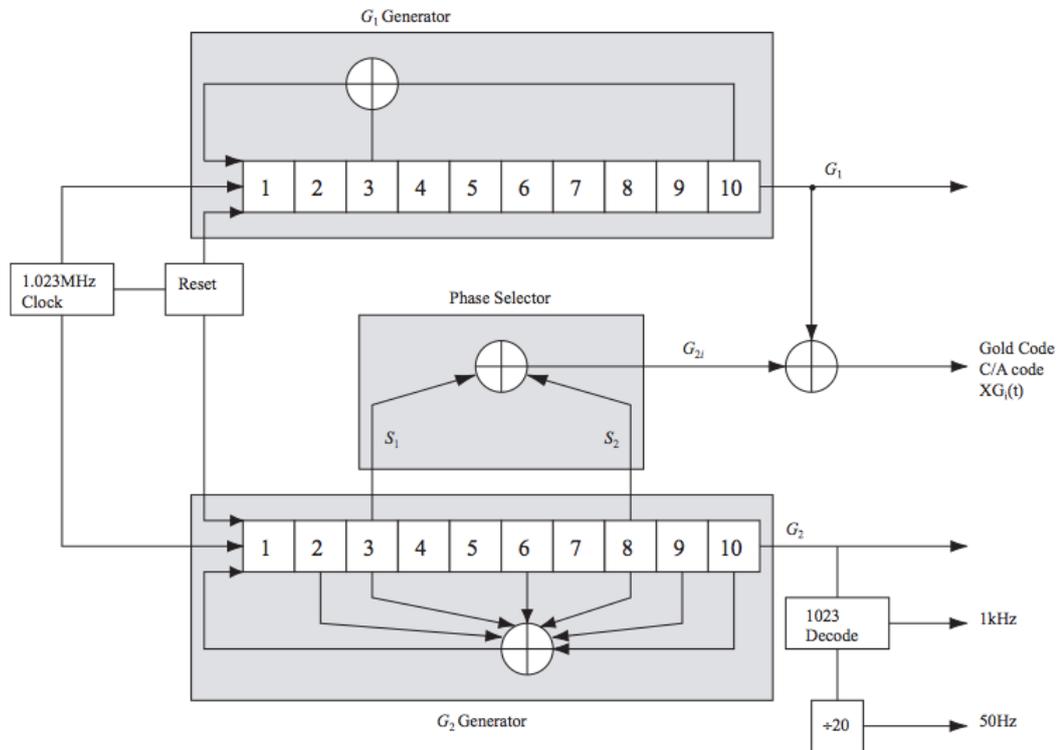


Figura 2.5. Generador de códigos C/A.

Los códigos C/A han sido seleccionados debido a sus características especiales. La más importante de estas características son las propiedades de correlación que cumplen todos los códigos C/A. Estas son:

- Prácticamente no existe correlación cruzada: Todos los códigos C/A están prácticamente incorrelados unos con otros. Dados dos códigos distintos C^i y C^k para los satélites i y k , la correlación cruzada es prácticamente nula:

$$r_{ik}(m) = \sum_{l=0}^{1022} C^i(l)C^k(l+m) \approx 0 \quad \forall m; i \neq j;$$

- Prácticamente no existe autocorrelación, excepto en el caso de que los códigos estén alineados. Esta propiedad se expresa:

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

Es decir, cuando la secuencia recibida es la misma que aquella con la que se va a correlar, y se encuentran alineadas obteniéndose un valor alto de correlación, como se muestra en la figura 2.6 izquierda. Este valor se denomina pico de auto correlación (pico de detección, DP). Además el valor obtenido en el caso de que el código recibido no corresponda con el esperado, como muestra la figura 2.6 derecha, o que siendo el código esperado no se encuentre alineado con el mismo, será un valor bajo. Este valor se denomina ruido de correlación cruzada (CN) y ruido de auto correlación (AN), respectivamente.

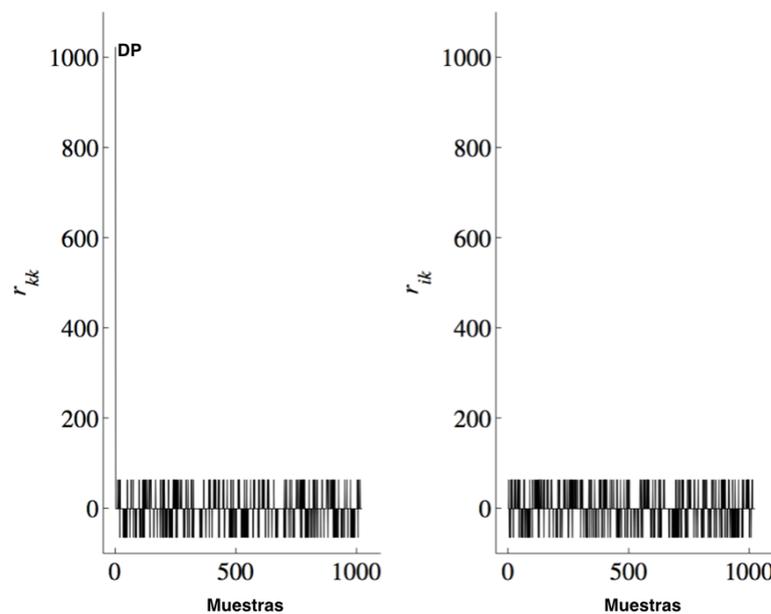


Figura 2.6. Auto-correlación (izq.) y cross-correlación (dcha.) del PRN 1.

En la práctica para detectar si un satélite es o no visible, los receptores comparan las secuencias de entrada que identifican a cada satélite recibidas en la frecuencia L1, con todas las posibles secuencias de GPS. El proceso de comparación se puede resumir en la siguiente ecuación:

$$Comp (PRN^{sat}, D^{rx}) = [a_1, a_2, \dots, a_L] \quad (2.1)$$

$$a_i = \sum_{j=1}^L PRN_{\text{mod}_L(i+j)}^{sat} * D_i^{rx}$$

Siendo L la longitud del PRN, es decir: 1023. D^{rx} es la señal de satélite GPS recibida. PRN^{sat} es el identificador del satélite GPS a comprobar.

Número ID de Satélite	Número de señal PRN GPS	Selección del <i>code phase</i>	Chips de retardo del código	Primeros 10 chips del código en octal
1	1	2⊕6	5	1440
2	2	3⊕7	6	1620
3	3	4⊕8	7	1710
4	4	5⊕9	8	1744
5	5	1⊕9	17	1133
6	6	2⊕10	18	1455
7	7	1⊕8	139	1131
8	8	2⊕9	140	1454
9	9	3⊕10	141	1626
10	10	2⊕3	251	1504
11	11	3⊕4	252	1642
12	12	5⊕6	254	1750
13	13	6⊕7	255	1764
14	14	7⊕8	256	1772
15	15	8⊕9	257	1775
16	16	9⊕10	258	1776
17	17	1⊕4	469	1156
18	18	2⊕5	470	1467
19	19	3⊕6	471	1633
20	20	4⊕7	472	1715
21	21	5⊕8	473	1746
22	22	6⊕9	474	1763
23	23	1⊕3	509	1063
24	24	4⊕6	512	1706
25	25	5⊕7	513	1743
26	26	6⊕8	514	1761
27	27	7⊕9	515	1770
28	28	8⊕10	516	1774
29	29	1⊕6	859	1127
30	30	2⊕7	860	1453
31	31	3⊕8	861	1625
32	32	4⊕9	862	1712
-	33	5⊕10	863	1745
-	34	4⊕10	950	1713
-	35	1⊕7	947	1134
-	36	2⊕8	948	1456
-	37	4⊕10	950	1713

Tabla 2.1. Asignación de fase en los códigos C/A.

La ecuación 2.1 toma en consideración todos los posibles alineamientos entre los datos de identificación de GPS recibidos y el PRN con el que se van a comparar. Este es el motivo por el que se obtendrá una matriz de valores y no un sólo valor. El tamaño de la matriz de valores $Comp (PRN^{sat}, D^{rx})$ es L , un valor para cada posible alineamiento entre los datos recibidos y el PRN comparado.

Coincidiendo con lo que se dijo anteriormente, cuando un satélite es visible, la función $Comp (PRN^{sat}, D^{rx})$ tendrá un único máximo que será el llamado pico de detección (DP). El índice en el que se encuentre este DP en la matriz de comparación se denomina la fase del código. Esta fase indica el primer bit de la secuencia PRN. El resto de valores en la matriz serán por tanto

menores que el DP. Estos valores se llamarán ruido (N). Si el satélite a comparar (PRN^{sat}) no es visible por el receptor, todos los valores de la función $Comp (PRN^{sat}, D^{rx})$ serán ruido.

Hoy en día existen muchas formas de implementar la ecuación de detección 2.1. Lo más inmediato sería trasladando esta ecuación del dominio del tiempo al de la frecuencia y utilizando la Transformada Rápida de Fourier (FFT) y su inversa (IFFT). Sin embargo en los receptores GPS existen básicamente dos técnicas de detección, que destacan por su simplicidad a la hora de ser implementadas y el poco esfuerzo computacional requerido frente a una solución basada en transformadas de Fourier.

En una de estas técnicas, la señal L1 es sobremuestreada, es decir cada bit del PRN es muestreado más de una vez, de forma que DP aumenta al aumentar la frecuencia de muestreo. En términos de la ecuación 2.1, a la hora de implementar el sobremuestreo basta con fijar el valor de L al valor adecuado.

La otra solución consiste en incrementar el periodo de muestreo. Al hacer esto, los datos de la señal GPS que se registran contienen más de una secuencia PRN completa. De esta manera el DP aumenta en proporción al incremento del periodo de muestreo, es decir, aumenta el tiempo de grabado de la señal de entrada.

Aumentar la frecuencia o el periodo de muestreo, aumenta la sensibilidad de un receptor de GPS en los casos en que la relación señal a ruido es demasiado baja. No obstante, para aplicaciones corrientes como aquellas que se dan en sistemas de navegación en cielo abierto (típico receptor para localización de vehículos) sólo se utiliza una de estas dos metodologías.

Pero además cada receptor GPS, ha de tener en cuenta el efecto Doppler que se produce en la frecuencia del satélite. Este desplazamiento de frecuencia es causado por el movimiento tanto del satélite, como del receptor. Un receptor utiliza por lo general como frecuencia intermedia, un valor entre 1 y 3 MHz, y el efecto Doppler desplaza esta frecuencia un máximo de ± 10 kHz. Con el fin de asegurar que el receptor GPS no pierde ningún satélite visible, se lleva a cabo

un escaneo de frecuencia en pasos de 500 Hz. Por lo tanto, un receptor GPS debe buscar cada código PRN en 41 frecuencias distintas. Esto hace que el número de operaciones de suma y XOR para determinar el satélite GPS visible entre todos los posibles es:

$$\begin{aligned} \text{tamaño (Sat}^{ID}) \times \text{tamaño (D}^x) \times \text{frecuencias} \times \text{satélites} &= & (2.2) \\ &= 1.373.046.048 \text{ operaciones} \end{aligned}$$

Este número requiere tiempos de CPU que no resultan prácticos para los receptores GPS comerciales en términos de esfuerzo computacional y consumo de energía. Potencialmente, existen dos formas de reducir dicho número de operaciones. Por un lado se podría reducir el número de frecuencias de búsqueda por debajo de 41. Esto tendría como consecuencia negativa la reducción de la relación señal a ruido debido al efecto Doppler, aunque repercutiría en una reducción del esfuerzo computacional directamente proporcional a dicha reducción.

Por otro lado existe la posibilidad de reducir la longitud del PRN (frecuencia de muestreo). Esto tiene como desventaja la disminución de la relación señal a ruido linealmente con la reducción de la frecuencia de muestreo, pero conlleva la ventaja de que decrece cuadráticamente el esfuerzo computacional requerido.

Esta última reflexión, nos lleva a focalizar la investigación de esta Tesis doctoral en aportaciones sobre la reducción de los códigos PRN empleados en la detección.

2.3. Conclusiones

En este segundo capítulo se ha detallado el proceso de detección de satélites en el receptor GPS. Se ha descrito cómo se generan las secuencias pseudoaleatorias PRN que identifican unívocamente a cada uno de los satélites, y cuales son las propiedades que los hacen tan interesantes de cara a la detección. Además, se plantea el problema asociado al cálculo necesario para la detección de satélites visibles. Por último se define el ámbito de investigación en el que se centran las aportaciones de esta Tesis Doctoral.

Capítulo 3

Estudio previo con aproximación determinista (GPS)

3.1. Introducción	23
3.2. Nuevo PRN de 341 bits	24
3.3. Resultados de la aproximación determinista para longitud 341.	26
3.4. Metodología determinista con secuencias de longitud no múltiplo	31
3.5. Conclusiones	39

Acrónimos

GPS	Global Positioning System
MATLAB	MATrix LABoratory
nPRN	new Pseudo Random Number
PRN	Pseudo Random Number

3.1. Introducción

Una vez definido el código PRN dentro del sistema GPS en el capítulo anterior, este capítulo se centra en la presentación de una metodología determinista que permite obtener códigos reducidos para la detección de satélites GPS. Con el objeto de probar la validez de las diversas propuestas deterministas presentadas, se hace uso del entorno de programación MATLAB.

Utilizando este entorno se presentan las funciones que modelan el proceso de generación de los códigos C/A cumpliendo las propiedades asociadas a estos. De esta forma se procede a estudiar la posibilidad de obtener nuevos códigos PRN más cortos, que mantengan las mismas propiedades de los originales. Por último se evalúa la bondad de las soluciones propuestas mediante la validación de los resultados obtenidos.

3.2. Nuevo PRN de 341 bits

Como se avanzó en los últimos párrafos del capítulo anterior, las aportaciones de esta tesis doctoral se centran en la reducción de las longitudes de los códigos PRN para identificar satélites GPS. Igualmente y como se ha explicado anteriormente, la longitud del PRN original es 1023. Este número puede ser factorizado por 3, 11 y 31. Multiplicando 31 por 11, se puede proponer 341 como la longitud del nuevo PRN. De esta forma se obtendrán 3 secuencias de 341 bits cada una, y a partir de estas se constituirá un nuevo PRN de 341 bits. Ahora bien, la forma de constituir este nuevo PRN resulta importante de cara a que mantenga las propiedades originales de correlación. Con este propósito, se proponen dos opciones que aportarán distintos beneficios. Ambas opciones se exponen a continuación.

Metodología basada en bits alternos

En este caso se divide el PRN original en 3 secuencias de 341 bits. Estas tres secuencias se construyen muestreando el PRN original a un tercio de la frecuencia, lo que significa que cada secuencia de 341 bits contendrá solamente un bit de cada tres de los correspondientes al PRN original. Estas secuencias se llamarán A, B y C, y serán creadas de la siguiente forma:

Para un PRN^{sat} dado,

$$PRN^{sat} = [a_1, a_2, a_3, \dots, a_{1023}];$$

Se formarán tres secuencias:

$$A = [a_1, a_4, a_7, \dots, a_{1021}],$$

$$B = [a_2, a_5, a_8, \dots, a_{1022}],$$

$$C = [a_3, a_6, a_9, \dots, a_{1023}]$$

A partir de ellas, una nueva secuencia de 341 bits puede ser calculado tomando el valor predominante de cada uno de los bits, de acuerdo a la tabla de verdad que se muestra en la Tabla 3.1. Este nuevo PRN (nPRN) satisface las mismas propiedades de correlación del PRN original y por tanto puede ser usado para determinar si un satélite está o no presente ahorrando un 88,9% del cálculo y la potencia utilizados anteriormente.

A	B	C	D
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Tabla 3.1. Tabla de verdad para el caso de 341 bits alternos.

Metodología basada en bits consecutivos

En este caso también se divide el PRN original en 3 secuencias de 341 bits. Pero en esta ocasión, estas se forman dividiendo el PRN original en tres partes iguales. Si se llaman a cada una de estas secuencias A, B y C, de la misma forma que en el apartado anterior, serán creadas de la siguiente forma:

Para un PRN^{sat} dado,

$$PRN^{sat} = [a_1, a_2, a_3, \dots, a_{1023}];$$

Se formarán tres secuencias:

$$A = [a_1, a_2, a_3, \dots, a_{341}],$$

$$B = [a_{342}, a_{343}, a_{344}, \dots, a_{682}],$$

$$C = [a_{683}, a_{684}, a_{685}, \dots, a_{1023}]$$

A partir de estas secuencias, de forma análoga al apartado anterior, una nueva secuencia de 341 bits puede ser calculada tomando el valor predominante de cada uno de los bits de las secuencias A, B y C, de acuerdo a la tabla de verdad que se muestra en la Tabla 3.1. Este nuevo PRN (nPRN) satisface las mismas propiedades de correlación del PRN original y por tanto puede ser usado para determinar si un satélite está o no presente en un tercio

de tiempo, ya que sólo se necesitará un tercio de los datos originales para llevar a cabo la detección.

3.3. Resultados de la aproximación determinista para longitud 341.

Para el caso en el que se genera un PRN de 341 muestras alternas, se realizará la correlación del nPRN de 341 muestras resultante de muestrear el PRN original a un tercio de la frecuencia de muestreo, y las distintas secuencias, A, B y C utilizadas para su generación. Puesto que la constelación GPS consta de 32 satélites, existirán 32 códigos nPRN de 341 bits y sus correspondientes secuencias A, B y C de 341 bits cada una. Para comprobar que la detección solamente se produce al correlar este nPRN con las secuencias A, B y C correspondientes al PRN original, es necesario llevar a cabo la correlación de cada nPRN con todas las posibles secuencias. Lo cual hace un total de 96 correlaciones por satélite.

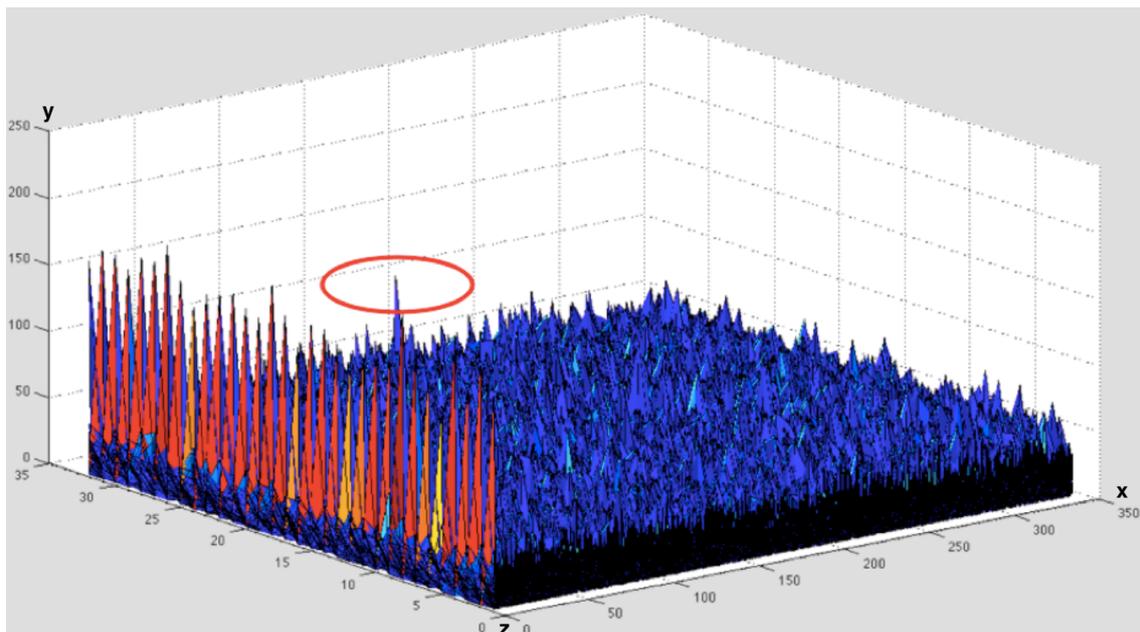


Figura 3.1: Correlaciones entre el nPRN y los tramos “A” de los 32 satélites.

En la figura 3.1 se muestra gráficamente el resultado de las correlaciones de los 32 nPRN con todas las secuencias ‘A’. En el eje ‘z’ se representan los 32 satélites de la constelación GPS. El eje ‘x’ muestra las 341 muestras que se obtienen de cada correlación, y el eje ‘y’ representa el valor de cada uno de estos puntos. Observando la figura, se comprueba como para la muestra 1 de cada nPRN, en la que las secuencias están correctamente alineadas, aparece

un pico que se destaca del resto notablemente, indicando que se produce detección. En el resto de casos los picos no pasan del valor 100, a excepción del caso concreto del satélite 23, que como se muestra marcado con un óvalo rojo en la figura, llega al valor 105 al correlarlo con el 31. Aun así el pico de detección de este satélite se encuentra en 173, por lo que existe bastante margen para discernir cuándo se produce detección o no.

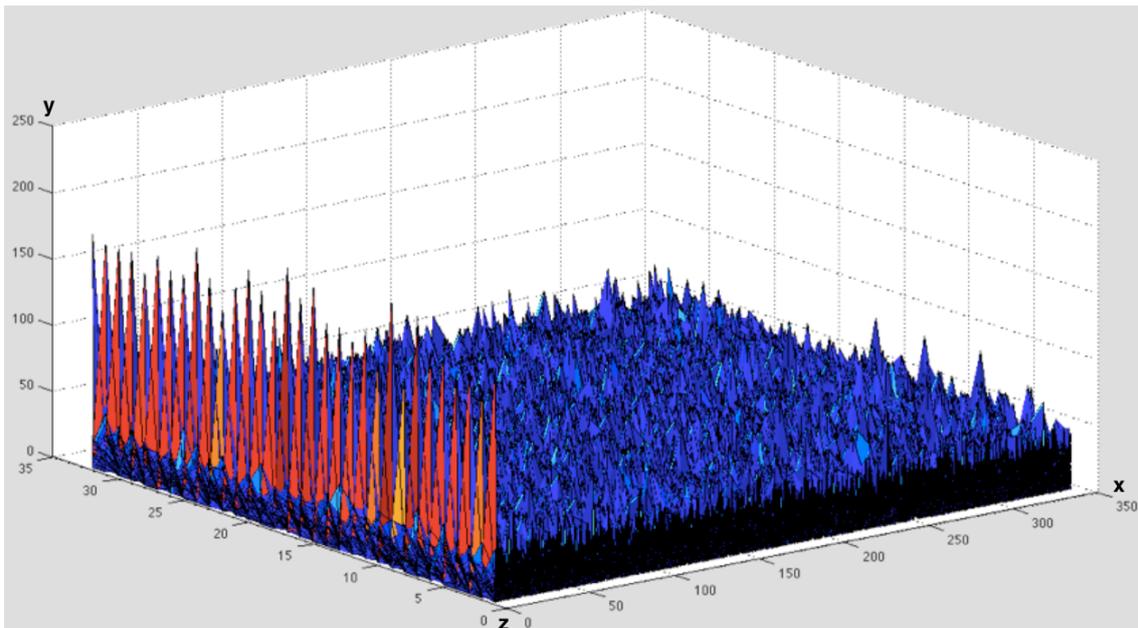


Figura 3.2: Correlaciones entre el nPRN y los tramos “B” de los 32 satélites.

De la misma forma la correlación con los tramos ‘B’ y ‘C’ presentadas en las figuras 3.2 y 3.3 muestra como los picos de detección aparecen para cada uno de los satélites. Cuando las secuencias están alineadas obteniéndose valores muy por encima de 100, que se diferencian claramente de los que constituyen el ruido producido cuando las secuencias no están alineadas o no pertenecen al satélite en cuestión.

La tabla 3.2 resume todos los datos presentados en las figuras 3.1, 3.2 y 3.3. Para cada satélite se enumera el pico de detección máximo y mínimo, así como el ruido máximo y medio en caso de detección (autocorrelación). En esta tabla 3.2 también se presenta el ruido máximo y medio en caso de no detección (correlación cruzada).

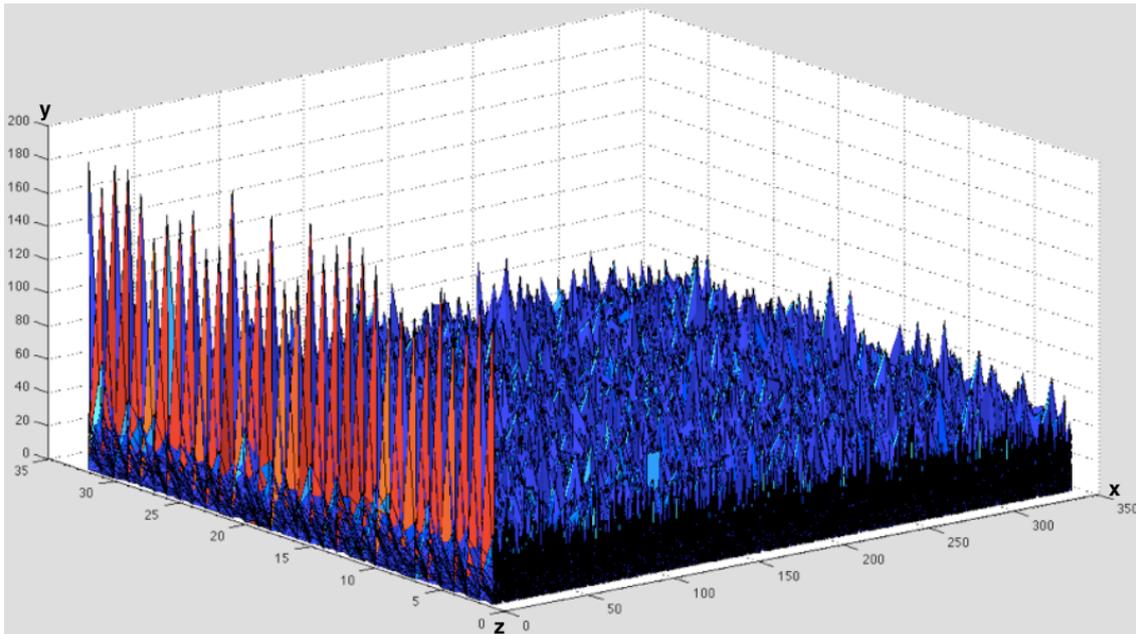


Figura 3.3: Correlaciones entre el nPRN y los tramos “C” de los 32 satélites.

Satélite	DETECCIÓN				NO DETECCIÓN	
	Pico Max.	Pico Min.	Ruido Max.	Ruido Medio	Ruido Max.	Ruido Medio
1	177	161	77	14,884	79	14,791
2	181	157	69	14,292	89	14,728
3	169	161	57	15,516	83	14,746
4	189	157	81	15,080	87	14,739
5	181	133	73	15,349	83	14,775
6	173	157	61	15,268	89	14,735
7	197	157	63	13,721	89	14,791
8	201	145	81	15,390	85	14,746
9	201	153	69	15,229	93	14,714
10	181	157	65	15,174	75	14,721
11	189	157	63	15,168	95	14,685
12	193	145	71	14,284	87	14,772
13	185	161	69	15,176	87	14,600
14	177	169	57	14,556	93	14,801
15	193	169	59	15,141	85	14,866
16	181	149	53	14,429	79	14,782
17	201	153	63	15,025	83	14,748
18	189	165	63	15,609	73	14,821
19	177	161	59	14,792	87	14,766
20	189	157	67	14,470	79	14,768
21	197	173	63	14,241	79	14,623
22	169	153	63	14,845	81	14,712
23	173	157	61	15,068	105	14,684
24	193	153	61	15,437	91	14,660
25	169	169	51	14,143	77	14,691
26	193	169	67	15,921	77	14,712
27	177	153	71	14,780	85	14,679
28	177	161	65	14,707	79	14,613
29	189	165	67	14,639	81	14,727
30	189	173	59	14,964	97	14,739
31	173	173	63	14,703	93	14,762
32	185	161	73	14,441	97	14,751

Tabla 3.2. Valores de correlaciones con nPRN de 341 bits muestreados a 1/3 f.

Como se muestra en la Tabla 3.2, para todos los casos se obtienen unos valores de autocorrelación entre 201 y 133 situándose el ruido en torno a 15, lo cual posibilita la detección para todos los satélites. Por otra parte, en los casos de no detección, el ruido se mantiene en torno a 14 con picos máximos por debajo de 100, a excepción del caso ya mencionado para el satélite 23 en el que se alcanza el valor 105 al correlarlo con el 31.

Para el caso de que el nuevo PRN de 341 bits se genere a partir de muestras contiguas del PRN original utilizando la misma frecuencia de muestreo, se obtienen unos resultados parecidos. De la misma forma que antes, se realiza la correlación de cada uno de los nPRN con las secuencias A, B y C de cada uno de los satélites. Los resultados obtenidos se representan gráficamente en las figuras 3.4, 3.5 y 3.6. Se observa que al igual que para el caso anterior, los picos correspondientes a la detección aparecen en los casos en que las secuencias corresponden al mismo satélite y están alineadas. Los valores obtenidos en estas condiciones de detección siguen manteniéndose por encima de 130. Para el caso en que no se produce detección, los valores máximos de ruido se mantienen por debajo de 100 facilitando la tarea de detección.

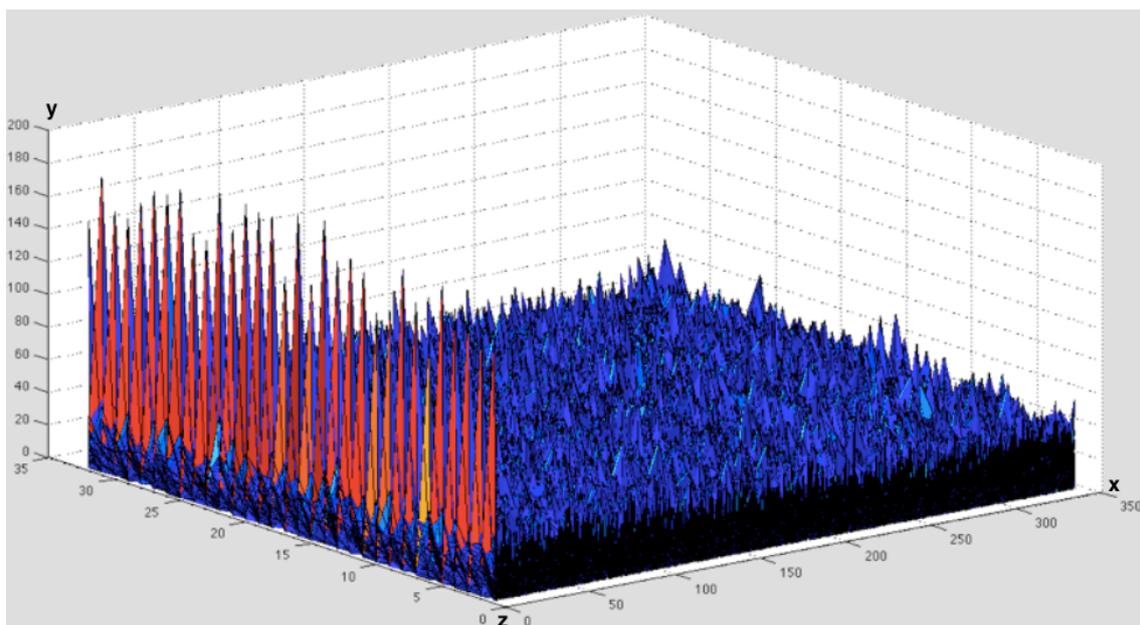


Figura 3.4. Correlaciones entre el nPRN y los tramos “A” de los 32 satélites.

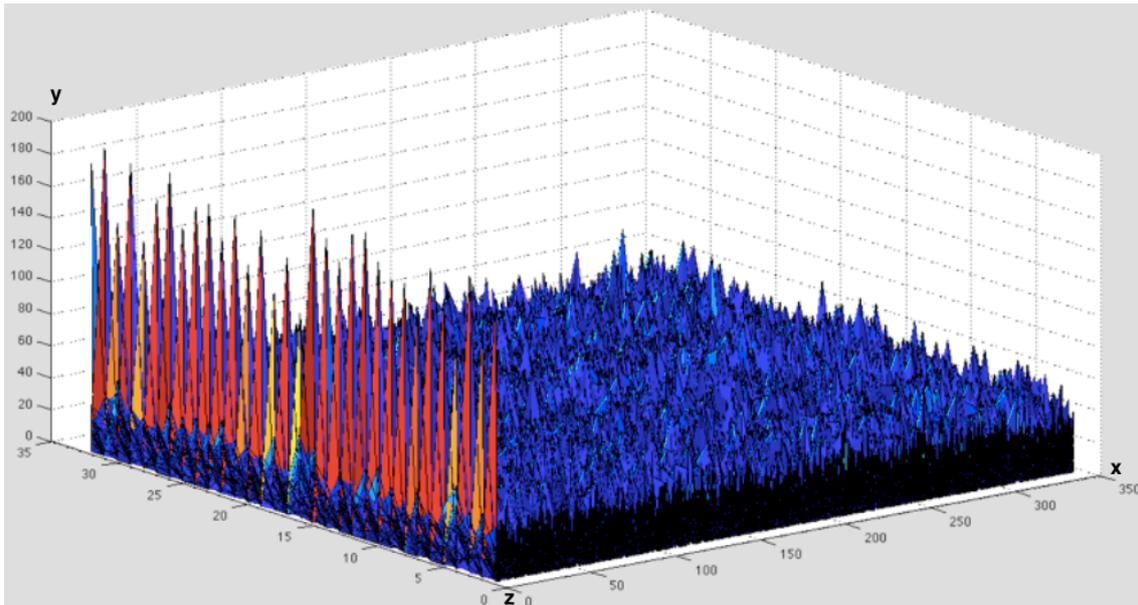


Figura 3.5. Correlaciones entre el nPRN y los tramos “B” de los 32 satélites.

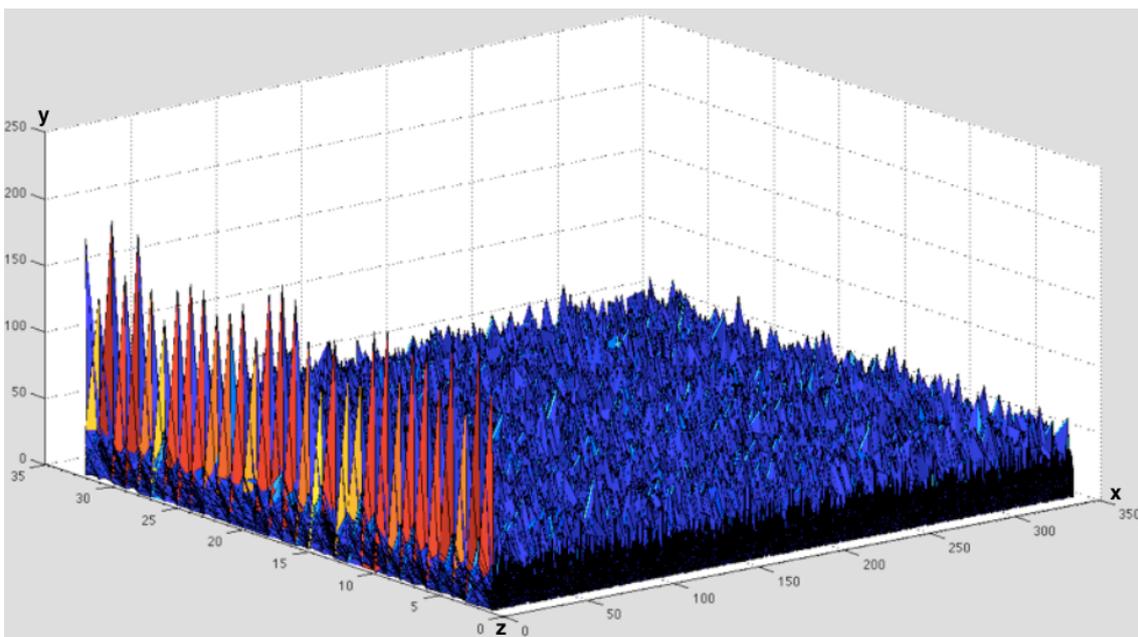


Figura 3.6. Correlaciones entre el nPRN y los tramos “C” de los 32 satélites.

La tabla 3.3 presenta los mismos parámetros que la tabla 3.2, pero ahora aplicados a esta nueva metodología basada en secuencias contiguas. En este caso, como se puede observar en la tabla 3.3, bajan sensiblemente los valores de ruido para el caso de la detección, y los valores de los picos se mantienen en valores siempre por encima de 130. Este hecho hace aun más sencilla la tarea de determinar cuando se produce detección y cuando no, mediante un sencillo detector de nivel, por ejemplo. Para los casos en los que no existe

detección, también se obtienen unos valores de ruido máximo ligeramente inferiores a los obtenidos en el caso anterior, lo cual hace que baje sensiblemente la media de ruido.

3.4. Metodología determinista con secuencias de longitud no múltiplo

Esta misma metodología se utilizó para generar nuevos códigos PRN de 512, 256 y 128 bits contiguos y alternos. Lógicamente cuanto más pequeño sea el nuevo código, más rápida será la detección. Sin embargo, como se podrá comprobar a continuación, no todo son ventajas, ya que a menor longitud, mayor será el ruido, y por tanto más complicado será llevar a cabo la detección.

Satélite	DETECCIÓN				NO DETECCIÓN	
	Pico Max.	Pico Min.	Ruido Max.	Ruido Medio	Ruido Max.	Ruido Medio
1	175	163	47	12,296	83	14,617
2	201	151	65	15,974	79	14,531
3	187	153	63	14,017	79	14,636
4	197	143	57	15,547	81	14,540
5	183	161	59	13,221	77	14,690
6	183	145	63	14,650	81	14,591
7	187	157	63	13,394	75	14,541
8	185	157	63	14,552	83	14,552
9	185	157	53	13,235	75	14,537
10	181	153	59	13,750	73	14,587
11	193	143	61	15,013	77	14,597
12	189	141	67	15,539	75	14,512
13	177	165	67	12,562	81	14,520
14	199	131	71	17,339	75	14,458
15	197	157	65	15,315	73	14,568
16	197	129	63	17,431	85	14,453
17	193	157	63	13,964	73	14,621
18	191	137	75	16,421	73	14,656
19	191	147	57	15,333	75	14,553
20	193	149	53	14,947	71	14,531
21	177	159	55	12,686	85	14,583
22	195	155	65	15,284	79	14,447
23	179	163	51	13,198	81	14,576
24	175	165	63	13,519	75	14,690
25	189	159	61	13,254	71	14,618
26	191	137	67	16,152	85	14,527
27	183	157	59	13,201	77	14,486
28	195	143	73	16,176	75	14,434
29	189	161	53	13,237	75	14,561
30	199	149	63	15,933	71	14,531
31	193	137	77	16,298	81	14,447
32	181	151	57	13,739	77	14,566

Tabla 3.3. Valores de correlaciones con nPRN de 341 bits muestreados a f.

Para los casos en que el nuevo PRN se constituye a partir de 512 bits, solamente se presentan resultados para los casos de detección. Esto es debido a que, tal y como se explicó en el capítulo anterior, es necesario comprobar la correlación del nPRN con la secuencia continua de 1.023 posibles desplazamientos de fase de 1.023 bits ya que, al tomar secuencias de 512 bits siempre se va tomando un bit del siguiente tramo del PRN. Esto ocasiona que en determinados casos, aún tratándose del PRN correcto, no se produzca detección al no estar correctamente alineadas las secuencias.

En la figura 3.7 se pueden comprobar gráficamente los resultados obtenidos para el caso del primer PRN de 512 bits. En la figura se representa el número secuencia en el eje 'x' con la que se está realizando la correlación del nPRN de 512 bits obtenido a partir del muestreo del PRN original a la mitad de su frecuencia. Tal y como se explicó anteriormente, debido a la rotación necesaria para obtener un PRN de 1.024 bits, es preciso comprobar la correlación del nPRN con 2044 secuencias. En el eje 'y' se representa en rojo el valor máximo obtenido en la correlación de cada una de estas 2.044 secuencias con el nPRN del satélite 1. Este valor corresponde al valor de pico máximo cuando se produce la detección. En verde se presenta el valor de ruido máximo correspondiente a cada caso. Como se puede observar en la figura, aun siendo todos los códigos pertenecientes al satélite 1, a medida que se pierde la alineación el valor del pico de detección baja y a su vez el valor del ruido máximo sube. En la parte de la izquierda de la tabla 3.4, correspondiente al caso en que el nPRN se genera muestreando el PRN original a la mitad de frecuencia, se puede comprobar como el valor de pico máximo pasa de 258 en el mejor caso, a solo 104 para el caso en que la alineación de las secuencias es más desfavorable. Aunque se puede comprobar que no en todos los casos se produce la detección, los resultados son positivos ya que esta se produce en un alto índice de casos.

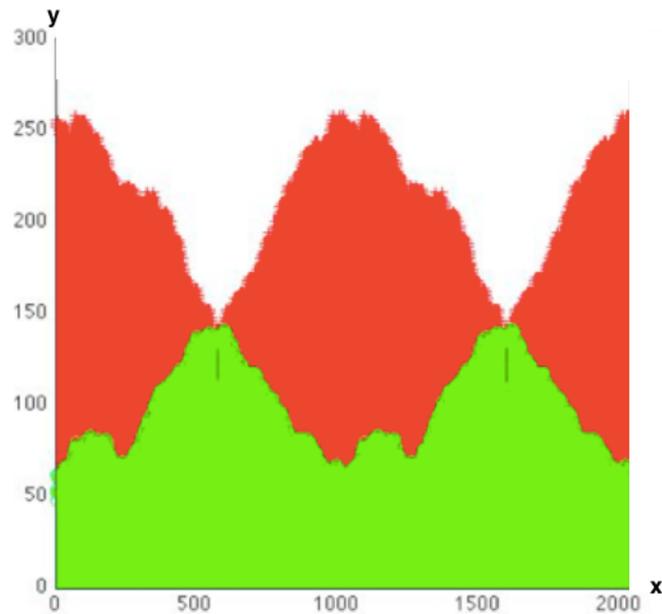


Figura 3.7. Relación señal a ruido para el nPRN 1 de 512 bits muestreados a $\frac{1}{2} f$.

Para el caso correspondiente al nPRN de 512 bits generado muestreando el PRN original a la misma frecuencia, se procede de la misma forma, obteniendo los resultados que se muestran en la figura 3.8. Como se puede observar los resultados son muy parecidos al caso anterior, para determinados satélites, como el primero, los valores de pico mejoran sensiblemente pasando de 258 y 104 a 280 y 130 respectivamente, aunque para otros, como el tercero, estos valores son algo peores, pasando de 286 y 116 a 258 y 110. En general el ruido se mantiene constante en casi todos los casos, en torno a 17, exceptuando el caso del satélite 12 que llega a 21 en el primer caso y del 18 que llega a 20 utilizando el segundo método. Aun así los resultados obtenidos siguen siendo muy positivos de cara a la detección en ambos casos.

Al reducir la longitud del nPRN hasta 256 bits, la dificultad a la hora de la detección aumenta, ya que no solo se reduce la longitud del nPRN, sino que además aumenta el número de secuencias a partir de las que generar este código. En este caso, el PRN original de 1.024 bits ($1.023 + 1$) se muestrea a $\frac{1}{4}$ de su frecuencia, obteniendo 4 secuencias: A, B, C y D de 256 bits cada una. Los bits de estas secuencias se combinan de forma que se obtenga un código nPRN de 256 bits que muestre un pico de valor alto al ser correlado con cualquiera de estas cuatro secuencias.

nPRN de 512 bit muestreado a 1/2 de f				nPRN de 512 bit muestreado a f		
Satélite	Pico Max.	Pico Min.	Ruido Medio	Pico Max.	Pico Min.	Ruido Medio
1	258	104	16,362	280	130	16,321
2	306	112	18,190	280	106	19,817
3	286	116	18,170	258	110	18,209
4	286	118	16,207	300	114	17,563
5	286	114	17,915	252	116	16,035
6	266	102	17,134	264	110	16,577
7	278	106	17,349	284	102	18,677
8	268	124	17,358	270	122	16,848
9	268	108	17,363	288	130	16,545
10	262	126	16,739	308	134	17,201
11	278	120	16,468	264	102	17,046
12	290	102	21,174	278	126	17,483
13	260	102	16,123	280	124	16,316
14	270	116	17,786	282	108	17,043
15	288	114	17,409	256	102	16,302
16	262	114	16,483	284	102	18,211
17	288	114	16,439	300	152	16,266
18	286	120	16,734	264	102	20,343
19	266	114	16,951	262	108	17,467
20	264	122	16,008	282	102	18,377
21	296	148	16,515	264	114	14,854
22	244	106	16,018	268	116	17,293
23	258	112	15,618	248	102	15,488
24	296	106	17,705	284	128	16,831
25	266	104	18,315	296	112	16,975
26	282	116	16,616	294	102	19,085
27	244	116	16,327	298	130	17,433
28	270	114	16,843	268	102	17,096
29	286	102	18,503	260	120	15,809
30	284	102	19,895	280	112	18,667
31	264	122	15,295	284	134	16,864
32	254	104	15,889	276	120	16,041

Tabla 3.4. Valores de detección para un nPRN de 512 bits.

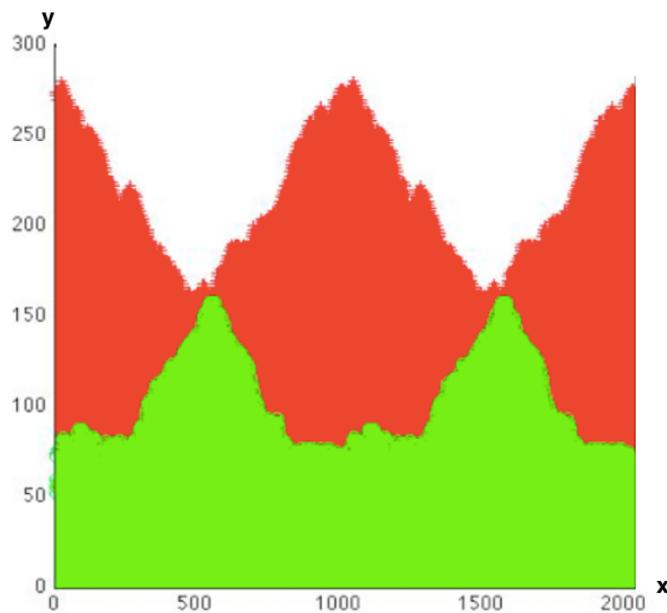


Figura 3.8. Relación señal a ruido para el nPRN 1 de 512 bits muestreado a f.

Como se representa en el eje 'y' de la figura 3.9, existen 4.088 posibles secuencias con las que correlar cada uno de los nPRN. De igual forma que para el caso anterior, en la figura, se presentan en rojo los valores máximos de pico de cada una de las correlaciones entre las 4.088 secuencias y el nPRN 1. En verde se representa el valor del ruido máximo para cada caso. Se comprueba como al tratarse de 4 secuencias de 256 bits, los problemas asociados a la alineación anteriormente comentados se hacen notar a medida que las secuencias se van desalineando. En concreto, observando la zona izquierda de la tabla, se obtiene un valor de pico máximo de 122 y un pico mínimo de 74 en el caso menos favorable de alineación.

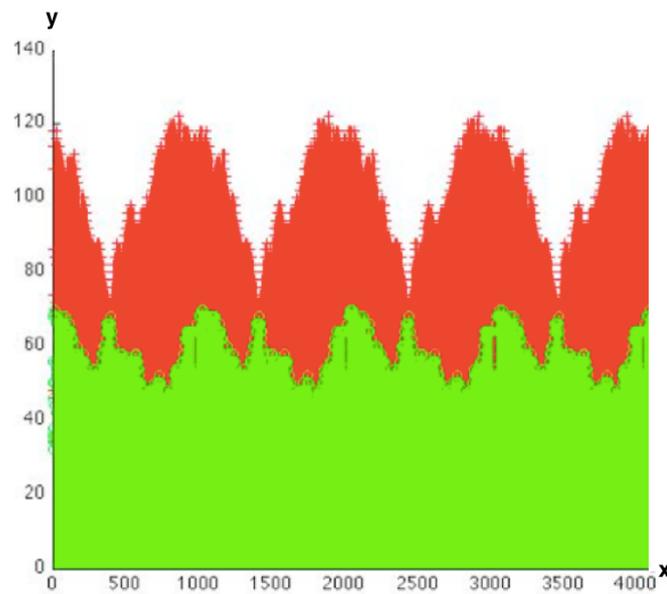


Figura 3.9. Relación señal a ruido para el nPRN 1 de 256 bits muestreados a $\frac{1}{4} f$.

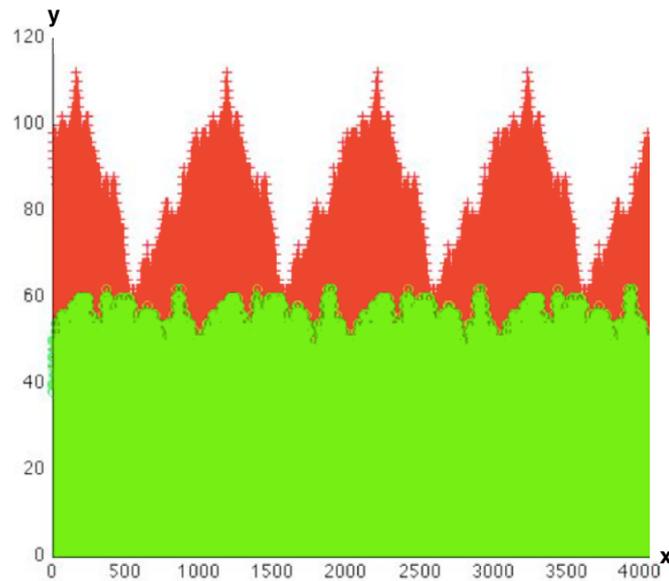


Figura 3.10. Relación señal a ruido para el nPRN 1 de 256 bits muestreados a f.

Aunque los valores disminuyen en magnitud comparados con los obtenidos para el caso de un nPRN de 512 bits, los resultados son positivos, ya que el margen existente entre los picos de detección y el ruido es amplio para la gran mayoría de los casos.

En el caso en el que el nPRN de 256 bits se obtiene muestreando el PRN original a la misma frecuencia, o lo que es lo mismo, dividir el PRN de $1.023 + 1$ muestras en 4 secuencias de 256 bits contiguos, se puede observar el mismo efecto que para el caso de 512 bits: los valores mejoran sensiblemente para algunos satélites, mientras que para otros son ligeramente más bajos. En ningún caso estas variaciones son dramáticas, por lo que se pueden considerar ambas opciones como exitosas y aptas para la detección. Se puede observar también en la tabla 3.5 que el ruido medio no presenta grandes variaciones manteniéndose en todo momento por debajo de 10.

Por último, se presentan los resultados para los casos en los que se obtiene el nPRN de 128 bits. En este caso el nPRN de 128 bits se obtiene a partir de 8 secuencias, por lo que, como es de esperar, el problema de la alineación y la sintetización de 8 secuencias en una de 128 bits se harán notar.

nPRN de 256 bit muestreado a 1/4 de f				nPRN de 256 bit muestreado a f		
Satélite	Pico Max.	Pico Min.	Ruido Medio	Pico Max.	Pico Min.	Ruido Medio
1	122	74	8,119	112	64	7,370
2	114	66	7,766	110	62	7,962
3	112	64	7,883	108	60	9,552
4	116	68	8,662	128	80	8,462
5	124	76	8,563	104	56	7,715
6	102	54	8,810	108	60	7,416
7	118	70	8,259	138	90	7,647
8	110	62	8,432	132	84	8,535
9	114	66	8,588	118	70	7,727
10	124	76	6,754	134	86	7,919
11	126	78	7,111	116	68	7,364
12	126	78	8,330	120	72	7,801
13	116	68	7,223	106	58	8,538
14	126	78	8,698	114	66	7,724
15	106	58	8,254	118	70	7,467
16	108	60	7,601	108	60	8,894
17	110	62	8,358	124	76	7,260
18	118	70	7,947	116	68	8,950
19	108	60	8,452	116	68	8,689
20	120	72	6,892	118	70	7,623
21	130	82	7,461	116	68	7,596
22	104	56	8,180	120	72	8,337
23	116	68	6,932	108	60	7,967
24	122	74	9,604	114	66	7,928
25	124	76	9,261	112	64	7,720
26	112	64	8,435	122	74	8,930
27	102	54	7,855	124	76	7,906
28	98	50	8,408	114	66	7,764
29	144	96	7,964	102	54	8,041
30	106	58	7,917	108	60	8,845
31	126	78	7,616	110	62	7,586
32	104	56	8,032	112	64	7,549

Tabla 3.5. Valores de detección para un nPRN de 256 bits.

Para el caso en el que las secuencias se conforman muestreando el PRN original de 1.023 bits a $\frac{1}{8}$ de la frecuencia original, es necesario llevar a cabo 8.176 correlaciones de secuencias de 128 bits para cada satélite con el fin de comprobar que es posible la detección del satélite. En la figura 3.11, se puede observar como el problema de alineación y de reducción en la longitud de las secuencias se hacen patentes. Como en casos anteriores, se representan en rojo los picos máximos para el caso de detección, y en verde los picos máximos de ruido para cada una de las 8.176 correlaciones. Como se puede observar en la figura, y comprobar en la parte izquierda de la tabla 3.6, el valor de pico máximo para el nPRN 1 es de 52 mientras que el mínimo es de solo 22. Teniendo en cuenta que se registran valores máximos de ruido por encima de

35, por el momento la detección utilizando códigos nPRN generados muestreando a $\frac{1}{8}$ el PRN original no es posible.

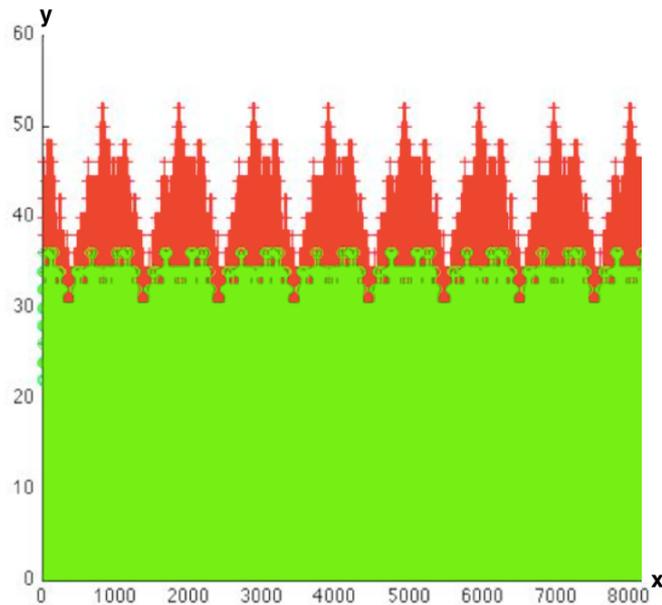


Figura 3.11. Relación señal a ruido para el nPRN 1 de 128 bits muestreados a $\frac{1}{8}$ f.

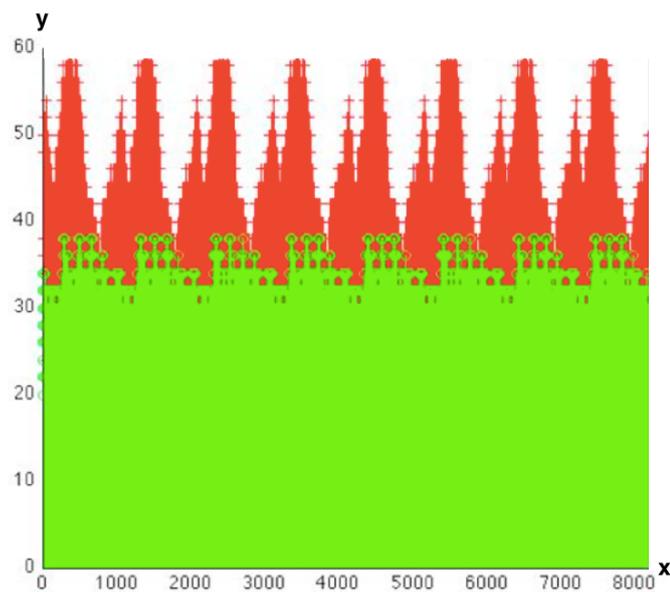


Figura 3.12. Relación señal a ruido para el nPRN 1 de 128 bits muestreados a f.

En el caso en el que el nPRN de 128 bits se obtiene muestreando el PRN original a la misma frecuencia, se obtienen resultados muy parecidos al resto de casos. Los valores varían sensiblemente en el caso de los picos de detección, mientras que el valor medio de ruido permanece invariable. No obstante, puesto que los valores máximos de ruido se siguen situando por

encima de 35, tampoco es posible utilizar este método para la detección de satélites por el momento.

nPRN de 128 bit muestreado a 1/2 de f				nPRN de 128 bit muestreado a f		
Satélite	Pico Max.	Pico Min.	Ruido Medio	Pico Max.	Pico Min.	Ruido Medio
1	52	22	8,469	60	22	8,120
2	52	24	8,923	56	22	8,577
3	50	24	9,003	50	24	9,069
4	46	22	8,980	54	24	8,439
5	48	22	8,939	52	22	8,735
6	56	22	8,650	70	22	8,116
7	54	24	8,464	56	26	8,508
8	58	24	8,957	50	22	8,627
9	50	22	8,775	56	22	8,289
10	52	22	8,626	58	24	7,940
11	52	26	8,827	52	22	8,720
12	54	24	8,446	56	22	8,301
13	60	24	8,701	50	22	8,575
14	58	22	8,851	56	22	8,609
15	44	22	9,306	52	22	8,271
16	58	24	8,479	56	24	8,570
17	54	22	8,922	56	24	8,007
18	62	24	8,650	54	24	8,487
19	60	24	8,524	52	22	8,599
20	48	24	8,810	54	26	8,741
21	56	24	8,532	66	20	8,249
22	48	24	9,444	54	22	8,744
23	48	24	8,997	46	24	8,799
24	58	20	8,591	62	24	8,022
25	50	26	8,954	46	26	9,118
26	48	22	9,109	56	20	8,988
27	50	22	8,746	54	24	8,932
28	64	22	8,756	48	24	8,991
29	56	22	8,450	60	22	8,425
30	52	22	9,078	52	24	9,210
31	48	24	8,906	54	22	8,771
32	54	22	8,528	56	24	8,454

Tabla 3.6. Valores de detección para un nPRN de 128 bits

3.5. Conclusiones

En este tercer capítulo el trabajo se ha centrado por una parte en la obtención con una metodología determinista de códigos ortogonales de menor longitud que proporcionen un ahorro computacional significativo a la hora de la detección de satélites con respecto a los códigos preestablecidos de longitud 1023. Para ello, inicialmente se han generado 32 códigos de 341 bits, uno para cada satélite GPS, obtenidos a partir de las secuencias originales. Estos códigos han demostrado cumplir las propiedades de correlación propias de los

códigos originales, haciendo posible la detección de los satélites utilizando tan solo un 11% de cómputo que en el caso tradicional.

A continuación se ha extendido esta idea para el caso real, en el que se recibe una secuencia continua en la que se repite un determinado código de 1023 bits. Tomando tramos de 1024 en esta secuencia, y procediendo de forma análoga al caso de 341, se han generado 32 nuevos códigos de 512 bits correspondientes a cada uno de los satélites GPS y se ha comprobado que cumplen las propiedades originales de correlación, con lo que también posibilitan la detección utilizando la mitad de cómputo con respecto al método original de detección. Sin embargo, los resultados revelaron la importancia del alineamiento en las secuencias con longitudes que no son múltiplos de 1023, ya que para determinados casos el pico de detección se acerca peligrosamente al nivel de ruido máximo.

Una vez comprobado el funcionamiento de esta idea con secuencias de 512 bits, se llevó a cabo el mismo procedimiento para códigos de 256 bits de longitud constatando que también son aptos para la detección, a pesar de que los problemas debidos a la falta de alineación de las secuencias se hacen notar con más frecuencia. Aun así, salvando estos casos, es posible realizar la detección de los satélites utilizando sólo un cuarto de cómputo con respecto a la metodología tradicional.

Por último se llegó a aplicar esta idea en secuencias de longitud 128. Obteniendo códigos PRN que, aunque conservaban las propiedades de correlación de los códigos PRN originales, presentaban dificultades a la hora de ser empleados en casos prácticos debido a la baja relación señal a ruido obtenida en las correlaciones.

Otro de los objetivos de este primer capítulo fue la obtención de códigos ortogonales de menor longitud que los originales, que permitan un aumento en la velocidad de detección de los satélites GPS. Para ello se ha partido de la idea desarrollada inicialmente generándose 32 códigos de 341 bits a partir de los PRN originales. Estos códigos también han demostrado cumplir las

propiedades de correlación propias de los códigos originales reduciendo la velocidad de detección a un tercio respecto a la original.

Desarrollando la idea para secuencias de longitudes múltiplo de 1.024 se han obtenido 32 secuencias de 512, 256 y 128 bits que cumplen también las propiedades de correlación de los códigos PRN originales de 1.023 bits, y que por tanto permiten la detección de satélites en la mitad, un cuarto y un octavo de tiempo. De la misma forma que en los casos anteriores, a medida que se reduce la longitud del nuevo código PRN aumentan los problemas asociados al alineamiento, por lo que los códigos de longitud 128 no son del todo aptos para un uso práctico a tenor de los resultados obtenidos empleando esta metodología determinista.

Ante los resultados expuestos queda patente que mediante las técnicas detalladas en este capítulo, es posible realizar la detección de satélites con un importante ahorro de computo y potencia respecto de las técnicas convencionales. Dependiendo de la técnica elegida y los requerimientos de la aplicación, se puede llegar a reducir la potencia requerida hasta en un cuarto con respecto a la necesaria para un PRN original de 1.023 bits.

Por otra parte se ha demostrado que formando las secuencias de bits tomando las muestras de forma continua, es posible reducir el tiempo de detección en hasta un cuarto del tiempo necesario para la detección utilizando un PRN convencional de 1.023 bits.

Capítulo 4

Aproximación basada en Algoritmos Genéticos

4.1. Introducción	42
4.2. Nuevos códigos PRN	47
4.3. Resultados de los métodos presentados	52
4.4. Introducción de nuevas variables de optimización	68
4.5. Experimentos	70
4.6. Conclusiones	71

Acrónimos

ADN	Ácido DesoxirriboNucleico
GA	Genetic Algorithm
GPS	Global Positioning System
NSGA	Nondominated Sorting Genetic Algorithm
PRN	Pseudo Random Number

4.1. Introducción

A partir de la metodología expuesta anteriormente, y para tratar de mejorar los resultados obtenidos mediante los distintos métodos deterministas presentados, se propone emplear algoritmos genéticos con el fin de encontrar

un nuevo código PRN para cada satélite. A lo largo de este capítulo se exploran distintas técnicas y metodologías de generación en las que se varían distintos parámetros que permiten obtener códigos de detección con diversas características.

Los algoritmos genéticos son métodos heurísticos de búsqueda de optimización inspirados en el proceso natural de la evolución. Desde el cuerno de un rinoceronte hasta la luz que algunas especies marinas emiten en los fondos abisales, se puede observar cómo la naturaleza es capaz de lograr soluciones ingeniosas para problemas de optimización física. Siguiendo esta tendencia, la ciencia computacional ha tratado de replicar la forma de proceder de la naturaleza por medio de los Algoritmos Genéticos. Este tipo de algoritmos tratan de modelar los componentes básicos del método evolutivo que sigue la naturaleza: la codificación de organismos como estructuras de datos genéticos (ADN), así como la recombinación, mutación y persistencia de las características deseables. En este tipo de técnicas entran en juego la selección natural, donde solamente los individuos con mejores características y mayor adaptabilidad sobreviven, la cooperación y competición entre las distintas estrategias de supervivencia, y también el siempre presente rol de la oportunidad.

La idea general detrás de los algoritmos genéticos es la evolución hacia la mejor solución posible a partir de elementos aleatorios, pasando por diferentes generaciones de individuos. Un algoritmo genético funciona a partir de un conjunto de individuos que son inicialmente seleccionados a partir del conjunto de posibles soluciones al problema a resolver. Este conjunto de individuos se denomina población.

Con el fin de comprender el funcionamiento de los algoritmos genéticos veamos cómo se comportarían a la hora de resolver un problema sencillo consistente en encontrar el máximo de una función coste. La función coste representa el problema que se va a resolver. En la mayoría de las ocasiones los problemas a resolver estarán enfocados a encontrar el valor óptimo de una función que satisfaga un conjunto de restricciones. La función representa la relación entre los distintos parámetros que se busca optimizar. Una vez que

estas relaciones están bien definidas y son lo suficientemente sencillas para ser modeladas matemáticamente, se deben aplicar métodos matemáticos analíticos. En los casos en los que estas relaciones son tan complejas que parecen ser impredecibles o aleatorias, solamente las búsquedas aleatorias o exhaustivas ofrecen alguna opción para encontrar una solución válida. Los algoritmos genéticos, son adecuados para los problemas del mundo real que se encuentran entre estos dos extremos.

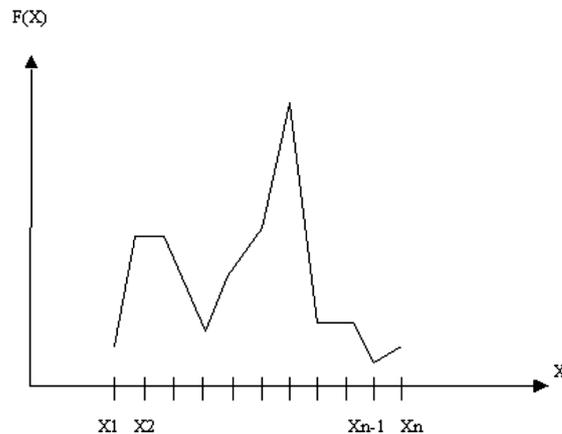


Figura 4.1. Ejemplo de una función coste.

Los algoritmos genéticos nos ayudarán a encontrar el individuo X_{\max} que maximice la función $F(X_i)$ entre todos los posibles X_i . Cada uno de los valores X_i representa un individuo, que es codificado como una concatenación de números binarios, que equivaldrán a genes de ese individuo. Para solucionar los problemas mediante algoritmos genéticos, se utiliza el esquema representado en la figura 4.2.

Como se ve en la figura 4.2, en un principio se genera una población. Es decir que aleatoriamente se elige un conjunto de individuos en el espacio de parámetros (X_i). A continuación cada individuo de la población se asocia con su correspondiente punto de la función coste, $F(X_i)$. Algunos individuos de la población son mejores que otros. De esta forma, se deben elegir los individuos atendiendo a su idoneidad con respecto a la función coste.

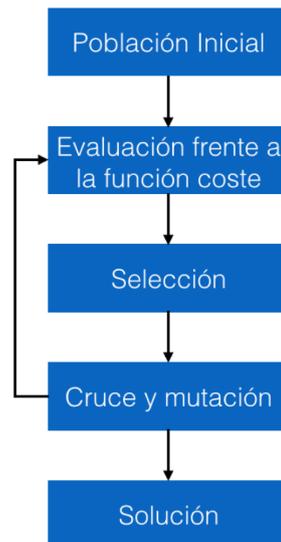


Figura 4.2. Esquema seguido por los algoritmos genéticos.

Una vez elegidos los individuos más idóneos de esa generación, se utilizarán dos operadores básicos para dar lugar a una nueva generación que permita obtener una mejor solución al problema de encontrar el máximo global. La nueva generación se vuelve a probar frente a la función coste, con el fin de encontrar el valor máximo de la función. En cada generación se genera una nueva población con la que se repetirá el proceso. Este proceso de generación continúa una y otra vez, sin que resulte sencillo saber cuando se debe detener. En esta decisión intervienen distintos parámetros que deberán ser tenidos en cuenta con el fin de localizar el máximo global y no quedarse en un máximo local.

El tamaño de la población inicial es un aspecto importante puesto que una población grande permite obtener una muestra efectiva del espacio paramétrico. Sin embargo a mayor población, mayor coste computacional, por lo que es necesario encontrar un compromiso entre ambos. Con el fin de mantener una muestra efectiva y significativa, así como un coste computacional razonable, una opción interesante es reducir la población después de la primera iteración del proceso. Por ejemplo, tomar una población inicial de 1.000 individuos, luego elegir los 500 mejores individuos y a partir de ahí trabajar con este tamaño de población. De esta forma, una vez tenemos un conjunto de individuos, con el fin de aumentar la idoneidad de la población, se genera una

nueva población a partir de la anterior según la calidad de cada individuo: cuanto mejor sea la calidad de un individuo, mayor será la probabilidad de que sea incluido en la siguiente generación. A la hora de generar nuevas poblaciones se utilizan operaciones que potencien el proceso de evolución de manera que se generen mejores individuos. Por una parte el operador *cruce*, simula el comportamiento del ADN: se eligen aleatoriamente dos individuos y se dividen en dos partes. Finalmente se intercambian las partes de ambos individuos. Como ejemplo, consideremos los individuos:

Individuo 1 = 10**010** y Individuo 2 = 00**100**

Tras el cruce las partes resaltadas en negrita serán intercambiadas quedando:

Individuo 1 = 10**100** y Individuo 2 = 00**010**

El segundo operador utilizado se denomina *mutación*, y tal como ocurre en la naturaleza, se basa en cambios aleatorios que modifican las características de la población. En el caso de algoritmos genéticos los cambios se llevan a cabo eligiendo aleatoriamente un individuo y uno de sus genes. Una vez elegidos, se invierte el gen. Es decir si el valor es 1, se cambia a 0, y viceversa.

En cada generación se busca introducir una innovación deseable en la población con el fin de que preserven sus mejores características. Las probabilidades de mutación y cruce serán un aspecto muy importante a gestionar con el fin de permitir que el algoritmo converja a un valor óptimo definitivo y único. Unas probabilidades muy altas complican la convergencia, mientras que unas probabilidades muy bajas conducen a una exploración insuficiente del espacio de posibles soluciones para localizar un máximo global.

Como se introdujo en el capítulo 2, los satélites visibles son seguidos utilizando sus códigos de identificación única. En GPS se emplean como códigos de identificación los llamados Números Pseudo Aleatorios (PRN, Pseudo Random Numbers). Los requisitos de transmisión/recepción mediante técnicas de espectro ensanchado en la norma GPS hace imprescindible emplear un PRN como identificador [30]. En cada receptor se utiliza un

conjunto único fijo de PRNs según indica la especificación del sistema GPS [31]. Sin embargo en la presente Tesis Doctoral se propone utilizar un nuevo conjunto de PRN con el fin de llevar a cabo la identificación y seguimiento de satélites de una manera unívoca y con menor esfuerzo computacional. Los nuevos números de identificación se obtienen utilizando algoritmos genéticos (GAs) tanto de objetivo único como múltiple.

4.2. Nuevos códigos PRN

Para comenzar a explorar la metodología basada en algoritmos genéticos se contempla la creación de nuevos códigos ortogonales de detección de longitud 341. Tal como ocurría en anteriores apartados, para codificar a los individuos se utiliza el siguiente esquema: se define el individuo como un array de 341 variables booleanas. Cada uno de los individuos representa una nueva propuesta de código PRN para ser evaluada, y cada elemento binario, o gen del array individual modela un chip del nuevo código PRN. En esta codificación, el índice del array se utiliza para enlazar cada elemento booleano del individuo a cada chip del nuevo código PRN.

Se define la función alineamiento (α) entre una subsecuencia ($nPRN^{sat}$) y el código original (PRN_j^{sat}) como:

$$\alpha(PRN_j^{sat}, nPRN^{sat}) \begin{cases} 1 \Leftrightarrow PRN_j^{sat} \text{ alineado con } nPRN^{sat} \\ 0 \text{ resto de casos} \end{cases} \quad \forall j = A, B \text{ ó } C \quad (4.1)$$

Tal como se explicó en el capítulo anterior, para cumplir las propiedades que tienen los códigos PRN originales, es necesario que al utilizar el nuevo código sea posible obtener un valor alto de correlación cuando la señal recibida se encuentra alineada con el código PRN correspondiente al satélite que se está buscando. A este valor se le conoce como pico de detección (DP). A su vez, se debe obtener un valor muy bajo cuando la secuencia recibida no se encuentra alineada o no corresponde con el satélite que se está buscando. A este valor bajo se le denomina ruido de auto correlación (AN) o ruido de correlación cruzada (CN) respectivamente.

De esta forma se pueden definir 5 funciones coste con el fin de ayudar a medir la calidad del nuevo código PRN propuesto. Las 5 funciones de coste serán introducidas más adelante en forma de funciones objetivo, tomando como base las mostradas a continuación. Las primeras tres funciones se refieren a los picos de detección:

$$Corr_A^{sat} = Corr(PRN_A^{sat}, nPRN^{sat})$$

$$Corr_B^{sat} = Corr(PRN_B^{sat}, nPRN^{sat})$$

$$Corr_C^{sat} = Corr(PRN_C^{sat}, nPRN^{sat})$$

Donde PRN_A^{sat} , PRN_B^{sat} y PRN_C^{sat} se encuentran alineados con $nPRN^{sat}$.

La siguiente función se refiere al ruido de auto correlación:

$AN^{sat} = Max(AN_A^{sat}, AN_B^{sat}, AN_C^{sat})$, con:

$$AN_A^{sat} = Corr(PRN_A^{sat}, nPRN^{sat}) | \alpha(PRN_A^{sat}, nPRN^{sat}) = 1$$

$$AN_B^{sat} = Corr(PRN_B^{sat}, nPRN^{sat}) | \alpha(PRN_B^{sat}, nPRN^{sat}) = 1$$

$$AN_C^{sat} = Corr(PRN_C^{sat}, nPRN^{sat}) | \alpha(PRN_C^{sat}, nPRN^{sat}) = 1$$

Donde PRN_A^{sat} , PRN_B^{sat} y PRN_C^{sat} , no se encuentran alineados con $nPRN^{sat}$.

Finalmente, la correlación cruzada se evalúa de la siguiente forma:

$$CN^{sat} = Max(CN_A^{sat}, CN_B^{sat}, CN_C^{sat}), y$$

$$CN_A^{sat} = Corr(PRN_A^{sat}, nPRN^{sati}) | \alpha(PRN_A^{sat}, nPRN^{sati}) = 1$$

$$CN_B^{sat} = Corr(PRN_B^{sat}, nPRN^{sati}) | \alpha(PRN_B^{sat}, nPRN^{sati}) = 1$$

$$CN_C^{sat} = Corr(PRN_C^{sat}, nPRN^{sati}) | \alpha(PRN_C^{sat}, nPRN^{sati}) = 1$$

$$\forall sati \in [1,32] | sati \neq sat$$

Una vez definidos los valores que ayudarán a evaluar el éxito de las soluciones propuestas, se pasa a cuantificar el número de operaciones que conlleva cada función de evaluación, es decir, el esfuerzo computacional que

requieren las funciones de evaluación. Para calcular el esfuerzo computacional total, se utilizará como figura de mérito la *unidad de esfuerzo computacional*, que reflejará el esfuerzo computacional para llevar a cabo la correlación de 1.023 chips.

Como se explicó anteriormente, dentro de la trama continua de 1.023 chips que el receptor recibe de cada satélite, existirán tres posibles subsecuencias de 341 chips por cada código PRN. Y por definición, según lo visto anteriormente, existe un único pico de detección por cada subsecuencia para el caso en el que los chips recibidos y el dato con el que van a ser comparados estén alineados.

De esta forma, las funciones de evaluación $Corr_A^{sat}$, $Corr_B^{sat}$ y $Corr_C^{sat}$ requieren una correlación de 341 chips cada una, lo que significa un tercio de la unidad de esfuerzo computacional anteriormente descrita. Solamente uno de los 341 posibles alineamientos corresponde al pico de detección, y el resto de las otras 340 subsecuencias no alineadas producen ruido de auto correlación AN^{sat} . Con el fin de evaluar el ruido de auto correlación máximo, es necesario ejecutar 1.023 (341 x 3) correlaciones de 341 chips.

Cuando los datos recibidos no corresponden con el $nPRN^{sat}$ buscado, el resultado de los cálculos será ruido de correlación cruzada CN^{sat} . Con el fin de validar la metodología, la correlación cruzada ha de ser calculada teniendo en cuenta todos los posibles satélites. Esto quiere decir que para cada satélite, es necesario realizar 3 x 341 correlaciones de 341 chips, y por lo tanto, para los 31 posibles satélites restantes habrá que repetir los cálculos. De esta forma, el número total de correlaciones que habrá que realizar será $341 \times 3 \times 31 = 31.713$.

Con el fin de llegar a la mejor solución posible, a continuación se enumeran varias funciones objetivo tanto simple como múltiple, que combinan las 5 funciones de evaluación mencionadas previamente, presentando sus ventajas e inconvenientes.

Función objetivo simple: solución 1.

En este caso la función objetivo que guíe al algoritmo genético con el fin de obtener el $nPRN^{sat}$ para un satélite en concreto, será una función de evaluación que maximice la suma de las tres correlaciones entre el $nPRN^{sat}$ y las subsecuencias $nPRN_A^{sat}$, $nPRN_B^{sat}$ y $nPRN_C^{sat}$. Es decir:

$$\text{Maximizar: } Corr_A^{sat} + Corr_B^{sat} + Corr_C^{sat}$$

donde $Corr_A^{sat}$, $Corr_B^{sat}$ y $Corr_C^{sat}$ utilizan el mismo individuo propuesto por el motor de optimización del algoritmo genético: $nPRN^{sat}$.

El cálculo del objetivo se enfoca en maximizar los tres picos de auto correlación. Este procedimiento no requiere calcular los ruidos de auto correlación y correlación cruzada, reduciendo así el esfuerzo computacional a un valor mínimo. En este caso, la principal ventaja es al mismo tiempo responsable de la mayor debilidad, ya que al no tener los ruidos en cuenta, existe un riesgo potencial de encontrar un nuevo código PRN con un alto pico de auto correlación pero con un alto ruido.

Función multi-objetivo 1: solución 2a.

Transformar la función con objetivo simple presentada en el caso anterior en una función multi-objetivo es sencillo. Existe una solución evidente que consiste en definir un problema con tres funciones objetivo a maximizar, donde cada una de ellas corresponde a la auto correlación entre el propuesto y cada posible subsecuencia.

- Objetivo 1: maximizar la auto correlación con la subsecuencia A.

$$\text{Maximizar: } Corr_A^{sat}$$

- Objetivo 2: maximizar la auto correlación con la subsecuencia B.

$$\text{Maximizar: } Corr_B^{sat}$$

- Objetivo 3: maximizar la auto correlación con la subsecuencia C.

$$\text{Maximizar: } Corr_C^{sat}$$

Esta solución mantiene el mismo esfuerzo computacional que la función objetivo simple, pero al igual que ocurría con esta, no considera los ruidos de auto correlación y correlación cruzada.

Función multi-objetivo 2: solución 2b

Esta segunda función multi-objetivo surge como una evolución natural de la solución 2a. En este caso lo que se propone es un nuevo problema con cinco objetivos, donde los tres primeros son los mismos que los que se presentan en la solución 2a, mientras que los otros dos son los ruidos de auto correlación y correlación cruzada.

- Objetivo 1: maximizar la auto correlación con la subsecuencia A.

$$\text{Maximizar: } Corr_A^{sat}$$

- Objetivo 2: maximizar la auto correlación con la subsecuencia B.

$$\text{Maximizar: } Corr_B^{sat}$$

- Objetivo 3: maximizar la auto correlación con la subsecuencia C.

$$\text{Maximizar: } Corr_C^{sat}$$

- Objetivo 4: minimizar el ruido de auto correlación.

$$\text{Minimizar: } AN^{sat}$$

- Objetivo 5: minimizar el ruido de correlación cruzada.

$$\text{Minimizar: } CN^{sat}$$

En esta propuesta se incluye ya el ruido en el método de optimización, lo cual incrementa el esfuerzo computacional requerido de manera importante.

Función multi-objetivo 3: solución 3

A tenor de los resultados obtenidos en el estudio realizado en el artículo “*Low power GPS pseudo random numbers using genetic algorithms*” [32], se propone una solución multi-objetivo basada en tres objetivos. El primero de ellos es maximizar la suma de los tres picos de detección, y los otros dos son minimizar los ruidos de auto correlación y correlación cruzada. De esta forma:

- Objetivo 1: maximizar la suma de las subsecuencias alineadas.

$$\text{Maximizar: } Corr_A^{sat}$$

- Objetivo 2: minimizar el máximo ruido de auto correlación.

$$\text{Minimizar: } Corr_B^{sat}$$

- Objetivo 3: minimizar el ruido máximo de correlación cruzada.

$$\text{Minimizar: } Corr_C^{sat}$$

El esfuerzo computacional en este caso es lógicamente mayor que en el caso de la estrategia basada en un objetivo simple, pero algo inferior a las soluciones multi-objetivo propuestas anteriormente.

Función Multi-objetivo 4: solución 4

Finalmente, con el fin de minimizar el número de objetivos y así la computación asociada, se propone un problema multi-objetivo basado solamente en dos objetivos. En esta cuarta solución, uno de los objetivos es maximizar la suma de los picos de detección, y el otro consiste en minimizar el ruido de correlación cruzada:

- Objetivo 1: maximizar la suma de las subsecuencias alineadas.

$$\text{Maximizar: } Corr^{sat}$$

- Objetivo 2: minimizar el ruido máximo de correlación cruzada.

$$\text{Minimizar: } CN^{sat}$$

En este caso, el esfuerzo computacional es todavía alto, ya que el gran número de correlaciones que se realizan se trata ahora de correlaciones cruzadas en lugar de auto correlaciones, como ocurría antes. En cualquier caso, esta cuarta solución multi-objetivo determina si al maximizar el pico de detección se produce una reducción del ruido de correlación cruzada.

4.3. Resultados de los métodos presentados

Con el fin de apoyar el modelado GPS, se desarrolló un sencillo y rápido simulador basado en lógica combinatorial en lenguaje C. Se utilizó GENEYS

v1.0 como motor de optimización con objetivo simple, y NSGA-II v1.1.6 [33] y Omni-Optimizer v1.0 [34] para optimizaciones con objetivo simple y multi-objetivo. El simulador lógico basado en lógica combinatorial y los tres motores de optimización están integrados completamente en la herramienta Atenea [35].

4.3.1. Complejidad de los algoritmos y calidad de los códigos generados

Como se ha dicho anteriormente, el objetivo último de este estudio está enfocado a obtener nuevos códigos PRN de alta eficiencia, o lo que es equivalente, pseudocódigos que ofrezcan picos de detección altos (DP) y bajo ruido de correlación cruzada (CN) y auto correlación (AN). Para ello se han generado los nuevos códigos PRN utilizando distintos motores de optimización basados en AG. Los códigos PRN obtenidos se utilizan como un conjunto de números pseudo-aleatorios en cada receptor con el fin de identificar y seguir a cada satélite. Por lo tanto se generan una sola vez y no en tiempo real. Una vez generados, estos códigos pueden ser almacenados en un receptor software, de forma que no incrementa el esfuerzo computacional en el momento de la detección. Puesto que lo que se persigue es encontrar soluciones con Picos de Detección altos y bajo ruido, la métrica utilizada está orientada a primar la calidad de resultados en favor del tiempo de CPU o memoria requerida.

4.3.2. Solución basada en un solo objetivo

La Tabla 4.1 resume la configuración del AG en los experimentos llevados a cabo. En los casos en los que se utiliza GENEYs, se establece una población de 50 individuos. La estrategia utilizada para el reemplazo de la generación antigua después de la generación de la siguiente población (descendencia) es *Steady State*. Se conoce como *Steady State* el caso en el que se reemplazan sólo algunos individuos en cada iteración. El esquema de selección de AG elegido ha sido el *Proporcional*. Además se utilizó cruce multipunto y mutación estándar. El criterio de parada ha sido finalizar cuando se alcancen las 2.000 generaciones. Dado un satélite, la solución basada en optimización sencilla proporciona un único pseudo-código (*nPRN*). El tiempo medio de ejecución en todos los experimentos presentados se encuentra en torno a los 3.5 segundos por satélite.

GENESyS

Población	50
Reemplazo de la generación anterior	'Steady State'
Selección de esquema	Proporcional
Cruce multipunto	2
Mutación	Estándar
Criterio de parada	2.000 generaciones
Tiempo de ejecución ^a	3,5 segundos

NSGA-II y omni-optimizador

Población	40
Cruce	0.8 [0.6, 1.0]
Mutación	0.0029 [0.001, 0.004]
Criterio de parada	10.000
Tiempo de ejecución ^b	475.2 segundos (NSGA-II)

Nota: ^aTiempo medio para alcanzar el óptimo;

^bTiempo de CPU medio para obtener la Frontera de Pareto completa

Tabla 4.1. Parámetros de configuración del algoritmo genético y detalles de ejecución.

La Tabla 4.2 presenta los resultados obtenidos al utilizar el software GENESyS con el fin de minimizar la función de objetivo simple: $DP_A + DP_B + DP_C$, para generar el $nPRN$. La primera, cuarta, séptima y décima columna de la Tabla 4.2 presenta los números del satélite. Las columnas segunda, quinta, octava y onceava muestran el pico mínimo de detección de la señal cuando el $nPRN$ propuesto y el dato recibido están alineados. Es decir, el valor mínimo de DP_A , DP_B y DP_C . Por último, la tercera, sexta, novena y la última columna ofrecen el ruido máximo (N) para todas las posibles correlaciones cruzadas y auto correlaciones no alineadas, es decir, el valor máximo entre AN y CN . Se hace referencia al valor mínimo del pico de detección y al valor máximo de ruido para contemplar siempre el peor caso.

Como se muestra en la Tabla 4.2, el valor mínimo del pico de detección se localiza en el satélite 5 con sólo 133 unidades de correlación. El pico máximo de correlación se encuentra en el satélite 30 y 31 con 173 unidades de correlación. En términos de ruido, el ruido máximo se da en el satélite 8 con 81 unidades de correlación.

En la implementación de la electrónica que compone un detector GNSS, siguiendo a los circuitos de convolución, existe un comparador único (detector

de umbral) con el fin de determinar si existe un pico de detección o no. Si el receptor utiliza el $nPRN$ obtenido por medio del algoritmo GENEsYs, el detector de umbral debe ser definido en el margen [133-81]. Por lo que existirá una relación señal a ruido de 52 unidades de correlación.

Sat	DP	N									
1	161	77	9	153	69	17	153	63	25	169	51
2	157	69	10	157	65	18	165	63	26	169	67
3	161	57	11	157	63	19	161	59	27	153	71
4	157	81	12	145	71	20	157	67	28	161	65
5	133	73	13	161	69	21	173	63	29	165	67
6	157	61	14	169	57	22	153	63	30	173	59
7	157	63	15	169	59	23	157	61	31	173	63
8	145	81	16	149	53	24	153	61	32	161	73

DP: Pico de detección mínimo. (Mínimo de DP_A , DP_B y DP_C).
N: Ruido máximo. Valor Máximo entre AN y CN.

Tabla 4.2. Picos de detección mínimos (valor mínimo de DP_A , DP_B y DP_C) y ruido máximo (N) para el $nPRN$ óptimo cuando las secuencias recibidas se encuentran alineadas utilizando GENEsYs y una función objetivo simple.

4.3.3. Solución multi-objetivo

En este caso se propone modificar la función objetivo presentada anteriormente aplicando otros motores de optimización de AG. Además se introducirá una optimización multi-objetivo con el fin de obtener un $nPRN$ mejor. Para esta solución se utilizó el algoritmo NSGA-II y algoritmos de Omni-Optimización. La configuración para ambos algoritmos fue una población de 40 individuos. El criterio de parada fue alcanzar las 10.000 generaciones. Siguiendo las indicaciones que se hacen en la literatura relativa a ambos algoritmos, se establecieron la probabilidad de cruce en 0,8 y la probabilidad de mutación en $1/341 \approx 0,0029$. No obstante, con el fin de obtener mejores soluciones, estos parámetros fueron modificados. En el caso de la probabilidad de cruce, se utilizó el rango [0.6,1.0] con un paso de 0.05, y para la probabilidad de mutación se estableció el rango [0.001, 0.004] en pasos de 0.0005. Este ajuste se realiza en el código gracias a un bucle 'for' para ejecutar cada una de las variaciones, como muestra la Tabla 4.3. El tiempo de ejecución

medio en todos los experimentos ronda los 475,6 segundos por satélite cuando NSGA-2 es el motor de optimización y 455,6 segundos por satélite en el caso de que se utilice la Omni-Optimización.

En las Tablas 4.3 y 4.4 se muestran los picos de detección y los ruidos para todos los satélites de GPS que utilizan las citadas funciones objetivo para el NSGA-II y los algoritmos de Omni-Optimización respectivamente. En ambas tablas, el significado de las columnas es el mismo: en la primera columna se muestra el número de satélite. A continuación se muestran, en parejas, los picos de detección y los ruidos para cada solución propuesta. El primer par, situado en las columnas 2 y 3, proporciona el dato obtenido mediante las simulaciones utilizando la estrategia de objetivo simple número 1. Las columnas 4 y 5 presentan los picos de detección y ruido obtenidos al utilizar la solución propuesta 2a. Las columnas 6 y 7 muestran el dato del motor de optimización del AG cuando se utiliza la solución 2b. Las columnas 8 y 9 muestran los datos obtenidos al aplicar la solución 3. Finalmente, las últimas dos columnas presentan el dato de la solución 4.

Tal y como se muestra en la Tabla 4.3, cuando a la solución 1 se le aplica NSGA-II, el pico de detección mínimo se encuentra en 173 y el ruido máximo en 101 unidades de correlación. El ruido es mayor con respecto al nPRN obtenido mediante GENEsYs. Por tanto NSGA-II conlleva mejores soluciones que GENEsYs en términos de detección de señal, pero a la vez peores valores de ruido cuando se utiliza para optimizar la solución 1 de objetivo simple. Cuando se aplica la segunda solución, el ruido de las correlaciones aumenta dramáticamente hasta las 131 unidades de correlación (ver la cuarta columna para el satélite 7) y la señal mínima de detección es menor (147 unidades de correlación en la quinta columna para el satélite 13). Por lo tanto, el margen de ruido son solamente 16 unidades de correlación.

Sat ID	Sol ₁		Sol _{2a}		Sol _{2b}		Sol ₃		Sol ₄	
	DP	N	DP	N	DP	N	DP	N	DP	N
1	185	85	171	91	123	79	175	77	185	87
2	177	93	179	89	101	77	179	83	173	79
3	177	77	175	83	107	83	173	77	175	81
4	177	81	169	79	125	73	169	83	181	83
5	177	81	169	81	129	79	169	81	179	79
6	185	81	175	101	81	75	181	77	179	79
7	177	89	161	131	105	85	173	85	177	75
8	181	89	175	89	131	75	169	75	175	77
9	177	83	177	85	125	81	171	85	173	79
10	185	85	171	79	129	89	175	73	181	89
11	177	77	159	93	81	79	173	79	171	79
12	173	83	171	81	119	79	173	71	181	75
13	175	81	147	111	117	77	179	83	175	81
14	181	81	165	79	107	73	167	73	173	79
15	183	79	175	77	129	75	171	81	189	85
16	193	85	159	99	131	77	171	75	173	79
17	185	85	157	93	107	81	177	77	183	87
18	175	77	181	81	117	83	181	79	185	79
19	173	95	171	99	119	83	169	75	173	77
20	177	83	181	81	109	79	177	77	179	75
21	181	89	171	85	89	71	171	79	181	89
22	181	81	181	81	123	71	169	77	179	81
23	179	81	153	85	137	77	169	83	175	87
24	185	85	179	87	99	77	171	69	185	85
25	181	81	179	85	123	71	165	77	177	85
26	181	89	157	95	129	77	167	71	181	77
27	179	101	163	89	115	73	175	99	175	91
28	179	93	177	77	127	77	175	81	175	73
29	175	87	165	91	107	77	175	75	181	81
30	185	93	167	91	95	91	167	73	179	79
31	179	83	165	87	121	79	169	69	175	87
32	177	79	165	89	131	79	175	87	181	85

DP: Pico de detección mínimo. (Valor mínimo entre DP_A , DP_B y DP_C).

N: Ruido máximo. Máximo entre *AN* y *CN*.

*Sol*₁: Maximizar $DP_A + DP_B + DP_C$

*Sol*_{2a}: Maximizar DP_A , DP_B , DP_C

*Sol*_{2b}: Maximizar DP_A , DP_B , DP_C y Minimizar *AN*, *CN*

*Sol*₃: Maximizar $DP_A + DP_B + DP_C$ y Minimizar *AN*, *CN*

*Sol*₄: Maximizar $DP_A + DP_B + DP_C$ y Minimizar *CN*

Tabla 4.3. Picos de detección (*DP*) mínimos y ruido máximo (*N*) para el *nPRN* cuando las señales de entrada están alineadas utilizando NSGA-II para todas las soluciones de funciones objetivas.

Sat ID	Sol ₁		Sol _{2a}		Sol _{2b}		Sol ₃		Sol ₄	
	DP	N	DP	N	DP	N	DP	N	DP	N
1	173	77	55	95	111	89	153	77	167	75
2	185	87	63	85	119	71	153	73	167	71
3	181	81	67	85	111	81	165	77	159	73
4	169	79	91	79	117	85	155	75	165	73
5	175	83	63	91	83	89	157	77	165	71
6	183	89	59	95	107	75	159	79	171	79
7	169	81	55	95	107	77	159	75	167	75
8	177	77	55	99	79	79	167	81	163	73
9	169	81	69	83	99	87	149	73	169	81
10	183	81	69	99	79	77	165	81	175	77
11	179	85	75	79	103	75	157	69	169	75
12	167	79	63	79	95	75	151	77	163	71
13	169	85	93	81	83	91	153	75	163	71
14	153	85	53	79	73	77	147	79	159	71
15	169	81	51	91	109	85	149	75	169	73
16	171	81	59	81	123	91	155	77	171	75
17	181	89	63	91	105	81	161	81	161	75
18	181	89	91	89	87	73	151	75	169	79
19	185	81	67	91	121	87	161	81	167	81
20	173	77	61	99	75	77	159	75	169	77
21	165	83	73	73	69	89	163	75	167	83
22	173	77	59	93	119	81	163	77	167	75
23	171	81	63	93	71	77	161	79	163	75
24	181	81	59	93	75	83	163	79	177	71
25	183	81	97	85	113	83	157	73	173	73
26	175	73	71	95	119	79	165	79	157	71
27	177	79	69	85	109	83	151	73	171	77
28	169	73	71	81	79	81	165	79	159	71
29	177	77	53	111	107	77	151	71	163	73
30	171	77	59	89	119	91	157	75	163	69
31	169	79	61	95	119	93	167	79	161	71
32	181	83	77	73	123	77	157	75	159	71

DP: Pico de detección mínimo. (Valor mínimo entre DP_A , DP_B y DP_C).

N: Ruido máximo. Máximo entre *AN* y *CN*.

*Sol*₁: Maximizar $DP_A + DP_B + DP_C$

*Sol*_{2a}: Maximizar DP_A , DP_B , DP_C

*Sol*_{2b}: Maximizar DP_A , DP_B , DP_C y Minimizar *AN*, *CN*

*Sol*₃: Maximizar $DP_A + DP_B + DP_C$ y Minimizar *AN*, *CN*

*Sol*₄: Maximizar $DP_A + DP_B + DP_C$ y Minimizar *CN*

Tabla 4.4. Picos de detección mínimos (DP) y ruido máximo (N) utilizando Omni-Optimización.

A tenor de los resultados obtenidos la solución 2b, no resulta práctica utilizando un detector de umbral en la implementación debido a que existe un pico mínimo de detección de 81 (ver la columna 6 para el satélite 6) y el máximo ruido se localiza en la séptima columna para el satélite 30 con 91 unidades de correlación. Es decir, que hay casos en los que el ruido supera el pico mínimo de detección, por lo que la detección no se podría garantizar. Por su parte, la solución 3 ofrece mejores resultados que las anteriores en términos de detección y ruido. El mínimo pico de detección para la solución 3 es 165 que se da para el satélite 25 (ver columna 8) y un ruido de 99 unidades de correlación. Por tanto, la relación señal a ruido son 66 unidades de correlación. Por último, cuando la solución 4 se aplica al algoritmo NSGA-II, el pico de detección mínimo es 171 y el ruido máximo es 91. Estos resultados aumentan la relación señal a ruido hasta un mínimo de 80 unidades de correlación.

Cuando se aplica la solución 1 al Omni-Optimizador (ver columnas 2 y 3 de la tabla 4.4), el pico de detección y el ruido máximo son 153 en el satélite 14 y 89 en los satélites 6, 17 y 18. La relación señal a ruido es 64 unidades de correlación. Al utilizar la solución 2a, el pico de detección mínimo cae hasta 51 (ver el satélite 15 en la columna 4) mientras que el ruido máximo crece hasta 111. Por lo tanto, el resultado de esta solución no es aplicable para una detección con umbral. De la misma forma, la solución 2b, proporciona un pico mínimo de detección de 71 y un ruido máximo de 93 (ver los satélites 23 y 31 en las columnas 6 y 7, respectivamente). Todo esto indica que utilizando estos métodos es imposible detectar satélites GPS utilizando la detección por umbral. El pico mínimo de detección al aplicar la solución 3 es 147 (satélite 14 en la columna 8) y el ruido máximo es 81 (ver satélites 8, 19, 17 y 19 en la columna 9). En este caso la relación señal a ruido es 66 unidades de correlación. Finalmente, cuando se utiliza la solución 4, el pico mínimo de detección es 157 para el satélite 26 (ver la columna 10) y el ruido máximo se localiza en el satélite 21 con un valor de 83 unidades de correlación (ver la última columna).

Objetivo	NSGA-II			Omni-Optimizer			GENEsYs		
	DP	N	SN	DP	N	SN	DP	N	SN
Sol_1	173	101	62	153	89	64	133	81	52
Sol_{2a}	147	131	16	51	111	NA		MU	
Sol_{2b}	81	91	NA	71	91	NA		MU	
Sol_3	165	99	66	147	81	66		MU	
Sol_4	171	91	80	159	83	76		MU	

DP: Pico de detección mínimo. (Mínimo entre DP_A , DP_B y DP_C).

N: Ruido máximo. Máximo entre AN y CN.

SN: Relación Señal a Ruido.

NA: Solución no aplicable.

MU: Optimización Multiobjetivo no disponible.

Tabla 4.5. Comparativa de los picos de detección (DP) y ruido máximo (N) obtenidos utilizando NSGA-II, Omni-Optimizador y GENEsYs.

La Tabla 4.5 resume los picos mínimos de detección y los ruidos máximos obtenidos a partir de los algoritmos de optimización al utilizar las funciones objetivo propuestas. Cada fila de la tabla corresponde a una de estas funciones. La primera columna identifica la función objetivo a considerar. Las tres columnas siguientes dan el pico de detección mínimo, el ruido máximo y la relación señal a ruido para NSGA-II. Las columnas de la cinco a la siete muestran estos mismos tres parámetros, pero estos son generados por el algoritmo Omni-Optimizador. Finalmente, las últimas tres columnas muestran los datos obtenidos utilizando GENEsYs. Puesto que GENEsYs es un motor de optimización simple, no existen resultados para las soluciones multi-objetivo.

En general, los resultados presentados en la Tabla 4.5, indican que es mejor utilizar un algoritmo multi-objetivo para maximizar el pico de detección mínimo con mínimo ruido, es decir obtener la máxima relación señal a ruido, en comparación con la solución de objetivo simple.

Las mejores soluciones se encuentran cuando el motor de optimización multi-objetivo maneja un número bajo de funciones objetivo independientes (no más de 3). Los mayores picos de detección se obtienen cuando el NSGA-II utiliza la solución número 4. En este sentido, hay que hacer notar que el ruido de auto-correlación es siempre menor que el ruido de correlación cruzada para todos los casos estudiados. Esto es así debido a que el ruido de auto-correlación es minimizado. Hay que recordar que la correlación es una función

que mide el grado de coincidencia entre las subsecuencias y los $nPRNs$ propuestos, y que los motores de optimización intentan maximizar la auto-correlación de las secuencias alineadas.

Por otra parte, se da un comportamiento inesperado en las soluciones 2a y 2b, donde el problema no se resuelve para algunos casos. Para el problema de encontrar el mejor $nPRN$ utilizando las soluciones 2a, el espacio de búsqueda se puede modelar como un cubo con una arista de longitud 341, donde cada arista del cubo es paralela a un sistema de coordenadas tridimensionales. Cada eje representa una función objetivo de la optimización multi-objetivo. En otras palabras, los ejes representan las tres auto-correlaciones DP_A , DP_B y DP_C . La Figura 4.3 representa este espacio de búsqueda.

En teoría, en este espacio de búsqueda, sólo existe un máximo que es el vector [341, 341, 341]. Debido a la naturaleza pseudo-aleatoria del código PRN original, es razonable afirmar que este máximo no es alcanzable. En otras palabras, si se obtiene un $nPRN$ de 341 chips de longitud con un pico de detección de 341 para las tres posibles subsecuencias a recibir, esto implica que las tres subsecuencias serían la misma. Pero como esto no puede ocurrir porque el código PRN original es una secuencia pseudo aleatoria, es imposible obtener el vector [341, 341, 341]. Por tanto, el problema global de buscar el mejor $nPRN$ se torna en realizar una búsqueda de la frontera de Pareto del mejor $nPRN$.

Resulta de interés el $nPRN$ cercano a la diagonal del cubo que va desde el origen al máximo teórico (representado en la figura 4.3 con línea de puntos) y que se encuentra cerca del máximo teórico. Estos puntos representan el $nPRN$ donde los valores del pico de detección son los mejores para las tres posibles subsecuencias.

En la Figura 4.3, se localizan los puntos óptimos que proporciona GENEYS cerca del centro del espacio de búsqueda (marcado con el símbolo '*'). La progresión de la búsqueda utilizando la solución 2a, NSGA-II y el Omni-Optimizador, está etiquetada con los símbolos '+' y 'x' respectivamente. Según lo dicho anteriormente, se asume que los motores de optimización deben

mantener las soluciones en cada generación, cerca de la diagonal del cubo y tan cerca como sea posible del vector solución [341, 341, 341]. Al principio ambos motores de optimización se mueven cerca de la diagonal del cubo. Pero según va creciendo el número de generaciones, ambos motores de optimización divergen durante varias generaciones. Finalmente ambos algoritmos multi-objetivo se aproximan a la misma área. Por desgracia, los algoritmos multi-objetivo caen en un área donde existe un mínimo local y no son capaces de abandonarla para encontrar mejores soluciones. Como se puede observar en la Figura 4.3, los resultados de las generaciones aumentan sustancialmente el objetivo 1 manteniendo alrededor del mismo valor los objetivos 2 y 3, cuando se utiliza el Omni-Optimizador. Por otro lado, el algoritmo NSGA-II empieza a incrementar el valor del objetivo global, pero llega a un punto donde solamente optimiza el objetivo 1 y reduce el valor de los objetivos 2 y 3.

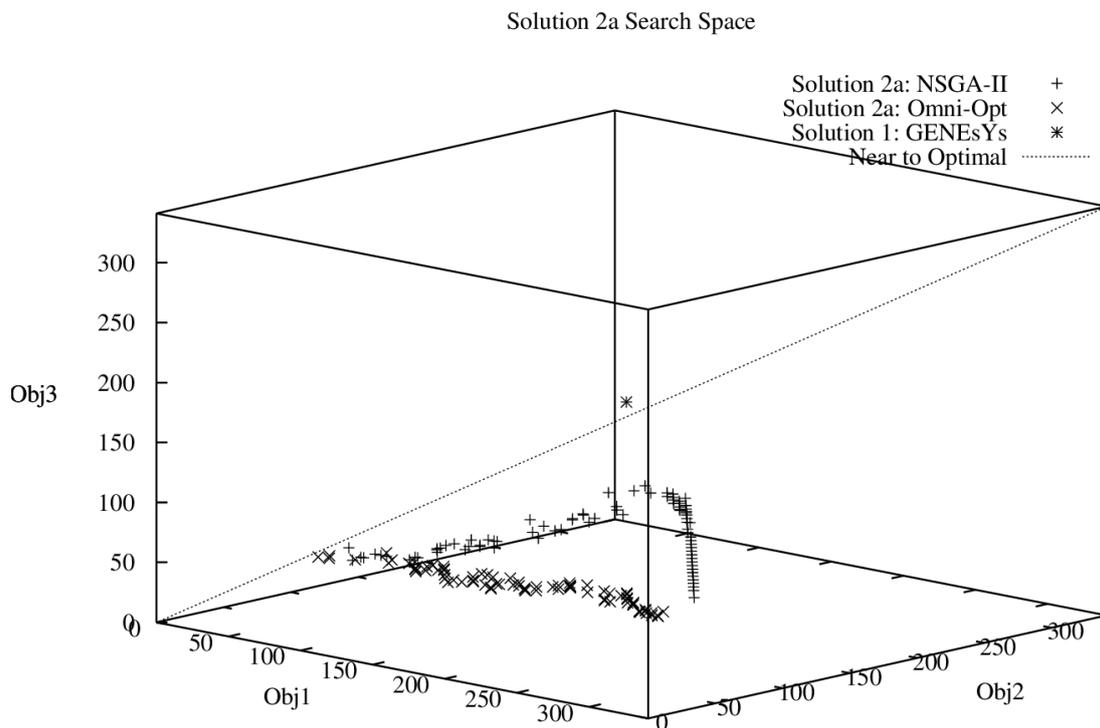


Figura 4.3. Espacio de búsqueda completo (cubo). *nPRN* óptimo para la solución 2a utilizando NSGA-II y el Omni-Optimizador y la solución 1 de GENESYs.

Los resultados obtenidos a partir de las funciones de objetivo simple / múltiple evaluadas y de los motores de optimización, indican que, en ocasiones, es posible obtener $nPRN$ óptimos en términos de picos de detección, pero no óptimos en términos de relación señal a ruido. Además, no existe ningún algoritmo sencillo que, combinado con una función de objetivo simple / múltiple, proporcione el mejor $nPRN$. En este sentido, la Tabla 4.6 presenta los mejores resultados en términos de picos de detección óptimos y mejor relación señal a ruido. La primera columna muestra el número de satélite. A continuación, de la segunda a la sexta columna, se presentan los resultados para los picos de detección más altos. De igual forma, las siguientes columnas muestran los resultados para los mejores valores de relación señal a ruido.

La figura 4.4 muestra la curva de la Frontera de Pareto típica obtenida en todos los experimentos utilizando el algoritmo NSGA-II y la función multi-objetivo 1, es decir, Objetivo 1: Maximizar ($Corr_A^{sat}$), Objetivo 2: Maximizar ($Corr_B^{sat}$) y Objetivo 3: Maximizar ($Corr_C^{sat}$) para encontrar el nuevo código PRN del satélite 1. Las figuras 4.4 (a), (b) y (c) muestran la dependencia entre objetivos utilizando gráficos en dos dimensiones. En las tres figuras, se incluye la diagonal del cubo mencionado anteriormente, considerada lo más cercano al punto óptimo. Los puntos que más se acercan a esta diagonal pertenecen a la frontera de Pareto. Comparando las figuras 3(a) y 3(b), se puede observar que el Objetivo 1 vs Objetivo 3 muestra el mismo comportamiento que el Objetivo 1 vs Objetivo 2 en cuanto a la frontera de Pareto.

En cualquier caso, la Figura 4.4(c) muestra una diferencia entre las comparaciones previas y el gráfico que muestra el Objetivo 2 vs el Objetivo 3. Los motores de optimización del algoritmo genético tratan de maximizar los tres Picos de Detección (DP_A , DP_B y DP_C). Cuando la frontera de Pareto aparece como se muestra en la figura 4.4 (c), quiere decir que existen solamente un conjunto reducido de soluciones pertenecientes a la frontera de Pareto. En términos de códigos $nPRN$, este comportamiento indica que dos de los Picos de Detección están demasiado cerca y el tercero no se encuentra tan cerca como los dos primeros.

Sat	Optimal DP			Algorithm	Solution	Optimal SN			Algorithm	Solution
	DP	N	SN			DP	N	SN		
1	185	85	100	NSGA-II	1	*1				
2	185	87	98	OMNI-opt	1	*1				
3	181	87	94	OMNI-opt	1	161	57	104	GENEsYs	1
4	181	83	98	NSGA-II	4	*1				
5	179	79	100	NSGA-II	4	*1				
6	185	81	104	NSGA-II	1	*1				
7	177	75	102	NSGA-II	4	*1				
8	181	89	92	NSGA-II	1	177	77	100	OMNI-opt	1
9	177	83	94	NSGA-II	1	*1				
10	185	85	100	NSGA-II	1	183	81	102	OMNI-opt	1
11	179	85	94	NSGA-II	1	*1				
12	181	75	106	NSGA-II	4	*1				
13	179	83	96	NSGA-II	3	*1				
14	181	81	100	NSGA-II	1	*1				
15	189	85	104	NSGA-II	4	169	59	110	GENEsYs	1
16	193	85	108	NSGA-II	1	*1				
17	185	85	100	NSGA-II	1	*1				
18	185	79	106	NSGA-II	4	*1				
19	185	81	104	OMNI-opt	1	*1				
20	179	75	104	NSGA-II	3	*1				
21	181	89	92	NSGA-II	1	173	63	110	GENEsYs	1
				NSGA-II	4					
22	181	81	100	NSGA-II	1	*1				
23	179	81	98	NSGA-II	1	*1				
24	185	85	100	NSGA-II	1	177	71	106	OMNI-opt	4
25	183	81	102	OMNI-opt	1	169	51	118	GENEsYs	1
26	181	77	104	NSGA-II	4	*1				
27	179	101	78	NSGA-II	1	171	77	94	OMNI-opt	4
28	179	93	86	NSGA-II	1	*1				
29	181	81	100	NSGA-II	4	*1				
30	185	93	92	NSGA-II	1	173	59	114	GENEsYs	1
31	179	83	96	NSGA-II	1	173	63	110	GENEsYs	1
32	181	83	98	OMNI-opt	1	*1				

DP : Pico de detección mínimo.

N : Ruido máximo.

SN : Relación Señal a Ruido.

*1: El DP óptimo es también el SN óptimo.

*2: OMNI-optimizer obtiene 177, 79 y 98 (DP , N y SN).

Tabla 4.6. Selección del mejor $nPRN$ a partir de los tres motores de optimización y soluciones.

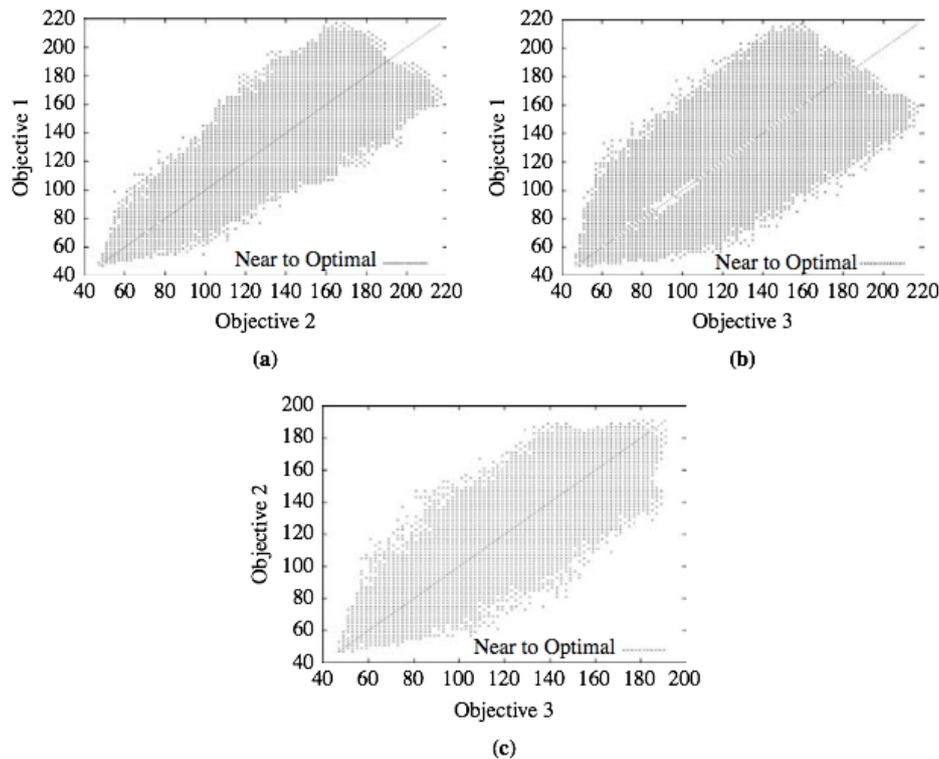


Figura 4.4. Representación de la Frontera de Pareto para la búsqueda del nuevo código PRN para el satélite 1 utilizando NSGA-II y la función multiobjetivo 1.

Los resultados obtenidos a partir de las funciones de objetivo simple / múltiple evaluadas y de los motores de optimización indican que en ocasiones es posible obtener $nPRN$ óptimos en términos de picos de detección, pero no óptimos en términos de relación señal a ruido. Además, no existe ningún algoritmo sencillo que combinado con una función de objetivo simple / múltiple, proporcione el mejor $nPRN$. En este sentido, la tabla 4.6 presenta los mejores resultados en términos de picos de detección óptimos y mejor relación señal a ruido. La primera columna muestra el número de satélite. A continuación de la segunda a la sexta columna se presentan los resultados para los picos de detección más altos. De igual forma, las siguientes columnas muestran los resultados para los mejores valores de relación señal a ruido.

El pico de detección mínimo en esta tabla es 177 unidades de correlación (ver satélites 7 y 9). El ruido máximo se produce en el satélite 27 con un valor de 101 unidades de correlación. Por tanto, la relación señal a ruido es de sólo 76 unidades de correlación al elegir cada $nPRN$ con el criterio de selección de máximo pico de detección. El único responsable para este valor tan alto de

ruido es el satélite 27, por lo que para reducir el valor máximo de ruido se puede elegir otro *nPRN* con menor pico de detección, siempre mayor o igual que 177, y con un ruido máximo menor. En este caso, el motor de optimización Omni-Optimizador basado en la solución 1, proporciona un pico de detección de 177 y un ruido máximo de 98. Ciertamente, de esta forma se reduce el ruido máximo, aunque el satélite 27 sigue siendo el responsable del mayor nivel de ruido.

En resumen, existe un compromiso entre la selección del pico de detección y la relación señal a ruido que permite en varios casos aumentar o disminuir uno de estos dos parámetros con el fin de obtener un rendimiento global mejor dentro de todos los *nPRN* elegidos. Este escenario define un nuevo problema que es: cómo elegir el mejor conjunto de *nPRN* a partir de las curvas de Pareto de cada *nPRN* independiente. Este nuevo escenario se encuentra fuera del objetivo de esta Tesis Doctoral.

Los valores óptimos para el *nPRN* que se presentan en la tabla 4.5 confirman que las soluciones 2a y 2b no producen resultados óptimos al comparar los datos obtenidos en términos de relación señal a ruido y número de objetivos requeridos. Se observa también que el algoritmo NSGA-II genera el 87,5% de los *nPRN* elegidos cuando la selección se realiza en términos de picos de detección, y un 60,71% de estos *nPRNs* fueron obtenidos con una función objetivo simple (Solución 1). Los *nPRN* obtenidos mediante el Omni-Optimizador, son mejores que los obtenidos con el algoritmo NSGA-II en sólo un 13,5% de los casos. En estos casos, el conjunto de *nPRNs* obtenidos mediante el Omni-Optimizador no está lejos de la solución obtenida mediante NSGA-II.

El 68,75% de los mejores *nPRN* en términos de picos de detección son también los mejores en términos de relación señal a ruido. Por último es necesario precisar que GENEYS ofrece las mejores soluciones para algunos casos (18,75%) en los que se utiliza el criterio de relación señal a ruido. Esto quiere decir que otros motores de optimización basados en AG caen en mínimos locales en estos casos.

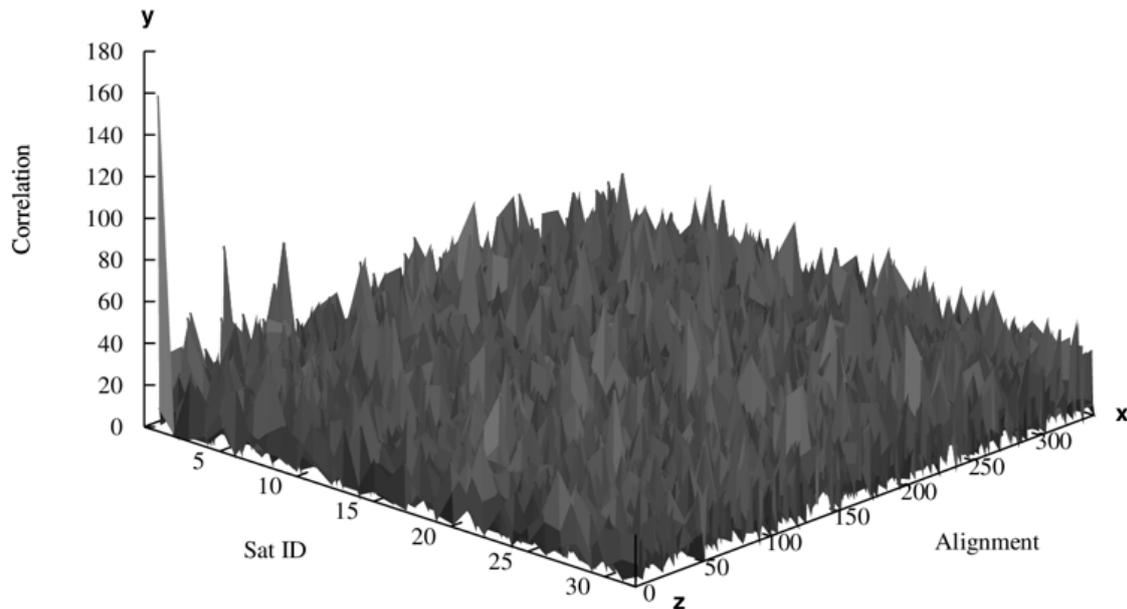


Figura 4.5. Correlación entre los datos que recibe el GPS y el $nPRN$ propuesto para el satélite 1.

La figura 4.5 muestra el proceso de correlación típico ejecutado en un receptor GPS cuando el satélite buscado se encuentra presente en la señal recibida. Esta curva muestra todas las posibles auto-correlaciones y correlaciones cruzadas del $nPRN$ obtenidas con GENEsYs para el satélite 1 con respecto al resto de satélites. En esta figura, a lo largo de los ejes X e Y, se encuentran representados el satélite correlado y la alineación entre la secuencia $nPRN$ y el dato recibido. En estos casos el pico de detección se localiza en la alineación 0 para el satélite 1. El pico de detección es 161 unidades de correlación. El ruido se representa mediante distintos picos y varios valores con un valor máximo de 77 unidades de correlación.

Una vez demostrada la existencia del nuevo PRN reducido, es posible evaluar el esfuerzo computacional total requerido utilizando este método. Tomando la ecuación (2) y sustituyendo $size (ID^{sat})$ y $size (D^x)$ por 341 se obtiene:

$$size (ID^{sat}) \times size (D^x) \times bins \times satellites = 152.560.672$$

Esta cifra representa el 11.11% del esfuerzo computacional necesario para determinar los satélites GPS visibles con respecto a la metodología utilizada hasta ahora. Esto supone un importante ahorro de computación, que

individuo se constituye mediante una secuencia de valores booleanos (ver Figura 4.7). El estado de cada valor booleano puede ser solamente un cero lógico, o un uno lógico. Los estados desconocido o error están prohibidos en este motor. La longitud de la secuencia determina la longitud del nuevo código de detección. Finalmente, cada elemento de la secuencia individual, se corresponde con un elemento del nuevo código de detección; esto hace que el índice sea el mismo en ambas secuencias.

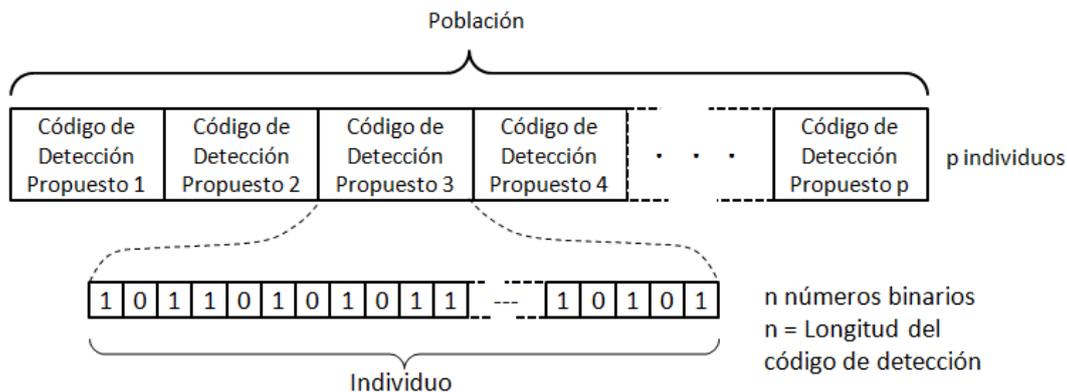


Figura 4.7. Codificación del Problema

Como es natural, la función coste del algoritmo genético será la ecuación 2.1. Tal como se mencionó anteriormente en la introducción de la metodología del algoritmo genético, esta función coste es una secuencia de valores comparativos. No obstante, estos valores no son interesantes. Los valores que resultan importantes para el estudio y los posteriores resultados son, como ocurre en apartados anteriores, el Pico de Detección (DP) y el Ruido (N). Es conocido además que la tasa de descarte determina el número total de Picos de Detección en la secuencia de comparación. Por ejemplo, si la longitud del nuevo código de detección se fija en 341, y por tanto la tasa de descarte se fija en 3, entonces existirán 3 Picos de Detección distintos correspondientes a cada código PRN original. Por lo que la secuencia de comparación contendrá tantos Picos de Detección como indique la tasa de descarte. El resto de valores existentes en el array de comparación, son valores de Ruido.

El interés de la investigación radica, como en anteriores ocasiones, en obtener nuevos códigos de detección GPS con un valor de Ruido bajo y unos Picos de Detección altos. Para obtener estos valores, tras evaluar cada individuo con la ecuación 2.1 la función de evaluación extrae el Pico de

Detección mínimo y el máximo valor de Ruido a partir de la secuencia de comparación generada.

4.5. Experimentos

Para llevar a cabo los experimentos se asume que el espacio de búsqueda de este problema crece junto con la longitud de los nuevos códigos de detección y la tasa de descarte utilizada. Si se provee de todos los grados de libertad a todas las variables del problema, el esfuerzo computacional requerido se incrementaría en gran medida. La magnitud de este esfuerzo computacional produce tiempos de ejecución de CPU que no son prácticos. En este sentido, con el fin de obtener las soluciones en un tiempo práctico de CPU, se incluirá la siguiente regla:

$$| \text{Longitud Propuesta} \times \text{Tasa de descarte} - 1.023 | < \xi \quad (4.1)$$

donde ξ es la desviación máxima permitida entre el PRN original y el nuevo PRN propuesto.

La tabla 4.7 presenta la configuración del motor de optimización NSGA-II. Como se puede ver, se definen solamente dos funciones objetivo a optimizar. La primera consiste en maximizar DP (minimizar -DP) y la segunda es minimizar el Ruido (N). El motor de optimización NSGA-II está controlado por un programa escrito en C que propone longitudes y tamaños de forma aleatoria, utilizando la regla introducida en la ecuación 4.1.

<i>Parámetro</i>	<i>Valor</i>
Número de Objetivo	2
Obj 1	-DP
Obj 2	N
Población	40
Generaciones	10k/100k
Crossover	0.8
Mutación	1/Longitud PRN
Semilla	Aleatorio / Distribución Uniforme

Tabla 4.7. Parámetros para el algoritmo NSGA-II utilizado.

El código C desarrollado para comprobar la aplicación, propone distintas longitudes y tasas de descarte y el motor de optimización NSGA-II obtiene nuevos códigos de detección. Esto quiere decir que la aplicación comienza proponiendo una nueva longitud y una nueva tasa de descarte con el fin de obtener un nuevo código de detección. Entonces el motor de optimización NSGA-II toma el control y produce una curva de la Frontera de Pareto para esos parámetros. Una vez que el proceso de optimización ha finalizado, la aplicación desarrollada extrae la mejor solución de entre las disponibles. Este procedimiento se lleva a cabo una y otra vez con varias longitudes y tasas de descarte. Después de extraer la mejor solución, la aplicación evalúa la convergencia de los parámetros propuestos. Si la aplicación determina que la solución puede ser mejorada, el NSGA-II se ejecuta de nuevo con más generaciones (entre 10.000 y 100.000).

La tabla 4.8 muestra algunos resultados obtenidos para el satélite de GPS 1. Para este caso ξ se fija en un máximo de 400 chips. Las dos primeras columnas de la tabla contienen la longitud del código propuesta y la tasa de descarte. El número de generaciones se muestra en la tercera columna (en términos de mil generaciones). La cuarta columna muestra el valor de la regla definida en la ecuación 4.1. Las siguientes dos columnas muestran el mejor pico de detección mínimo y el ruido máximo. La séptima columna indica el salto de detección, que es el margen entre el pico de detección mínimo y el ruido máximo. La siguiente columna mide el esfuerzo computacional requerido en las operaciones de multiplicación y comparación necesarias para ejecutar la ecuación 2.1. La novena columna muestra el esfuerzo computacional previo. La última columna muestra finalmente la comparativa entre el esfuerzo computacional necesario al utilizarla nueva propuesta y la metodología tradicional respectivamente. Además, las dos últimas filas de esta tabla muestran la mejor solución obtenida utilizando la metodología tradicional y la mejor solución obtenida anteriormente utilizando los algoritmos genéticos.

4.6. Conclusiones

En este capítulo se explora la generación de códigos a través de distintas metodologías utilizando algoritmos genéticos. Se llevan a cabo varios

experimentos utilizando distintos motores de optimización y funciones objetivo, obteniéndose en general mejores resultados que con el método determinista.

Tal como cabía esperar, los resultados que se muestran en la tabla 4.8 son mejores en términos de ruido máximo y/o pico de detección mínimo, conforme aumenta el número de generaciones totales. La última columna presenta una comparación entre la metodología tradicional y los nuevos códigos de detección reducidos. La comparación se lleva a cabo en términos de esfuerzo computacional requerido y salto de detección con el fin de determinar si un satélite es o no visible. Los valores negativos de esta columna indican una mejor solución que para el caso en que se utilizase la metodología tradicional.

Longitud del código	Tasa de descarte	Generaciones (x1k)	Valor de la regla	Min(DP)	Max(N)	D _G	CPU _O	CPU _E	CPU _{diff}
512	2	10	1.024	276	72	204	262,144	1.285,0	188
212	3	10	636	90	46	44	44,944	1.021,5	-76
288	3	10	864	116	54	62	82,944	1.337,8	241
304	3	10	912	126	56	70	92,416	1.320,2	223
320	3	10	960	134	56	78	102,400	1.312,8	216
341	3	10	1.023	141	59	82	116,281	1.418,1	321
256	4	10	1.024	90	52	38	65,536	1.724,6	628
512	2	100	1.024	276	70	206	262,144	1.272,5	176
288	3	100	864	136	54	82	82,944	1.011,5	-85
304	3	100	912	148	56	92	92,416	1.004,5	-92
320	3	100	960	156	56	100	102,400	1.024,0	-73
341	3	100	1.023	177	57	120	116,281	1.424,7	-128
256	4	100	1.024	98	52	46	65,536	1.424,7	328
341	3	40	1.023	177	89	88	116,281	1.321,4	224 ^{*1}
1023	1	-	1	1.023	69	954	1.046,529	1.097,0	0 ^{*2}

Min (DP): Pico de Detección mínimo, max(N): máximo ruido, D_G: Gap de Detección, CPU_O: Operaciones de CPU, CPU_E: Esfuerzo Computacional, CPU_{diff}: Diferencia de esfuerzo computacional. ^{*1} Datos del estudio anterior. ^{*2} Metodología Tradicional.

Tabla 4.8. Experimentos con NSGA-II para el Satélite 1 y $\xi < 400$.

A partir de la tabla 4.8, se observa que por ejemplo, la combinación 320 x 3 (longitud x tasa de descarte) con 100.000 generaciones tiene valores

similares en la última columna, -73 y -76, respectivamente. En la primera columna el Salto de Detección es el doble que en la segunda, pero la segunda propuesta requiere la mitad de operaciones para evaluar la ecuación 2.1. Por lo tanto, esta comparación demuestra que existen varias combinaciones de longitudes de código y tasas de descarte que tienen similares tasas de operaciones de CPU y Saltos de Detección.

Los resultados obtenidos con la aplicación de este método, utilizando los parámetros que se muestran en la tabla 4.7, son siempre mejores que los presentados en los casos anteriores en términos de máximo Ruido. Para los Picos de Detección, este último método obtiene siempre al menos los mismos valores o mejores.

Sin embargo como se puede observar, en los apartados anteriores se ha dejado de lado el estudio de la atenuación, y se ha centrado más en la posibilidad de detección de los satélites con los nuevos códigos, que en el punto de vista de la implementación hardware del receptor.

Con el fin de cubrir estos interesantes aspectos, en el siguiente capítulo se llevará a cabo un estudio para reducir el esfuerzo computacional de un receptor Galileo utilizando un método similar al que se ha seguido para los receptores GPS.

Capítulo 5

Metodología utilizando Algoritmos Genéticos aplicados a GALILEO

5.1. Galileo	74
5.2. Introducción de la nueva metodología	75
5.3. Experimentos	77
5.4. Conclusiones	85

Acrónimos

AltBOC	Alternative BOC modulation
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
NSGA	Nondominated Sorting Genetic Algorithm
RF	Radio Frequency

5.1. Galileo

La característica más importante desde el punto de vista del uso civil es que, a diferencia de lo que ocurre con el sistema GPS, Galileo proporciona señales para determinar la posición con mucha precisión [36, 37]. GPS ofrece un servicio de posicionamiento estándar utilizando los códigos L1 [38], mientras que el servicio de posicionamiento preciso utilizado a través de L5, se

encuentra actualmente en desarrollo y no son accesibles para aplicaciones civiles [39, 40]. Los códigos utilizados en el servicio de precisión de Galileo E5a y E5b en cambio, están disponibles para aplicaciones civiles [41, 42, 43].

5.2. Introducción de la nueva metodología

En los apartados anteriores relativos a GPS, se proponía una metodología de submuestreo de la señal recibida del satélite. Desde el punto de vista de la implementación hardware, seguir esta técnica de submuestreo requiere fijar la frecuencia de trabajo del front end analógico de RF que recibe las señales del satélite, y a su vez modificar la frecuencia de trabajo de las unidades hardware de correlación. Sin embargo, los receptores GNSS actuales utilizan una misma frecuencia de trabajo prefijada para ambas unidades. Una posible solución para solventar este problema es poner en marcha el proceso de correlación solamente durante un intervalo específico de tiempo. De esta forma se evita la limitación para utilizar estos nuevos métodos en un receptor GNSS actual con frecuencia de trabajo fija. En esta nueva metodología que se ilustra en la figura 5.1, primero el código PRN estándar se divide en varias secciones de igual longitud. Cada sección es generada con los bits contiguos del código original. A continuación se buscan nuevos códigos o patrones de detección de la misma longitud que las secciones anteriores utilizando el algoritmo genético, tal como se estudió para los casos de GPS. De forma similar que en apartados anteriores, se producirán nuevos códigos PRN E5a/b para la constelación de satélites Galileo de longitud reducida. Una vez obtenidos los nuevos códigos de detección, éstos pueden ser utilizados en un receptor comercial GNSS. Obviamente el software del receptor debe de estar actualizado con la nueva secuencia de arranque de las unidades de correlación. De esta forma no se requerirá modificar la frecuencia de trabajo ni ningún otro parámetro del receptor GNSS como ocurría con los casos vistos anteriormente relativos a GPS.

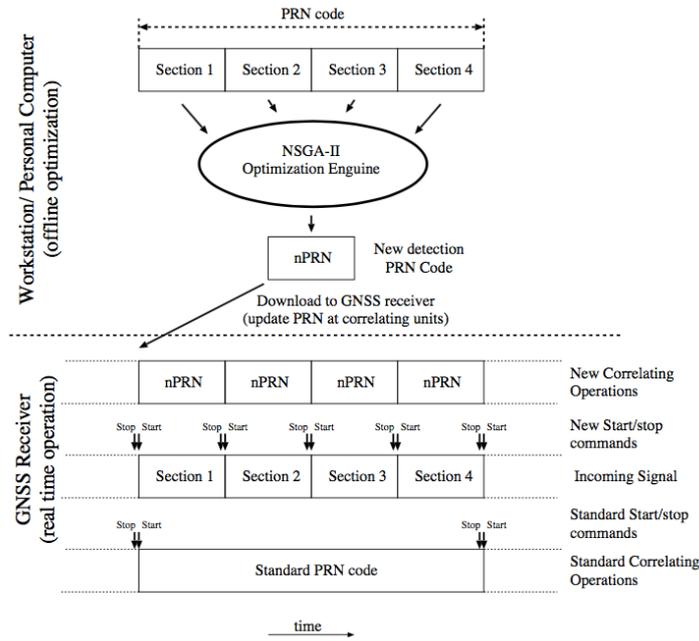


Figura 5.1. Metodología propuesta para la obtención de los nuevos códigos Galileo E5a/b.

Puesto que previamente se ha demostrado para los casos relativos a GPS que el NSGA-II es el motor de optimización que produce los mejores resultados, éste ha sido el elegido también para el caso del sistema Galileo. De la misma manera que ocurría en los casos anteriores, en este nuevo caso de estudio cada individuo representa un nuevo código PRN en potencia. Por lo tanto cada individuo es una matriz unidimensional o sucesión de valores binarios. La longitud de cada una de estas matrices es la misma que la del nuevo código de detección PRN propuesto. La longitud del nuevo código de detección PRN es una de las variables a optimizar.

De la misma forma que para los casos anteriores, una vez que un potencial nuevo código PRN es sugerido, es necesario comprobar sus propiedades de auto-correlación y correlación cruzada. Para ello la ecuación 2.1 se reescribe de la siguiente forma:

$$Corr(PRN^{sat}, nPRN) = [c_1, c_2, \dots, c_L];$$

$$c_i = \sum_{j=1}^L PRN_{\text{mod}_L(i+j)}^{sat} * nPRN_i \quad (5.1)$$

Donde la modificación ha sido reemplazar la señal recibida del satélite en la ecuación 2.1 por el nuevo código PRN de detección. Gracias a la ecuación 5.1, es posible obtener los picos de detección (DP) y el ruido (N) para cada código PRN Galileo E5a/b.

La función coste multiobjetivo del motor de optimización NSGA-II se define como la maximización de DP, y la minimización de N. Esto es equivalente a buscar la minimización de la tupla $[-DP, N]$. Es decir se buscarán nuevos códigos PRN E5a/b de Galileo con bajo ruido (N) y altos picos de detección (DP), tal como ocurría en casos anteriores. Además, con el fin de reducir todo lo posible el esfuerzo computacional requerido a la hora de evaluar la ecuación 5.1, la longitud de los nuevos códigos de detección deberá ser tan corta como sea posible.

5.3. Experimentos

Los códigos E5a/b de Galileo son diez veces más largos que los códigos L1 de GPS. Esto significa que es obligatorio ejecutar 10.230×10.230 operaciones de multiplicación-suma para evaluar la correlación representada en la ecuación 5.1. No obstante esta fórmula solamente compara el nuevo código PRN propuesto con otro de los ya establecidos en el conjunto de códigos que Galileo tiene definidos para la frecuencia E5a. El estándar Galileo define 50 códigos PRN para cada uno de sus componentes en fase (E5a_I) y en cuadratura (E5a_Q). Por tanto, esta correlación debe ser ejecutada 100 veces, una por cada código PRN Galileo E5a_{I/Q}. Los códigos E5a y E5b son transmitidos con la misma modulación AltBOC puesto que están ubicados en distintas frecuencias. Por lo tanto pueden ser procesados de manera independiente. Los experimentos se han llevado a cabo tanto para los códigos presentes en la señal E5a como para aquellos presentes en la señal E5b. No obstante en este apartado, con el fin de no aportar información redundante, sólo se hará referencia a los códigos correspondientes a la señal E5a, ya que los resultados son enormemente similares.

La configuración para el motor de optimización NSGA-II se muestra en la tabla 5.1, y como se puede observar es similar al utilizado para el caso de GPS. Tal como se muestra, se definen dos objetivos de optimización.

Observando la ecuación 5.1, se puede concluir que al aumentar el valor de los picos de detección disminuye el valor del ruido de auto-correlación. No obstante, no se asegura que ocurra lo mismo para el ruido asociado a la correlación cruzada. Esta es la razón para maximizar el pico de detección y minimizar el ruido de correlación al mismo tiempo. En este escenario la población total se compone de 16 individuos y el número total de generaciones se limita a mil. La probabilidad de cruce se fija en 0,8 y la tasa de mutación es 1 sobre la longitud del individuo.

<i>Parámetro</i>	<i>Valor</i>
Número de Objetivo	2
Obj 1	-DP
Obj 2	N
Población	16
Generaciones	1000
Crossover	0,8
Mutación	1/Longitud PRN
Semilla	Aleatorio / Función mayoría junto con

Tabla 5.1. Parámetros para el algoritmo NSGA-II.

Originalmente se utilizó como semilla una matriz unidimensional binaria generada con una función de distribución uniforme. De esta manera, cada vez que un individuo nacía, su matriz era rellenada con valores aleatorios. Como regla general todos los individuos generados que utilizaban este procedimiento de inicialización tenían un alto nivel de ruido, y su pico de detección era menor que el ruido. Por lo tanto, los individuos propuestos inicialmente carecían de calidad suficiente. Esto hacía que la primera etapa del experimento con este tipo de inicialización, produjese individuos lejos del conjunto óptimo de Pareto de este problema de optimización. Este método de inicialización requiere además tiempos de ejecución elevados con el fin de alcanzar una solución cercana a la frontera de Pareto. Además, si consideramos que el esfuerzo computacional requerido para ejecutar la ecuación 5.1 crece con la longitud de los nuevos códigos PRN E5a de Galileo, los tiempos de ejecución necesarios hacen que esta metodología sea muy poco práctica.

Para lograr que el pico de detección crezca hasta un valor mayor que el del ruido producido, se lleva a cabo un nuevo procedimiento de inicialización

basado en la función mayoría. Este procedimiento se muestra en el siguiente código:

```

begin
Given PRN                                % Código Galileo E5a
Given bit-length                          % Tamaño del nuevo código
                                           % secciones = 10230 / longitud
                                           % Cada individuo es un array de bits
                                           % Bucle para indexar individuos
for i=1 to bit-length
  auxVal = 0
  for ii=1 to (10230/bit-length)          % Se acumulan los bits de las secciones con igual índice
    ipos=i+ii*bit-length                  % Se calcula el índice del PRN Galileo
                                           % Se cambia el rango [0,1] -> [-1,1] para considerar el valor 0
    auxVal = auxVal + PRN[ipos]*2-1
  end for ii
  if (auxVal>0)                            % Gana el valor 1
    Individual [i] = 1
  elseif (auxVal<0)                        % Gana el valor 0
    Individual [i] = 0
  else                                     % Mismo número de 0's y 1's
    Individual [i] uniform random bin [1,0]
  endif
end for i
end

```

Código 5.1. Algoritmo para la inicialización de individuos.

Este algoritmo requiere como entradas el número de secciones o la longitud correspondiente y el código PRN E5a Galileo original a optimizar. Lo que hace este algoritmo en líneas generales es aplicar la función mayoría para obtener cada uno de los bits de las secciones utilizadas con el fin de obtener la composición de cada individuo. La figura 5.2 muestra un ejemplo de ejecución del algoritmo de inicialización. Según se muestra en la figura, este algoritmo enumera cada matriz unidimensional que representa cada sección del código PRN Galileo E5a así como el individuo en sí, con un valor que va de 1 a la nueva longitud seleccionada. En el ejemplo concreto que se muestra en la figura 5.1 esta longitud es 5.115, es decir la mitad de un código Galileo convencional. En todas las matrices unidimensionales el bit de la izquierda conforma el índice 1, y el que se encuentra más a la derecha el valor más alto, que en este caso es 5.115. Por lo tanto el algoritmo mayoría contabilizará el número de bits que contiene cada sección con el mismo índice. Este procedimiento de 'cuenta y decisión' se ejecuta para cada uno de los índices bit a bit. Por ejemplo, el bit en la posición 1 de la sección 1, y el bit de la posición 1 de la sección 2 se estudian para generar el bit de la posición 1 del individuo resultante. Cuando el número de ceros es mayor que el de unos, el algoritmo determina que el valor correspondiente en el nuevo individuo a generar sea

también 0. Cuando el algoritmo encuentra mayor número de unos que de ceros, el algoritmo establece el valor 1 en la posición a determinar. En el caso de que el número de ceros y unos sea el mismo, el algoritmo elige un valor aleatorio para esa posición siguiendo una distribución uniforme. Esto queda representado en la figura por la letra 'R'.

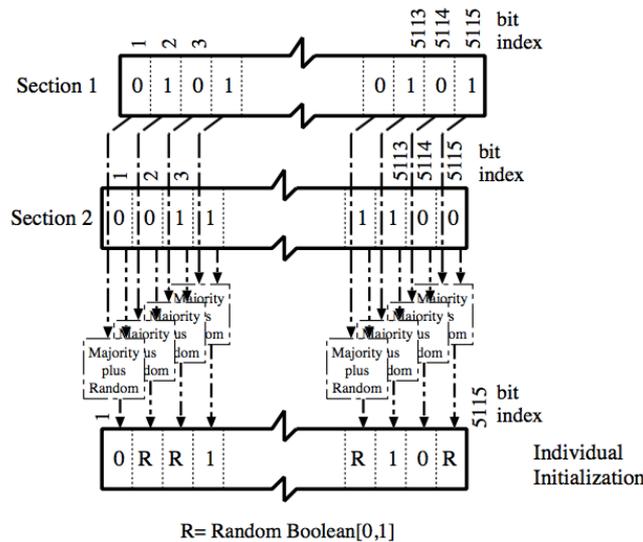


Figura 5.2. Ejemplo de la técnica de inicialización propuesta.

Con el fin de entender la siguiente explicación, así como los experimentos llevados a cabo, es necesario introducir la definición de brecha de detección o detection gap (D_G). La brecha de detección hace referencia a la distancia entre el pico de detección y el ruido. Un valor positivo de D_G implica que el valor del pico de detección es mayor que el ruido. Valores negativos de D_G implican que no es posible llevar a cabo la detección de un satélite utilizando el nuevo código propuesto.

La tabla 5.2 presenta una comparativa de ambas metodologías de generación de individuos. En la tabla se muestran los picos de detección y el ruido obtenidos al utilizar tanto la técnica de generación aleatoria, como la metodología basada en la función mayoría que se ha presentado en este apartado. Las dos primeras columnas muestran la configuración de los individuos, es decir, el número de secciones y la longitud de los códigos que será $10.230/\text{el número de secciones}$. Las siguientes tres columnas corresponden los valores DP, N y D_G cuando se aplica el generador de

números aleatorios con distribución uniforme. Las últimas tres columnas presentan los estos tres mismos parámetros al utilizar la inicialización basada en la función mayoría propuesta.

Características PRN		Generación Aleatoria			Nueva Propuesta		
Secciones	Longitud	DP	N	D _G	DP	N	D _G
10	1.023	108	159	-51	203	171	32
9,5	1.077	108	162	-54	265	171	94
9	1.137	112	167	-55	295	177	118
8,5	1.204	117	173	-56	264	160	104
8	1.279	121	178	-57	343	171	172
7,5	1.364	126	182	-56	362	188	174
7	1.461	130	189	-59	423	187	236
6,5	1.574	135	199	-64	422	196	226
6	1.705	141	207	-66	451	225	226
5,5	1.860	149	220	-71	638	208	430
5	2.046	158	225	-67	734	220	514
4,5	2.273	170	238	-68	783	247	536
4	2.556	184	253	-69	1.196	250	946
3,5	2.923	198	273	-75	1.453	263	1.190
3	3.410	214	293	-79	1.622	300	1.322
2,5	4.092	248	322	-74	2.002	312	1.653
2	5.115	274	359	-85	2.531	349	2.182

DP: detection peak; N: noise, D_G: detection gap

Tabla 5.2. Experimentos de inicialización de individuos para Galileo.

A la hora de llevar a cabo los ejercicios para este apartado, con el fin de mantener los tiempos de ejecución en valores prácticos, se han limitado el número de secciones a 10. Ya que para este caso, la longitud de los nuevos códigos propuestos es la misma que los códigos GPS L1 estudiados en anteriores apartados. Por otra parte el número mínimo de secciones se ha fijado en dos. Por lo tanto la tabla 5.2 presenta solamente los datos de los experimentos que se han llevado a cabo para secciones de entre 2 y 10 en pasos de 0.5. En dichos experimentos el valor del ruido ha sido mayor que el pico de detección para todos los casos en los que se ha utilizado la función aleatoria para inicializar al individuo. Esto se refleja en los valores negativos de

la quinta columna. Sin embargo al utilizar el nuevo algoritmo propuesto, el pico de detección es siempre mayor que el ruido. A partir de estos datos obtenidos se concluye que a menor número de secciones, mayor es el valor D_G . Además queda probado que el algoritmo de inicialización propuesto resulta mejor que utilizar un generador de números aleatorios basados en una distribución uniforme como ocurría en anteriores apartados.

Con el fin de evaluar la carga computacional al utilizar distintas longitudes de código o número de secciones, se considera el número total de operaciones de multiplicación/suma requeridas para evaluar la ecuación 5.1. Este parámetro ha sido señalado como Operaciones Mul/Sum en la siguiente tabla. La tabla 5.3 muestra los resultados obtenidos al aplicar la metodología propuesta en este apartado.

La primera columna muestra el número de secciones propuestas en el experimento de optimización. La segunda columna muestra la longitud de los individuos y/o las secciones. En la tercera columna se muestran el número de multiplicaciones/sumas necesarios para evaluar la ecuación 5.1 (Operaciones Mul/Sum). Las tres columnas siguientes muestran el pico de detección, el ruido y la brecha de detección obtenida utilizando la metodología propuesta en este apartado. La séptima columna muestra el cociente entre las multiplicaciones/sumas requeridas para las longitudes propuestas cuando la longitud de los individuos es la original: 10.230 (Ratio Mul/Sum). La octava columna indica el pico de detección logrado cuando se utiliza el mismo esfuerzo computacional que para un código PRN Galileo estándar (D_G equivalente). La última columna muestra la mejora de utilizar el nuevo código PRN obtenido en comparación con el código PRN estándar (mejora).

La primera fila en la tabla 5.3 representa los valores obtenidos al utilizar el código Galileo E5a directamente sin optimización, es decir el original de 10.230 bits. El resto de filas muestran los resultados para distintas longitudes, desde 1.023 hasta 5.115. Todos los códigos Galileo E5a obtenidos ofrecen valores de D_G positivos. Sin embargo, es necesario buscar su eficiencia en términos de valor del pico de detección en relación al número de multiplicaciones/suma requeridos. Como se muestra en la última columna, el peor caso aparece para

la longitud 5.115. A pesar de que el pico de detección en este caso es 2.248, la mejora es del -7,6%. Esto implica que se está utilizando un mayor esfuerzo computacional para obtener una detección que cuando se utiliza el código estándar. Por otra parte, la mayor eficiencia se obtiene cuando la longitud de código es 2.556. En este caso concreto se obtiene una mejora del 67,6% sobre el código Galileo E5a estándar.

Características PRN			Valores de correlación			Comparaciones		
Secciones	Longitud	Operaciones Mult/Sum	DP	N	D _G	Ratio Mult/Sum	D _G	Mejora (%)
1	10320	106502400	10320	498	9732	1	9732	1
2	5115	26163225	2553	305	2248	4	8992	-7,6
3	3410	11628100	1664	254	1410	9	12690	30,4
3,5	2923	8543929	1455	239	1216	12	14895	53
4	2556	6533136	1236	218	1018	16	16307	67,6
4,5	2273	5166529	849	203	646	20	13085	34,5
5	2046	4186116	756	196	560	25	14000	43,9
5,5	1860	3459600	674	186	488	30	14762	51,7
6	1705	2907025	517	175	342	36	12312	26,5
6,5	1574	2477476	490	170	320	42	13517	38,9
7	1461	2134521	441	165	276	49	13532	39,0
7,5	1364	1860496	398	160	238	56	13388	37,6
8	1279	1635841	367	157	210	64	13435	38
8,5	1204	1449616	302	150	152	72	10973	12,8
9	1137	1292769	319	145	174	81	14086	44,7
9,5	1077	1159929	283	141	142	90	12812	31,6
10	1023	1046529	239	141	98	100	9800	1

Operaciones Mult/Sum: Multiplicaciones/Sumas necesarios en la Ecuación 5.1

DP: detection peak; N: noise, D_G: detection gap

Ratio Mult/Sum: Operaciones Mult/Sum divididos entre las Operaciones necesarias para longitud 10230

D_G equivalente: Ratio Mult/Sum multiplicado por D_G

Mejora: D_G equivalente dividido por D_G cuando la longitud es 10.230

Tabla 5.3. Experimentos de inicialización de individuos para satélites E5a Galileo.

A la vista de estos resultados, tal como se esperaba, se puede observar que la metodología propuesta en este apartado para detectar satélites visibles utilizando los nuevos códigos PRN Galileo E5a requieren de un esfuerzo computacional menor, a cambio de la obtención de unos picos de detección

más bajos. Esto se traduce en una degradación de la señal en comparación con los códigos Galileo E5a estándar. En la figura 5.3 se muestra la atenuación de la señal recibida del satélite frente a la longitud de los códigos al ejecutar la ecuación 5.1 una sola vez. En el eje vertical se muestra la atenuación en decibelios (dB) mientras que en el eje horizontal se muestra la longitud de los códigos. Tal como se puede ver en la figura 5.3, la atenuación se hace mayor a medida que la longitud de los códigos se hace menor. La máxima atenuación es 20 dB cuando la longitud del código es de 1.023. Esta atenuación implica que la potencia mínima de la señal recibida del satélite ha de ser mayor que la del estándar Galileo.

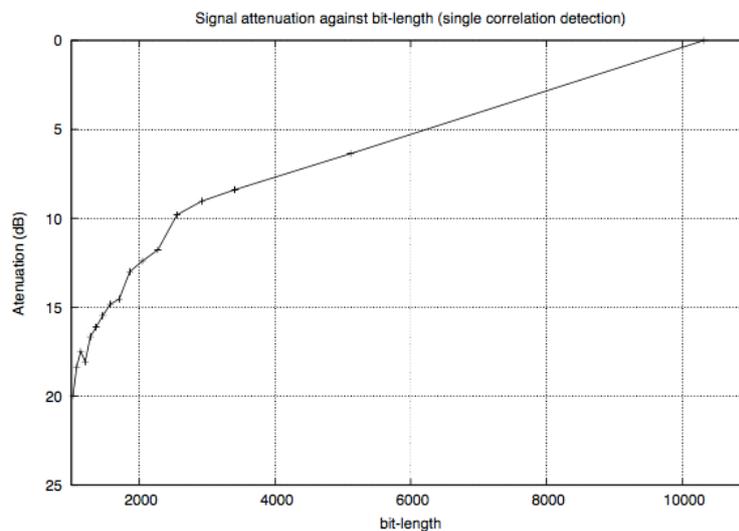


Figura 5.3. Atenuación de la señal que se encuentra al ejecutar el proceso de correlación una sola vez en términos de longitud del código.

Esta atenuación puede ser reducida repitiendo la ejecución de la ecuación 2.1 varias veces. Cada vez que se aplica la correlación, se reduce la atenuación, pero a la vez aumenta la carga computacional. La figura 5.4 muestra las mejoras obtenidas durante el proceso de convolución repetido tantas veces como el número de secciones en que se divide el código original.

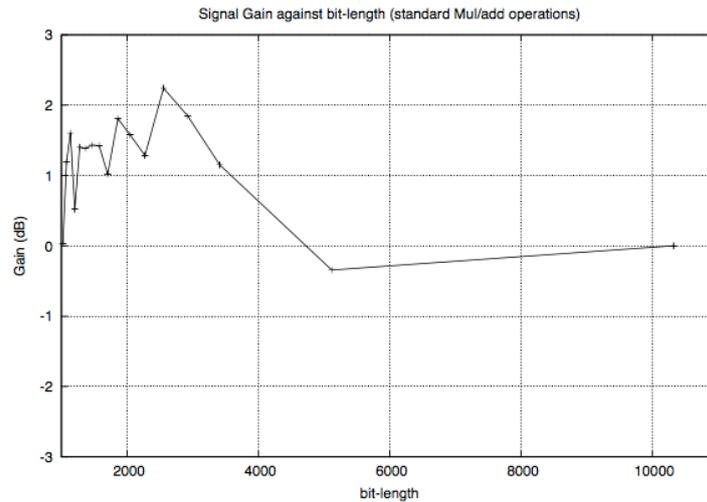


Figura 5.4. Ganancia de señal vs longitud de código para el mismo esfuerzo computacional que los códigos PRN estándar.

En el eje vertical y horizontal se pueden ver la ganancia en dB y la longitud del código. Los códigos PRN Galileo E5a, producen una ganancia positiva en la mayoría de los casos. Solamente para los códigos de longitud 5.115, es decir al utilizar 2 secciones, los códigos PRN E5a Galileo producen una degradación de la señal recibida del satélite.

La mejor ganancia es 2.2 dB cuando la longitud de código es 2.556, que es el caso en el que se toman 4 secciones. Esta solución es también el mejor punto de la Tabla 5.3 presentada anteriormente (con una mejora del 67,6%). La mejora media, tomando todos los códigos Galileo obtenidos, es 1,3 dB.

5.4. Conclusiones

A partir de los resultados obtenidos se puede concluir que la nueva metodología de detección presentada en este apartado produce mejores individuos semilla que en los apartados anteriores relativos a GPS. Esto es gracias al nuevo método de inicialización presentado. Se demuestra además que utilizando la metodología propuesta es posible generar nuevos códigos PRN de detección para la constelación de satélites Galileo que resultan más útiles que los originales en términos de esfuerzo computacional.

En este sentido, puesto que lo que hacen las metodologías propuestas es buscar la presencia de un patrón determinado en todas las secciones definidas de los códigos E5a Galileo, es posible concluir que los códigos E5a Galileo no

son números puramente pseudo-aleatorios. Es decir que los códigos E5a Galileo están formados por códigos PRN más cortos, de longitud cercana a 2.556 bits.

A la luz de los resultados obtenidos en este apartado, se pueden originar dos posibilidades similarmente ventajosas. Por una parte se hace posible la detección de satélites Galileo por parte de un receptor, con un considerable ahorro computacional, que revertirá directamente en un ahorro de batería, aunque con una pérdida de calidad de señal. Esta posibilidad es especialmente adecuada para localizaciones donde existe una buena visibilidad de satélites e interesa maximizar la autonomía de baterías por parte del receptor. Por otra parte, es posible mantener el esfuerzo computacional al mismo nivel que lo hacen los códigos originales, enfocándose en la obtención de una mejor calidad de señal. Esto resulta especialmente interesante en aplicaciones en las que el receptor no tiene una buena visibilidad de los satélites, y la señal llega degradada.

Es importante señalar que, una vez se han obtenido estos nuevos códigos de detección utilizando algoritmos genéticos y NSGA-II, esta metodología puede ser aplicada a la práctica totalidad de los receptores GNSS actuales sin que sea necesaria ninguna modificación de hardware.

Capítulo 6

Conclusiones

La presente Tesis Doctoral se centra en aportaciones en el ámbito de la geolocalización explorando distintos métodos con el fin de mejorar la eficiencia en el proceso de detección de satélites desde el punto de vista del esfuerzo computacional. Para ello se ha comenzado introduciendo los conceptos básicos relativos a la geolocalización así como el funcionamiento y los diferentes componentes del sistema de posicionamiento GPS, y el papel que juegan en el mismo. Una vez descritos estos conceptos básicos, se ha introducido el método de triangulación utilizado para determinar la posición de un receptor.

A continuación se ha dedicado un segundo capítulo a describir los procesos que tienen lugar en el receptor GPS para llevar a cabo la detección de satélites. Se introduce el término de secuencias pseudoaleatorias o PRN que identifican a cada uno de los satélites, describiendo las propiedades que los hacen tan interesantes para llevar a cabo la detección de satélites.

Una vez definido el ámbito bajo estudio, se comienzan presentando métodos deterministas para obtener códigos ortogonales de menor costo computacional. Se exploran varios métodos que generan códigos de longitudes comprendidas entre los 512 y 128 bits, a partir de los cuales se obtienen distintas ventajas relativas a reducción de potencia y/o costo computacional con respecto a los códigos PRN originales de longitud 1023.

Tras los resultados obtenidos con la metodología determinista, se introduce el concepto de algoritmo genético, como método alternativo para lograr códigos más eficientes. Se exploran diferentes opciones utilizando distintos motores de optimización siendo el NSGA-II el que mejores resultados ofrece conforme aumenta el número total de generaciones.

Con el objetivo de validar la metodología propuesta en esta Tesis Doctoral, se han introducido al estudio con algoritmos genéticos los factores relativos a la atenuación. Sin dejar de lado el punto de vista de la implementación hardware del receptor, se han aplicado las metodologías propuestas al sistema de geolocalización europeo (GALILEO), presentándose en este caso los resultados obtenidos con el uso del motor de búsqueda NSGA-II.

Por último, se aporta una nueva técnica de inicialización de los individuos para los métodos heurísticos basados en algoritmos genéticos, que demuestran generar semillas de inicialización mejores que las desarrolladas en los capítulos anteriores.

La consecución del objetivo planteado en este último capítulo, demuestra que las metodologías propuestas pueden ser aplicadas a la práctica totalidad de los receptores GNSS actuales sin necesidad de modificar el hardware existente.

Bibliografía

- [1] E. D. Kaplan, Understanding GPS: principles and application. Boston, MA: Artech House Publishers, 1996.
- [2] USAF. Navstar GPS, "Where am I? Are we there yet?", Air & Space Power Journal, Vol. 6 No. 2, pp. 182-197 (2003).
- [3] Lohnert E., wittmann E., Pielmeier J. and Sayda F., PARAMOUNT public safety & commercial Info-Mobility applications & Services in the Mountains. ION GPS 2001 , 11-14 september 2001.
- [4] S. Gunawardena and F. van Graas, "Split-sum correlator simplifies range computations in GPS receiver", Electronic Letters, vol 42, issue 25, pp 1469-1471, Dec 2006.
- [5] Michalski, J. Czajewski, The accuracy of the global positioning systems, IEEE Instrumentation and Measurement Mag. 7(1), 56–60 (2004).
- [6] R. Prasad, M. Ruggieri, Applied Satellite Navigation Using GPS, GALILEO, and Augmentation Systems, Artech House Publishers (2005).
- [7] J. B. Woodford and H. Nakamura, "Briefing- Navigation Satellite Study", 24 August, 1966.
- [8] Proc, Jerry, "Loran-A", Hyperbolic Radionavigation Systems, Etobicoke, Ontario, 2012.
- [9] http://m.esa.int/Our_Activities/Navigation/The_future_-_Galileo/What_is_Galileo consultada por última vez en Abril de 2017.
- [10] Helen E. Worth and Mame Warren. Transit to Tomorrow. Fifty Years of Space Research at The Johns Hopkins University Applied Physics Laboratory, 2009.
- [11] C.-L. Chang and J.-C. Juang, "Adaptive logic control approach for fast GNSS acquisition", Electronics Letters, vol 44, issue 28, pp 821-822, Jun 2008.
- [12] Donald E. Gustafson, John R. Dowdle, John M. Elwell, and Karl W. Flueckiger, "A nonlinear code tracking filter for GPS-Based Navigation", Selected Topics in Signal Processing, IEEE Journal of, vol 9, issue 4, pp 627-638, Aug 2009.

- [13] S. A. Dale and P. Daly, "The Soviet Union's GLONASS Navigation Satellites," in *IEEE Aerospace and Electronic Systems Magazine*, vol. 2, no. 5, pp. 13-17, May 1987.
- [14] Cao, F., Yang, X., Li, Z., Sun, B., Kong, Y., Chen, L., & Feng, C. (2014). Orbit determination and prediction of GEO satellite of BeiDou during repositioning maneuver. *Advances in Space Research*, 54(9), 1828-1837.
- [15] Zhao, L.; Liu, A.; Ding, J.; Wang, J. BeiDou Signal Acquisition with Neumann–Hoffman Code Modulation in a Degraded Channel. *Sensors* 2017, 17, 323.
- [16] Ahmed Nait-Sidi-Moh, Mohamed Bakhouya, Jaafar Gaber, Maxime Wack, "Geopositioning and Mobility" ISBN 9781118743683.
- [17] USAF. Navstar GPS, "Where am I? Are we there yet?", *Air & Space Power Journal*, Vol. 6 No. 2, pp. 182-197 (2003).
- [18] Flores, A., de Arellano, J.V.-G., Gradinarsky, L.P. and Rius, A., "Tomography of the lower troposphere using a small dense network of GPS receivers", *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 39 No. 2, pp. 439-447 (2001).
- [19] Lin, T.-H., Hsu, N.C., Tsay, S.-C. and Huang, S.-J., "Asian dust weather categorization with satellite and surface observations", *International Journal on Remote Sens*, Vol. 32 (2011).
- [20] Jerrett, D. and Nash, J., "Potential uses of surface based GPS water vapour measurements for meteorological purposes", *Physics and Chemistry of the Earth International Journal, Part A: Solid Earth and Geodesy*, Vol. 26 Nos 6-8 (2001).
- [21] Hsin-Min, P., Chia-Shu, L. and Jeng-Kuang, H., "Performance testing of time comparison using GPS-smoothed P3 code and IGS ephemerides", *IEEE Transactions on Instrum and Measurement*, Vol. 54 No. 2, pp. 825-828 (2005).
- [22] <http://airforcelive.dodlive.mil/2015/03/10-common-myths-surrounding-gps/> consultada por última vez en Abril de 2017.
- [23] Borre, Akos, Bertelsen, Rinder, Holdt Jensen, "A software-defined GPS and GALILEO receiver", Birkhäuser Boston, 2007.

- [24] Hamza, G., Zekry, A. and Motawie, I., "Implementation of a complete GPS receiver using simulink", IEEE Circuits and Systems Magazine, Vol. 9 No. 4, pp. 43-51 (2009).
- [25] Klukas, R., Lachapelle, G., Ma, C. and Jee, G.-I., "GPS signal fading model for urban centres", IEEE Proceedings on Microwaves, Antennas and Propagation, Vol. 150 No. 4, pp. 245-252 (2003).
- [26] Larson, K.M., Braun, J.J., Small, E.E., Zavorotny, V.U., Gutmann, E.D. and Bilich, A.L., "GPS multipath and its relation to near-surface soil moisture content", IEEE Journal of Selected Topics in Appl. Earth Observations and Remote Sensing, Vol. 3 No. 1, pp. 91-99 (2010).
- [27] Chih-Hung, W., Wei-Han, S. and Ya-Wei, H., "A study on GPS GDOP approximation using support-vector machines", IEEE Transactions on Instrumentation and Measurement, Vol. 60 No. 1, pp. 137-145. (2011).
- [28] Duncan, M.J., Badland, H.M. and Mummery, W.K., "Applying GPS to enhance understanding of transport-related physical activity", Journal of Science and Medicine in Sport, Vol. 12 No. 5, pp. 549-556 (2009).
- [29] Toledo-Moreo, R., Zamora-Izquierdo, M.A., Úbeda-Miarro, B. and Gómez-Skarmeta, A.F., "High-integrity IMM-EKF-based road vehicle navigation with low-cost GPS/SBAS/INS", IEEE Transactions on Intelligent Transportation Systems, Vol. 8 No. 3, pp. 491-511 (2007).
- [30] R. Gold, "Optimal binary sequences for spread spectrum multiplexing (Corresp.)," in IEEE Transactions on Information Theory, vol. 13, no. 4, pp. 619-621, October 1967.
- [31] Kaplan, E. and Hegarty, C., Understanding GPS Principles and Applications, Artech House Publishers, Norwood, MA, (1996).
- [32] Sosa, J., Montiel-Nelson, J.A. and Nooshabadi, S., "Low power GPS pseudo random numbers using genetic algorithms", paper presented at Evolutionary and Deterministic Methods for Design, Optimisation and Control with Applications to Industrial and Societal Problems, EUROGEN11, Capua, Italy (2011).

- [33] Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T., "A fast and elitist multi-objective genetic algorithm: NSGA-II", *IEEE Transactions on Evolutionary Computation*, Vol. 6 No. 2, pp. 182-197 (2002).
- [34] Deb, K. and Tiwari, S., "Omni-optimizer: a generic evolutionary algorithm for single and multi-objective optimization", *European Journal of Operational Research*, Vol. 185 No. 3, pp. 1062-1087 (2008).
- [35] Sosa, J., Montiel-Nelson, J.A., Navarro, H. and García, J.C., "Minimum power consumption optimization using genetic algorithms", paper presented at *Evolutionary and Deterministic Methods for Design, Optimisation and Control with Applications to Industrial and Societal Problems, EUROGEN05*, Munich, Germany (2005).
- [36] Seung-Woo Lee, Jeongrae Kim, Myeong-Sook Jeong, Young Jae Lee, Monitoring atomic clocks on board GNSS satellites, *Advances in Space Research*, 47(10), 1654–1663 (2011).
- [37] Toho Herman Diessongo, Torben Schaler, Stefan Junker, Precise position determination using a Galileo E5 single-frequency receiver. *GPS Solutions*, Volume 18, Issue 1, pp 73–83, January 2014.
- [38] Michael J. Dunn, *Navstar GPS Space Segment/Navigation User Interfaces (IS-GPS-200)*, U.S. Coast Guard Navigation Center (2012).
- [39] Global Positioning System Directorate (SMC/GP), *Systems Engineering and Integration (SE&I), Interface Specification*. Department of the Air Force, USA. 2013.
- [40] Jung Yi Lin, *Using Evolutionary Computation on GPS Position Correction*, *The Scientific World Journal*, Volume 2014, Hindawi Publishing Corporation, 2014.
- [41] *European GNSS (Galileo) Open Service: Signal in Space Interface Control Document*. Luxembourg: Publications Office of the European Union, 2010.
- [42] Guenter W. Hein, Jeremie Godet, Jean-Luc Issler, Jean-Christophe Martin, Rafael Lucas-Rodriguez, Tony Pratt, *Status of Galileo frequency and signal design*. Proc. ION GPS, Portland, Oregon, USA, The Institute of Navigation, 2002.

[43] L. Esteve, J. Arribas, C. Fernandez-Prades, P. Closas, First Acquisition of Galileo E1C Signals. Report. 2012.

Anexo I

Publicaciones

La investigación desarrollada a lo largo de esta Tesis Doctoral ha originado las siguientes publicaciones:

Alcaraz, D., Sosa, J., & Montiel-Nelson, J. A. (2012). A novel fast and efficient GPS detection method. DCIS 2012 conference Avignon (France) proceedings, 498-499. ISBN: 978-2-9517461-1-14.

Sosa, J., Bautista, T., Alcaraz, D., Nelson, J. A. M., & Alonso, S. G. (2013). Generation of New Detection Codes for GPS Satellites Using NSGA-II. In EUROGEN 2013 International Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems. Las Palmas de Gran Canaria (Spain).

Sosa, J., Alcaraz, D., Bautista, T., Montiel-Nelson, J. A., García-Alonso, S., Monzón-Verona, J. M., et al. (2013). Lite GPS pseudo-code generation using genetic algorithms. *Engineering Computations*, 30(8), 1170–1192.

Sosa, J., Bautista, T., Alcaraz, D., García-Alonso, S., & Montiel-Nelson, J. A. (2015). Generation of New Detection Codes for GPS Satellites Using NSGA-II. In D. Greiner, B. Galván, J. Périaux, N. Gauger, K. Giannakoglou, & G. Winter (Eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences* (pp. 511–520). *Computational Methods in Applied Sciences*, 36. Springer International Publishing. ISBN: 978-3-319-11541-2.

Alcaraz, D., Sosa, J., & Montiel-Nelson, J. A. (2012). A novel fast and efficient GPS detection method. DCIS 2012 conference Avignon (France) proceedings, 498-499. ISBN: 978-2-9517461-1-14.

A novel, fast and efficient GPS detection method

Daniel Alcaraz Real-Arce, J. Sosa and Juan A. Montiel-Nelson

Institute for Applied Microelectronics
 Department of Electronics Engineering and Automatics
 University of Las Palmas de Gran Canaria
 E35017 Las Palmas de Gran Canaria
 Email: {dalcaraz, jsosa, montiel}@iuma.ulpgc.es

Abstract—The process by which a Global Navigation Satellite System (GNSS) receiver determines its position involves a few stages. One of the highest time consuming operations in this process implies detecting and tracking the visible satellites. This procedure is performed using the pseudo random codes that identifies univocally each satellite. In this paper, we introduce a new method that reduce the computational effort for detecting GPS satellites. Presented results prove that the detection process can be achieved using just the 11.11% of the required computation effort in comparison with the traditional detection method.

Index Terms—GNSS, GPS, Gold Code, C/A Code, Satellite, Power Saving, Detection.

I. INTRODUCTION

Determine the position where an object or a person is located on the Earth surface has been one of the most important progresses achieved on the last decades [1]. On the seventies, the U.S. army developed the Global Positioning System (GPS) which was the first worldwide Global Navigation Satellite System (GNSS) [2]. The GPS constellation is composed by at least 32 satellites, where only 24 are active. It was developed as a military tool, but nowadays the wide range of civil applications using it have overridden its original military orientation [3]. In this scenario, results clear that GNSS and in particular GPS turns as a very interesting research area for scientists.

II. GPS DETECTION METHOD

The principle in which the GPS system is based, is to make measurements of the propagation time of signals transmitted by the satellites. As these signals travel at the speed of light, it is quite simple to obtain the distance between the transmitters and the receiver. The position establishment is made when at least four satellites are detected and the delays between the GPS satellite signals are determined. The C/A code is a sequence of 1,023 bits. This sequence is modulated on BPSK and transmitted each ms continuously by the satellite at 1,575.42 MHz. These C/A codes belong to a unique family of sequences referred as Gold Codes or pseudo-random noise sequences (PRN sequences). The operation which allows detection is called correlation, and can be written as:

$$corr_{ik}(m) = \sum_{l=0}^{1022} C^i(l)C^k(l+m) \quad (1)$$

The C/A codes properties assures that for two given codes C_i and C_k , the cross correlation, i.e. the correlation of two different codes, results on a near zero value. The same result is obtained when a code is correlated with itself, but it is not aligned. The unique case where the correlation gives a high value is when the same code is correlated with itself and it is also aligned. In this case, this correlation peak is clearly distinguishable from the rest of values, making the process of detection a simple matter. If the correlation of two codes gives a value over a certain threshold, the corresponding satellite is present. Other way it is not. This process must be repeated, correlating every code with each other, in order to determine which satellites can be detected.

Another fact to consider on the detection process is the Doppler frequency shift caused by the motion of the satellite relative to the receiver. For a GPS receiver moving at high speed it is reasonable to assume that the maximum Doppler frequency shift is around ± 10 kHz. In order to ensure the receiver does not lose any visible satellite, this Doppler frequency margin is scanned on bins of 500Hz. Considering this, the GPS receiver must search each satellite in 41 different frequencies. Thus, to detect a satellite a double search must be done in both frequency and code phase. As each PRN is 1,023 bits long, the correlation between two of them implies $1,023 \times 1,023$ binary XOR operations. This number must be multiplied by 32 as the receiver must check all the possible satellites. But taking on account the Doppler frequency shift, the total number of XOR operations to determine all visible GPS satellites ascends to 1,373,046,048.

TABLE I
 TRUTH TABLE FOR THE NEW 341 BITS PRN.

A(i)	B(i)	C(i)	nPRN(i)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

III. PROPOSED DETECTION METHOD

As there is not much we can do to avoid the Doppler frequency shift, the attention must be focused on the PRN codes length. The idea is to generate shorter PRN codes to reduce the code phase search in the receivers detection process. To achieve this, an original PRN code is sampled in order

to form shorter sequences from which obtain a new shorter PRN code. As this new PRN code has been synthesized from the original one, it satisfies the same correlation properties, and thus it assures detection. Sampling the PRN alternatively, several PRN sequences can be shaped. Combining the PRN samples of these sequences results in a shorter PRN code that leads to a computational cost saving.

As said before, the length of the original PRN is 1,023, which can be factorized by 3, 11 and 31. Multiplying 31 by 11, 341 can be taken as the new PRN code length. In order to obtain this new PRN code, three sequences must be shaped from the original PRN and its samples must be combined to synthesize a final 341 bits sequence. Sampling the original PRN code at one third of the frequency, three sequences of 341 bits length can be formed. For a given PRN code:

$$PRN^{sat} = [a_1, a_2, a_3, \dots, a_{1023}]; \quad (2)$$

The three sequences will be formed as follows:

$$A = [a_1, a_4, a_7, \dots, a_{1021}], \quad (3)$$

$$B = [a_2, a_5, a_8, \dots, a_{1022}], \quad (4)$$

$$C = [a_3, a_6, a_9, \dots, a_{1023}], \quad (5)$$

The new PRN code can be calculated combining the bits of these three sequences in according to Table I. This new PRN assures detection requiring only a 11.11% of the computation needed in the usual method, as only one third of the information will be processed.

TABLE II
CORRELATION VALUES USING THE 341 PRN CODES FREQ.

DETECTION				NON-DETECTION	
Max. Peak	Min. Peak	Max. Noise	Mean Noise	Max. Noise	Mean Noise
201	133	81	14.88	105	14.73

IV. EXPERIMENTAL RESULTS

The method above explained was simulated on Matlab to prove that detection is possible using the new PRN codes. The simulation made involved the correlation between the new PRN code of each satellite and the three sequences (A, B and C) obtained from sampling the 32 original PRN. That is, 96 correlations for each satellite. After the simulation, the success of detection using these new 341 bits long PRN is proved, as the detection peak only appears when they are correlated with the corresponding three sequences from the original PRN.

Table II shows the values obtained for the maximum and minimum peaks, and the maximum and mean noise for the detection and non-detection cases. In this context, all the correlation results that are not detection peaks are considered noise, as it will interfere the detection process. For all the cases, the detection peak values obtained are between 201 and 133 while the noise value is around 15, which assures detection for all the satellites. On the non-detection cases, the noise is around 14 with maximum peaks under 100. So there

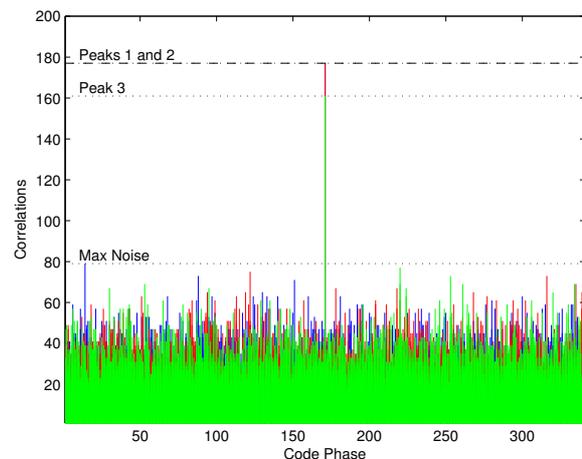


Fig. 1. Correlation of the nPRN 1 with the corresponding three sequences.

is no place for a false detection. There is an exception for the 23th satellite, where the noise value is close to 105 in a particular case (when correlated with the 31st satellite PRN code). This is a unique case and does not affect the detection of the satellite, as its lower detection peak is 157, which is far enough from 105.

Fig. 1 presents graphically the case where the new PRN code generated for the first satellite is correlated with all the PRN codes. The abscissa axis represents the code phase, which goes from 1 to 341, while the ordinate axis gives the value obtained on the correlation of the sequences for each code phase. The figure shows that there are three peaks that indicate detection, one for each A, B and C subsequences. The peak values are far away from the noise plate, being the maximum noise value 77. It is clear that there is enough margin to assure that it will not be any false detection.

V. CONCLUSION

Through this paper a new method for detection of GPS satellites have been proposed. This method has been focussed on the generation of new shorter PRN codes. The results obtained have revealed that detection can be achieved saving a 88.89% of the computation needed originally.

ACKNOWLEDGMENT

This work was funded under projects TEC2011-29148-C02-01 by the Spanish Ministry of Science and Technology and SolSubC200801000282 by the Canary Agency for Research, Innovation and Information Society.

REFERENCES

- [1] U.S. Air Force, NAVSTAR GPS: where am I? Are we there yet?, Air & Space Power Journal, Sep 2003.
- [2] Scott Gleason, Demoz Gebre-egiabher, GNSS Applications and Methods (GNSS Technology and Applications), Artech House Publishers, ISBN 1596933291.
- [3] Thomas Pany, Navigation Signal Processing for GNSS Software Receivers (GNSS Technology and Applications), Artech House Publishers, ISBN 1608070271.

Sosa, J., Bautista, T., Alcaraz, D., Nelson, J. A. M., & Alonso, S. G. (2013). Generation of New Detection Codes for GPS Satellites Using NSGA-II. In EUROGEN 2013 International Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems. Las Palmas de Gran Canaria (Spain).

Generation of New Detection Codes for GPS Satellites Using NSGA-II

J. Sosa*, Tomás Bautista, Daniel Alcaraz, Juan A. Montiel-Nelson and S. García-Alonso

Institute for Applied Microelectronics (IUMA),

Dept. of Electronic Engineering and Automatics (DIEA),

University of Las Palmas de Gran Canaria (ULPGC), 35015 Las Palmas de GC, Spain

Email: jsosa@iuma.ulpgc.es

Summary

In this paper we obtain new detection codes, to determine whether a GPS satellite in particular is visible, using NSGA-II as multi-objective optimization engine. Our approach takes in consideration the length of the code and the sampling frequency in comparison with other approaches found in the literature that fix those design parameters. Results demonstrate that both design parameters must be taken in consideration to obtain high quality detection codes.

Keywords: GNSS, GPS, Gold Code, Low computational effort, Genetic Algorithms, Multi-objective optimization

1 Introduction

Nowadays, the detection of GPS signals for performing location tasks is one of the most commonly demanded applications. In particular, the fast expansion of the mobile telephony, the increasing of the CPU capabilities and the reduced battery charging times provide a wide market where GPS applications are potentially explored. However, in this scenario, since the GPS receiver just becomes one application running concurrently with some others in a single device, the requirements of low computational effort and low power consumption are mandatory. Reducing the computational effort to determine whether a GPS satellite is visible or not is a hot topic in this research area. In the existing literature three basic approaches are presented to cope with this problem. The first one consists in reusing most of the computation with additional hardware. This approach is called *split-sum methodology*¹. Other authors propose to obtain a single detection code that allows to know if more than one satellite is visible or not.² Finally, in another approach³ authors present a methodology to obtain GPS detection codes of 341 bits achieving a lower computational effort. In this paper we explore the idea of obtaining reduced length detection codes for GPS satellites presented in³ using a

multi-objective approach and we introduce as new optimization variables the length of the reduced code and the sampling frequency.

2 Problem Definition

In order to detect whether a satellite is visible or not, receivers compare the incoming GPS identifiers (at a frequency L1 of 1,575.42 MHz), with all the possible GPS satellite identifiers. A satellite identifier is a Pseudo Random Number (PRN). Basically, a PRN is an array of binary digits where each digit is called *chip*. The length of the PRN array is 1023 chips. Each satellite has assigned an unique PRN as identifier. Every satellite transmits its own PRN identifier every millisecond. The comparison function is as follows:

$$\text{Comp}(\text{PRN}^{\text{sat}}, D^{rx}) = [a_1, a_2, \dots, a_L] \quad (1)$$

$$a_i = \sum_{j=1}^L \text{PRN}_{\text{mod}_L(i+j)}^{\text{sat}} * D_i^{rx}$$

where L is the length of the PRN, that is, 1023 chips for each GPS satellite. D^{rx} is the incoming radio frequency data that is acquired by the GPS analog front-end receiver. PRN^{sat} is the PRN identification of the GPS satellite (*sat*).

This function takes in consideration all the possible alignments between the incoming GPS identification data and the compared PRN. Therefore, this is the reason to obtain an array of values and not only a single value. The size of array $Comp(PRN^{sat}, D^x)$ is L , one value for each possible alignment between the incoming data and the compared PRN.

Following the theory, when a satellite is visible the array $Comp(PRN^{sat}, D^x)$ has an unique maximum. This maximum value is called Detection Peak (D_P). The location/index where the D_P is placed in the comparison array is called code-phase. The code-phase determines the starting chip of the PRN sequence. The other values in this array are lower than the D_P value. These lower values are called Noise (N). If the compared satellite (PRN^{sat}) is not visible, all values in array $Comp(PRN^{sat}, D^x)$ are noise.

Nowadays, in the literature there exist multiple approaches to implement the detection equation 1. For instance, it is quite easy to translate this equation 1 from the time domain to the frequency domain and use the Fast Fourier Transform (FFT) and its inverse (IFFT) to obtain the same results. However, in practical GPS receivers there exist only two basic detection techniques. Their main key in comparison with other approaches are their implementation simplicity and the required computational effort.

In one of these approaches the L1 incoming signal is oversampled, that is, every chip of the PRN is sampled more than once, so the D_P grows with the increasing sampling frequency. In terms of equation 1, implementing the oversampling only requires to set the correct value to L .

The other solution, instead of increasing the sampling frequency, increases the sampling period. As a consequence, the recorded incoming GPS data contains more than one complete PRN sequence. Therefore, the Detection Peak increases its value in proportion to the increment of the sampling period, that is, the signal recording time is increased.

Stepping up the sampling frequency and/or the sampling period increases the sensibility of a GPS receiver when the GPS signal-to-noise ratio is too low. However, ordinary applications like open-sky navigation systems, i.e., typical GPS receiver for car tracking, only takes one of both methodologies with reduced increasing factors.

In this research, we propose to obtain new detection

codes to determine whether a satellite is visible or not. The main feature of those new codes is their reduced length in comparison with the original PRN. Proposed new codes require lower computational effort than the traditional PRN. Moreover, we are based on the approach presented in³ where authors introduce a novel methodology to obtain new PRN detection codes using GA. In this previous approach, the research is focused in determining what kind of multi/single objective algorithm is more suitable for this type of application. However, in order to narrow the search space of the problem, authors fix the length of the new detection codes to a submultiple of the original length. This submultiple is 341 chips ($341 \times 3 = 1023$).

Our proposal in this work is redefining the problem presented in³ with two new variables to optimize. The first is the length of the detection code and the second is the sampling ratio. We define the sampling ratio as the size of the new detection code divided by 1023 chips. Since we reduce the sampling ratio below the unity, we will use the term dropping ratio as a more adequate concept or definition. The approach presented in³ has a length of 341 and a dropping ratio of 3 (that is, to take 1 sample and drop 2 for every 3 samples of the incoming signal). This results in a sampling ratio of $1/3$.

3 Problem codification

Based on the approaches found in the literature, we choose the NSGA-II⁴ as GA multi-objective optimization engine. The codification for the individual is as follows. Each individual represents a new detection code. The individual is made by an array of Boolean values. The status of each Boolean value can be only a logic zero and a logic one. The unknown or error states are forbidden in this engine. The length of the individual array determines the length of the new detection code. Finally, each element of the individual array corresponds with an element of the new detection code; this makes that the index is the same in both arrays.

Unsurprisingly, the cost function is basically the equation 1. As before mentioned, this cost function is an array of comparison values. However we are not interested in all these values but our attention is focused on the Detection Peak and the Noise. Moreover, we know that the dropping ratio determines the total number of Detection Peaks in the comparison array. For example, if the length of the new detection code is set to 341 and the dropping

ratio is set to 3, then there exist 3 different Detection Peaks. So the comparison array contains as many Detection Peaks as set the dropping ratio. Other values on this comparison array are Noise values.

We are interested in obtaining new reduced GPS detection codes with a low Noise and high Detection Peaks. In order to obtain those values, after evaluating each individual with equation 1, the evaluation function extracts the minimum Detection Peak and the maximum Noise from the computed comparison array.

4 Experiments

We assume that the search space of this problem grows with the length of the new detection codes and dropping ratio used. If we provide all the degrees of freedom on all our problem variables, the required computational effort is increased in great matter. The magnitude of this computational effort produces non-practical CPU execution times. In this sense, in order to obtain the solutions in a practical CPU time, we include the following rule:

$$|Proposed\ length \times Dropping\ Ratio - 1023| < \xi \quad (2)$$

where ξ is the maximum allowed deviation between the original PRN and new proposed PRN.

Table 1 presents the NSGA-II optimization engine setup. We define two objective functions to optimize. The first one is to maximize D_p (minimize $-D_p$) and the second is to minimize the Noise (N). The NSGA-II optimization engine is controlled by a program written in C that proposes lengths and sizes randomly using the rule introduced in equation 2. In other words, our applica-

Table 1: NSGA-II setup in our experiments.

Parameter	Value
Num. Objectives	2
OBJ_1	$-D_p$
OBJ_2	N
Population	40
Generations	10k/100k
Crossover	0.8
Mutation	1/PRN length
Seed	Random / Uniform Distribution

tion (C program) proposes different lengths and dropping

ratios and the NSGA-II optimization engine obtains new detection codes. That is, the application starts with a proposal of a new length and dropping ratio to obtain a new detection code. Then the NSGA-II optimization engine takes the control and produces a Pareto-Front curve for those parameters. Once the optimization is finished, our application extracts the best solution of the Pareto-Front. This procedure starts over and over with several lengths and dropping ratios. In addition, after extracting the best solution our application evaluates the convergence of the proposed parameters. If the application determines that the solution can be refined/improved the NSGA-II is executed again with more generations (from 10k to 100k).

Table 2 presents some results of our application when we look for reduced codes of GPS satellite ID 1. In this experiment we set ξ to a maximum of 400 chips. The first and second columns of the table contain the proposed code length and the dropping ratio. The number of generations are shown in the third column (in terms of 1k generations). The following two columns exhibit the best optimized minimum Detection Peak and maximum Noise. The sixth column indicates the detection gap, that is, the margin between the minimum Detection Peak and the maximum Noise. The following column measures the required computational effort in terms of multiplication/comparison operators required to execute equation 1. The eighth column evaluates previous CPU effort in terms of detection gap. Finally, the last column presents the comparison between the new proposal and the traditional methodology for the CPU effort in terms of the detection gap. In addition, please note that the last two rows in this table 2 show the best solution obtained in³ and also using the traditional methodology, respectively. As expected, results from table 2 are better as greater is the total number of generations in terms of maximum Noise and/or minimum Detection Peak. Last column presents a comparison between the traditional methodology and our new reduced detection code. The comparison is done in terms of required computational effort and detection gap to determine whether a satellite is visible or not. The negative values in this column denote better solutions than using the traditional methodology.

From table 2, we observed that, for instance, the combination 320×3 (length \times dropping ratio) with 100k generations and 212×3 with 10k generations have similar values in last column, -73 and -76 respectively. The first

Table 2: NSGA-II experiments for Satellite ID 1 and $\xi < 400$. *1 Data from³. *2 Traditional Methodology.

Code Length	Dropping Ratio	Generations ($\times 1k$)	Min(D_P)	Max(N)	$D_P - N$	CPU Effort Units	CPU / ($D_P - N$)	CPU diff
512	2	10	276	72	204	262,144	1285.0	188
212	3	10	90	46	44	44,944	1021.5	-76
288	3	10	116	54	62	82,944	1337.8	241
304	3	10	126	56	70	92,416	1320.2	223
320	3	10	134	56	78	102,400	1312.8	216
341	3	10	141	59	82	116,281	1418.1	321
256	4	10	90	52	38	65,536	1724.6	628
512	2	100	276	70	206	262,144	1272.5	176
288	3	100	136	54	82	82,944	1011.5	-85
304	3	100	148	56	92	92,416	1004.5	-92
320	3	100	156	56	100	102,400	1024.0	-73
341	3	100	177	57	120	116,281	969.0	-128
256	4	100	98	52	46	65,536	1424.7	328
341	3	40	177	89	88	116,281	1321.4	224 *1
1023	1	-	1023	69	954	1,046,529	1097.0	0 *2

one has the double of detection gap than the second one, but the second proposal requires a half of the CPU Effort to evaluate equation 1. Therefore, this comparison demonstrates that there exist several combinations of code lengths and dropping ratios that have similar ratios of CPU effort and detection gap.

The results obtained with our application, as shown on table 1, are always better than the presented in a previous work³ in terms of maximum Noise. In case of the Detection Peaks our approach obtains at least the same or better values than those in the referred work³.

5 Conclusions

In this paper a methodology is proposed to take into account new design parameters to obtain reduced GPS detection codes to determine whether a satellite is visible or not. Those new design parameters are the length of the reduced detection code and the dropping ratio. New optimal detection codes are obtained using NSGA-II as optimization engine. Results demonstrate that new detection codes exist that exhibit both similar or better performance in terms of CPU effort, detection gap, code length and/or dropping ratio.

6 Acknowledgements

This work is patent pending and was funded under project BATTLEWISE (TEC2011-29148-C02-01) of the Ministry of Economy and Competitiveness.

References

- [1] Gunawardena, S. and van Graas, F. Split-sum correlator simplifies range computations in gps receiver. *Electronics Letters* **42**(25), 1469–1471 (2006).
- [2] Jan, S.-S. and Lin, Y.-C. A new multi-C/A code acquisition method for GPS. *GPS Solutions* **13**(4), 293–303 (2009).
- [3] Sosa, J., Montiel-Nelson, J. A., and Nooshabadi, S. Low power GPS pseudo random numbers using genetic algorithms. In *Evolutionary and Deterministic Methods for Design, Optimisation and Control with Applications to Industrial and Societal Problems, EUROGEN11* (, Capua, Italy, 2011).
- [4] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on* **6**(2), 182–197 Apr (2002).

Sosa, J., Alcaraz, D., Bautista, T., Montiel-Nelson, J. A., García-Alonso, S., Monzón-Verona, J. M., et al. (2013). Lite GPS pseudo-code generation using genetic algorithms. *Engineering Computations*, 30(8), 1170–1192.

JCR Impact Factor

JCR Year	COMPUTER SCIENCE, INTERDISCIPLINARY APPLI			ENGINEERING, MULTIDISCIPLINARY			MATH
	Rank	Quartile	JIF Percentile	Rank	Quartile	JIF Percentile	
2015	92/104	Q4	12.019	55/85	Q3	35.882	7
2014	42/102	Q2	59.314	21/85	Q1	75.882	;
2013	63/102	Q3	38.725	30/87	Q2	66.092	;
2012	59/100	Q3	41.500	24/90	Q2	73.889	;
2011	63/99	Q3	36.869	26/90	Q2	71.667	;
2010	67/97	Q3	31.443	36/87	Q2	59.195	;
2009	78/95	Q4	18.421	43/79	Q3	46.203	;
2008	85/94	Q4	10.106	48/67	Q3	29.104	;
2007	80/92	Q4	13.587	49/67	Q3	27.612	;
2006	54/87	Q3	38.506	25/67	Q2	63.433	;
2005	62/83	Q3	25.904	28/65	Q2	57.692	;
2004	69/83	Q4	17.470	NA	undefined		
2003	54/83	Q3	35.542	NA	undefined		
2002	37/80	Q2	54.375	NA	undefined		
2001	43/76	Q3	44.079	NA	undefined		
2000	45/75	Q3	40.667	NA	undefined		



EC
30,8

Lite GPS pseudo-code generation using genetic algorithms

Javier Sosa

*Microelectromechanic Systems Division, Institute for Applied Electronics,
University of Las Palmas de Gran Canaria,
Las Palmas de Gran Canaria, Spain*

Daniel Alcaraz Real-Arce, Tomás Bautista,
Juan A. Montiel-Nelson, S. Garcia-Alonso and
José M. Monzón-Verona

*Microelectromechanic Systems Division, Institute for Applied Microelectronics,
University of Las Palmas de Gran Canaria, Las Palmas de Gran Canaria,
Spain, and*

Saeid Nooshabadi

*Michigan Tech, Department of Electrical and Computer Engineering,
Houghton, Michigan, USA*

1170

Received 30 March 2012
Revised 11 August 2012
16 October 2012
Accepted 4 April 2013

Abstract

Purpose – In a global positioning system (GPS) receiver, one of the most time-consuming tasks is to identify and track the visible satellites. The paper aims to propose and examine in detail new and shorter identification patterns or lite pseudo-codes – pseudo-random numbers (PRNs) – that allow GPS receivers to reduce dramatically the computational effort to identify and track GPS satellites. Obtaining lite pseudo-codes is a multi-objective optimization problem that the paper resolves using genetic algorithms (GAs).

Design/methodology/approach – The lite PRNs are obtained by using NSGA-II and omni-optimizer multi-objective optimization techniques.

Findings – The new PRNs obtained with the proposed single/multi-objective solutions are always better than previously presented when the highest detection peak (DP) is required for the GPS receiver.

Originality/value – Results demonstrate that the problem of “obtaining lite pseudo-codes” is a multi-objective optimization problem. In other words, the solutions obtained with the single-objective approach could belong to a local minimum. The multi-objective approach provides a better solution than the single-objective approach in a 37 percent of the satellites while in other cases the multi-objective approach reaches the same DPs with a similar noise.

Keywords Multi-objective optimization, Genetic algorithms, GNSS, Gold Code, GPS, Low computational effort

Paper type Research paper



Engineering Computations:
International Journal for
Computer-Aided Engineering and
Software
Vol. 30 No. 8, 2013
pp. 1170-1192
© Emerald Group Publishing Limited
0264-4401
DOI 10.1108/EC-03-2012-0070

1. Introduction

In 1973 the US Department of Defense started the development of the global positioning system (GPS) (USAF. Navstar GPS, 2003), and in 1995 it was fully

This work has partially funded under projects BATTLEWISE (TEC2011-29148-C02-01) by the Spanish Ministry of Science and Technology and PURE-GNSS (ACIISI SolSubC200801000282) by the Canary Agency for Research, Innovation and Information Society.

functional. The main purpose of GPS applications is determining the location with a pre-established precision. Initially, the main goal of the specification and development of this global navigation satellite system (GNSS) was to become part of military equipment and to assist the location of troops. Currently its use in civil applications is widespread (Kaplan and Hegarty, 1996), and the number of civil GPS applications grows everyday.

The technology concerning GPS is nowadays so mature that it is even used in geodesy or cartography (Herring, 1999), where it is mandatory to perform very accurate measurements. On the other hand, emergent applications for studying the atmospheric conditions make use of some features derived from GPS technology (Flores *et al.*, 2001). An example is the specific application for identifying yellow dust in Asia for weather forecasts (Lin *et al.*, 2011), or for helping astronomers to evaluate the amount of image distortion due to the level of humidity in the atmosphere (Jerrett and Nash, 2001); and to use GPS signals as a precise clock source (Hsin-Min *et al.*, 2005; Seung-Woo *et al.*, 2011).

With respect to the original purpose of the GPS development, which is to obtain rather accurate information of the position of the application user, a big progress has been derived from understanding better the behaviour of the GPS signals (Klukas *et al.*, 2003). This has been mostly supported by a precise modelling of these signals and their multipath behaviour when affected by collisions (Larson *et al.*, 2010; Zavorotny *et al.*, 2010). This knowledge has been incorporated into current GPS applications to achieve a better accuracy (Chih-Hung *et al.*, 2011). These ultimate applications have experienced a big demand mainly due to the use of last generation mobile phones, with their high processing capabilities and performance (Duncan *et al.*, 2009). In these devices, the GPS is an additional application which has to operate independently, but competing in power consumption with the rest of applications (Toledo-Moreo *et al.*, 2007). Due to this, nowadays it is so important that the computation demanded by the GPS application consumes the least energy. The work presented in this paper tries to provide additional means of lowering the power consumed by the GPS application in the same way that others currently emerging in the field (Gunawardena and van Graas, 2006). A GPS receiver establishes its position using a three-step algorithm. First, the receiver identifies and tracks the visible satellites (Saab and Kassas, 2006). Then, the algorithm measures the delays among the satellite signals. Finally, the receiver computes its position when at least there are four GPS satellites identified and tracked. To determine the receiver position based on the measured delays from the GPS satellite signals is a well-known iterative process (Michalski and Czajewski, 2004). The delays between the signals of the GPS satellites are obtained with simple subtractions. In general, those two steps are implemented using software in an efficient way. In addition, to discover and to track the visible satellites are quite simple but expensive tasks in terms of required operations (Hamza *et al.*, 2009).

The visible satellites are tracked using their unique identification codes. Each code is a pseudo-random number (PRN) that maximizes the signal to noise ratio of a given satellite. A fixed set of PRNs is used on each receiver following the GPS standard (USAF. Navstar GPS, 2003). However, in this work we propose to use a new set of reduced PRNs to perform the satellite identification and tracking in a univocal manner but by sacrificing some signal to noise ratio. The new reduced identification numbers are obtained using single-objective and multi-objective genetic algorithms (GAs).

We use as GA optimization engines NSGA-II (Deb *et al.*, 2002), omni-optimizer (Deb and Tiwari, 2008) and GENESyS (Back, 1992). By reducing the noise to signal ratio in 11 dB, the obtained pseudo-codes allow to identify and track univocally the visible GPS satellites using only an 11.11 percent of the required computation effort – a reduction of an 88.89 percent, in comparison with the traditional methodology.

This paper is organized as follows. Section 2 introduces the identification and tracking procedures in GPS and evaluates the required computation effort using PRNs (identification codes). The idea of new PRNs (new sequences) to identify and track GPS satellites is introduced in Section 3. Optimization techniques for obtaining these new sequences are presented in Section 4. Both simple objective and multi-objective functions are described here in detail. Experimental results and comparisons are discussed in Section 5. Finally, Section 6 presents the conclusions of this work.

2. Satellite identification and tracking

The GPS constellation is composed by 24 active satellites from a set of 32. Each GPS satellite is identified with a unique identification code, named Gold Code (Prasad and Ruggieri, 2005). A Gold Code is a PRN of 1,023 bits, which is established by the GPS standard (USAF. Navstar GPS, 2003). Each satellite transmits continuously at 1,575.42 MHz (L1) its own PRN using a BPSK modulation with a bit rate of 1.023 MHz. This means that a PRN is 1 ms long. The main feature of the PRNs used in GPS is that the correlations among different PRNs are small and the auto-correlation for every PRN is big. Receivers use this feature to identify visible satellites unequivocally. Equation (1) shows the correlation function to detect a GPS satellite, where PRN^{sat} is the PRN for satellite number sat and D^{rx} is the received data:

$$Corr(PRN^{sat}, D^{rx}) = [a_1, a_2, \dots, a_n] \quad (1)$$

$$a_i = \sum_j^N PRN_{mod_N(i+j)}^{sat} + D_i^{rx}$$

N number of bits in PRN^{sat}

This equation requires the computation of $1,023 \times 1,023$ binary XOR and $1,023 \times 1,023$ addition operations. On the other hand, since the receiver does not know which satellites are the visible ones, it is mandatory to compare the received sequence with all possible PRNs. So, the total number of XOR and Addition operations must be multiplied by 32.

Furthermore, every GPS receiver must take into account the L1 carrier Doppler frequency. The received satellite frequency shift is caused by the movements of both the satellite and the receiver. In general, GPS receivers use 1.0-3.0 MHz as intermediate frequency (IF) and the Doppler effect shifts this IF a maximum of ± 10 kHz. In order to ensure that the GPS receiver does not lose any visible GPS satellite, the Doppler range that appears at the IF frequency is scanned using 500 Hz bins. Therefore, the GPS receiver must look for each Gold Code in 41 different frequencies.

In summary, the number of binary XOR and addition operations to determine the visible GPS satellites among all possible is:

$$size(ID^{sat}) \times size(D^{rx}) \times bins \times satellites = 1,373,046,048 \quad (2)$$

This number of operations requires non-practical CPU times for a commercial GPS receivers, and they are too many to implement this inside a pure software detector and tracker in a low-cost CPU like ARM7, MIPS or ATOM (Borre, 2006). Therefore, to speed up this computation GPS receivers usually implement these operations inside an *ad hoc* ASIC.

However, the hardware implementation of equation (1) is rather simple. A high-speed implementation requires a shift register of 1,023 bits, 1,023 XOR gates, 1,023 adders and 1,023 ten-bit registers as shown in Figure 1. With this circuit, after 1,023 clock cycles the outputs give as result the vector $Corr(PRN^{sat}, D^{rx})$.

The registers in the correlator (Figure 1) must contain the PRN code sequence. At GPS receiver setup time, this sequence is provided by a circuit called PRN code uploader. There are two basic and quite simple hardware implementations for the PRN code uploader. One is based on finite state machines (FSM) and the other one on memories. The FSM solution was proposed to override the initial ASIC limitations of early GPS receivers. The FSM-based uploader is designed to produce always the same predefined and known sequence (PRN code).

Since the beginning of the GPS in 1973 up to nowadays, electronic devices and memory technologies had increased their performance in terms of area, delay, and power consumption, which allowed a new solution based on memory. This new solution stores the PRN code in a First In, First Out (FIFO) memory, where each alignment index of the code is a position of the memory. This memory contains 1,023 available positions of one-bit word length (1,023 positions per 1 bit \leq 1 Kbit).

Nowadays, the FSM solution is less flexible than the memory based one. This is because, after finished the design stage of the GPS, the PRN code generated cannot be modified on the FSM solution. Using Flash memory, Electrically Erasable Programmable Read-Only Memory (EEPROM), among others, allows to change PRN codes with a simple ASIC firmware update of the GPS receiver.

It is well-known how to design a FSM circuit (Comer, 1994) to store a specific sequence of bits like PRN codes. Using one or another solution depends on the desire to make the GPS receiver updatable or not.

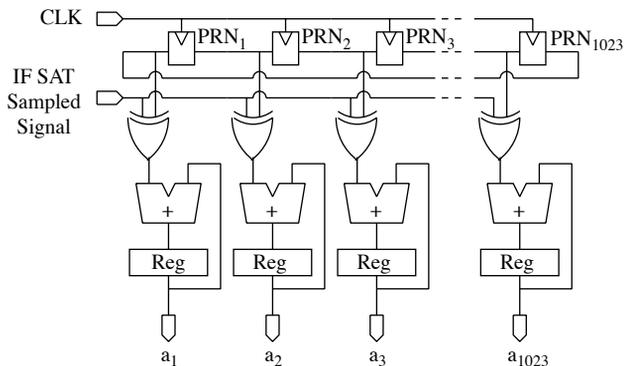


Figure 1.
Satellite correlator circuit

In addition, at working time and once the GPS receiver setup has finished, that is, the PRN codes are loaded into the correlators, those uploaders stop. The required effort, measured in terms of execution time and power consumption, of the PRN code uploaders in comparison with the PRN code correlators is negligible.

In this sense, we focus our research in reducing the number of required correlations to detect GPS satellites on current GPS receivers. In this article we propose a new pseudo-random sequence ($nPRN^{sat}$) that requires only one third of these 1,023 clock cycles, that is, 341 clock cycles.

3. New PRN

Our purpose in this work is to look for a new pseudo-random sequence ($nPRN$) that allows GPS receivers to detect the visible satellites using less operations compared to the traditional method (equation (1)). The main advantage of this idea is the reduction in the number of operations, also reducing the power consumed for detecting the visible satellites.

As mentioned before, the PRN length is 1,023 bits. This number can be factorized by 3, 11 and 31. We have chosen 341 (3 times this number is $3 \times 341 = 1,023$) as length of the $nPRN$ we are looking for. As a consequence, we reduce the speed with respect to the circuit shown in Figure 1 by a factor of three.

In this work, we assume that the GPS satellites do not change their own PRN. Our proposal allows reducing the working frequency of the correlators in GPS receivers. Therefore, the correlators of the satellite receiver will sample one bit of every three transmitted bits. The original PRN^{sat} sequence that a GPS satellite in particular transmits is:

$$PRN^{sat} = [a_1, a_2, \dots, a_{1023}] \quad (3)$$

In these conditions, our proposed GPS receiver can obtain only three possible subsequences of 341 bits just sampling one of every three:

$$PRN_A^{sat} = [a_1, a_4, \dots, a_{1021}], \quad (4)$$

$$PRN_B^{sat} = [a_2, a_5, \dots, a_{1022}], \quad (5)$$

and:

$$PRN_C^{sat} = [a_3, a_6, \dots, a_{1023}] \quad (6)$$

Our idea is to use GAs in order to find a $nPRN$ for each satellite:

$$nPRN^{sat} = [na_1, na_2, \dots, na_{341}] \quad (7)$$

where $nPRN^{sat}$ allows the GPS receiver to determine unequivocally the presence of the PRN^{sat} whatever the incoming subsequences are. In the next section, we discuss how to obtain a $nPRN^{sat}$ with a length of 341 bits.

4. Codification, evaluation and computational effort

We use as codification for individuals the same schema presented in Sosa *et al.* (2011). That is, we define the individual as an array of 341 Boolean variables. Basically, each individual represents a $nPRN$ proposal to be evaluated. Each binary element of the

individual array models a bit of the $nPRN$. In this codification, the array index is used to link each Boolean element of the individual to each bit of the $nPRN$.

At this point, we need to define the function alignment (α) between a subsequence ($nPRN^{sat}$) and the original code (PRN^{sat}) as:

$$\alpha(PRN_j^{sat}, nPRN^{sat}) \begin{cases} 1 \Leftrightarrow PRN_j^{sat} \text{ is aligned with } nPRN^{sat} \\ 0 \text{ other cases} \end{cases} \quad \forall j = A, B \text{ or } C \quad (8)$$

our interest is, for each PRN^{sat} , to obtain a $nPRN^{sat}$ with:

- (1) High correlation when:
 - it is compared with one of the three subsequences, and they are aligned.
- (2) Low correlation when:
 - it is compared with any of the three subsequences, and they are not aligned;
or
 - none of the three subsequences is present.

In other words, statement 1 says that it is mandatory to obtain a high value in the correlation when the received sequence is aligned with the $nPRN^{sat}$. In the existing literature, this value is named auto-correlation peak (detection peak (DP)). Statement 2 specifies that the magnitude of the $nPRN^{sat}$ correlation must be low when the incoming subsequence is not aligned or if the searched satellite is not present. Those cases are called in literature auto-correlation noise (AN) and cross-correlation noise (CN), respectively.

Therefore, in this scenario we can define five evaluation functions that assist to measure the quality of the proposed $nPRN$. We define five evaluation functions that will be shown later in the form of objective functions taking as basis the following ones. The first three functions are related to DPs (aligned auto-correlations):

$$Corr_A^{sat} = Corr(PRN_A^{sat}, nPRN^{sat}) \quad (9)$$

$$Corr_B^{sat} = Corr(PRN_B^{sat}, nPRN^{sat}) \quad (10)$$

$$Corr_C^{sat} = Corr(PRN_C^{sat}, nPRN^{sat}) \quad (11)$$

where PRN_A^{sat} , PRN_B^{sat} and PRN_C^{sat} are aligned with $nPRN^{sat}$. The next function is related to the AN:

$$AN^{sat} = Maximum(AN_A^{sat}, AN_B^{sat}, AN_C^{sat}), \quad (12)$$

and:

$$AN_A^{sat} = Corr(PRN_A^{sat}, nPRN^{sat}) \mid \alpha(PRN_A^{sat}, nPRN^{sat}) = 1$$

$$AN_B^{sat} = Corr(PRN_B^{sat}, nPRN^{sat}) \mid \alpha(PRN_B^{sat}, nPRN^{sat}) = 1$$

$$AN_C^{sat} = Corr(PRN_C^{sat}, nPRN^{sat}) \mid \alpha(PRN_C^{sat}, nPRN^{sat}) = 1$$

where PRN_A^{sat} , PRN_B^{sat} and PRN_C^{sat} are non-aligned with $nPRN^{sat}$. Finally, we evaluate the CN as follows:

$$CN^{sat} = \text{Maximum}(CN_A^{sat}, CN_B^{sat}, CN_C^{sat}), \quad (13)$$

and:

$$CN_A^{sat} = \text{Corr}(PRN_A^{sat}, nPRN^{sati}) \mid \alpha(PRN_A^{sat}, nPRN^{sati}) = 0$$

$$CN_B^{sat} = \text{Corr}(PRN_B^{sat}, nPRN^{sati}) \mid \alpha(PRN_B^{sat}, nPRN^{sati}) = 0$$

$$CN_C^{sat} = \text{Corr}(PRN_C^{sat}, nPRN^{sati}) \mid \alpha(PRN_C^{sat}, nPRN^{sati}) = 0$$

$$\forall sati \in [1, 32] \mid sati \neq sat$$

now we quantify the number of operations involved in each evaluation function, this is, how much computation effort the evaluation functions do require. For calculating the total computational effort we will use as figure of merit the computational effort unit, which reflects the computational effort to perform the correlation of 1,023 bits.

From the GPS receiver point of view, there are three potential incoming subsequences of 341 bits and by definition there is only a DP – one for each subsequence – when the incoming bits and data for comparison are aligned. Then the evaluation functions $Corr_A^{sat}$, $Corr_B^{sat}$ and $Corr_C^{sat}$ require a correlation of 341 bits each, that is, one third of the computation effort unit. Only one of the 341 possible alignments is the DP, and the other 340 non-aligned incoming subsequences produce AN^{sat} . In order to evaluate the maximum AN it is required to execute $340 \times 3 = 1,020$ correlations of 341 bits (340 correlation units).

In addition, when the incoming data do not correspond with the compared $nPRN^{sat}$, the result of the computation is CN. However, the cross-correlation must be computed taking into account all possible satellites. This implies that, for a given satellite, it is necessary to execute 3×341 correlations of 341 bits, and, therefore, these computations have to be also performed for the other 31 possible satellites. Hence, the total number of correlations that we need to execute is $341 \times 3 \times 31 = 31,713$ (10,571 correlation units).

In the following sections we enumerate several single/multi-objective functions that combine the five evaluation functions previously mentioned. For those objective functions, we also present their pros and cons.

4.1 Single objective function: solution 1

Based on Sosa *et al.* (2011), to obtain the $nPRN^{sat}$ for a satellite in particular, an objective function to guide a GA is an evaluation function that maximizes the sum of the three correlations between the $nPRN^{sat}$ and the subsequences $nPRN_A^{sat}$, $nPRN_B^{sat}$ and $nPRN_C^{sat}$. That is:

$$\text{Maximize} : Corr_A^{sat} + Corr_B^{sat} + Corr_C^{sat} \quad (14)$$

where $Corr_A^{sat}$, $Corr_B^{sat}$ and $Corr_C^{sat}$ (equations (9)-(11)) use the same individual proposed by the GA optimization engine as $nPRN^{sat}$.

The main advantage here produces, at the same time, its major weakness. The computation of the objective is focused in optimizing the three auto-correlation peaks. This procedure does not require computing the auto-correlation and the CNs, reducing the computational effort to a minimal value. However, this optimization does not take

the noises into account. Therefore, a potential risk exists of finding a $nPRN$ with high auto-correlation peak but with high noises, i.e. a local minimum.

4.2 Multi-objective function 1: solution 2a

Transforming the single-objective function presented in the previous paragraphs to a multi-objective one is quite simple. There exists an evident solution that consists in defining a multi-objective problem with three objectives functions to maximize, where each objective function is the auto-correlation between the proposed $nPRN^{sat}$ and each possible subsequence:

- Objective 1: to maximize the auto-correlation with subsequence A:

$$\text{Maximize : } Corr_A^{sat} \quad (15)$$

- Objective 2: to maximize the auto-correlation with subsequence B:

$$\text{Maximize : } Corr_B^{sat} \quad (16)$$

- Objective 3: to maximize the auto-correlation with subsequence C:

$$\text{Maximize : } Corr_C^{sat} \quad (17)$$

This solution keeps the same computational effort as the single-objective solution. However, it does not take in consideration the auto-correlation and the CNs.

4.3 Multi-objective function 2: solution 2b

As a natural evolution of the multi-objective solution 2a, in this case we propose a new multi-objective problem with five objectives, where the three first objectives are the same as in solution 2a and the other two are the auto-correlation and the CNs:

- Objective 1: to maximize the auto-correlation with subsequence A:

$$\text{Maximize : } Corr_A^{sat} \quad (18)$$

- Objective 2: to maximize the auto-correlation with subsequence B:

$$\text{Maximize : } Corr_B^{sat} \quad (19)$$

- Objective 3: to maximize the auto-correlation with subsequence C:

$$\text{Maximize : } Corr_C^{sat} \quad (20)$$

- Objective 4: to minimize the AN:

$$\text{Minimize : } AN^{sat} \quad (21)$$

- Objective 5: to minimize the cross-correlation noise:

$$\text{Minimize : } CN^{sat} \quad (22)$$

of course, this proposal includes the noise in the optimization method, but, on the other hand, this increases the required computational effort in a high degree.

4.4 Multi-objective function 3: solution 3

Founded on the usefulness of the results presented in Sosa *et al.* (2011), we propose a multi-objective solution based on three objectives where the first one is to maximize the sum of the three DPs and the other two are to minimize the auto-correlation and the CNs. That is:

- Objective 1: to maximize the sum of the aligned subsequences:

$$\text{Minimize : } \text{Corr}^{\text{sat}} \quad (23)$$

- Objective 2: to minimize the maximum AN:

$$\text{Minimize : } \text{AN}^{\text{sat}} \quad (24)$$

- Objective 3: to minimize the maximum CN:

$$\text{Minimize : } \text{CN}^{\text{sat}} \quad (25)$$

As mentioned in previous paragraphs, including the evaluation of the auto-correlation and CNs increases the computational effort in comparison with the single objective strategy and this effort is just a little bit lower than the previous proposed multi-objectives solutions.

4.5 Multi-objective function 4: solution 4

Finally, in order to evaluate the influence of the AN versus the CN in the multi-objective optimization in terms of maximum noise, we propose a multi-objective problem based on two objectives only. In this solution 4, one objective is to maximize the sum of the DPs and the other is to minimize the CN. That is:

- Objective 1: to maximize the sum of the aligned subsequences:

$$\text{Maximize : } \text{Corr}^{\text{sat}} \quad (26)$$

- Objective 2: to minimize the maximum CN:

$$\text{Maximize : } \text{CN}^{\text{sat}} \quad (27)$$

in this case, the computation effort is still high, since the high number of correlations is placed now in cross-correlations instead of the auto-correlations as previously. Anyway, this multi-objective solution 4 determines whether maximizing the DP produces a reduction of the AN.

5. Experimental results

The results presented in this paper were obtained in an Intel Core 2 Duo (P9600), using only one CPU core at 2.53 GHz with 4 Gbits of RAM, running Linux Debian v3.0.0-1. In order to support our GPS modelling, we developed a simple and fast combinational logic simulator on C language. Based on Sosa *et al.* (2011) we used GENE_sYs v1.0

(Back, 1992) as single-objective optimization engine, and NSGA-II v1.1.6 (Deb *et al.*, 2002) and Omni-Optimizer v1.0 (Deb and Tiwari, 2008) for single-objective and multi-objective optimization. Our combinational logic simulator and those three optimization engines were fully integrated in our Atenea tool (Sosa *et al.*, 2005).

5.1 Algorithms complexity and quality of the generated codes

Our research is focused on obtaining new and high quality PRN codes, that is, high DP, low CN and low AN pseudo-codes, as introduced in Section 4. We have generated the new PRN codes using several GA single and multi-objective optimization engines. The obtained PRN codes are used as a set of PRNs on each receiver for identifying and tracking each satellite; therefore, they are generated off-line and one time only. Since our specification requires high DPs and low noise pseudo-codes the used metric is oriented to the quality of the results against other approaches which emphasize in CPU time and required memory.

5.2 Single objective solution

Table I summarizes the GA setup in our experiments. In case of using GENESyS, we set a population of 50 individuals. The strategy for old generation replacement after generating the subpopulation (offspring) was the Steady State. The GA selection scheme selected was the Proportional one. We also used multi-point crossover and standard mutation. The stop criterion has been to finish when the optimization reached 2,000 generations. Given a satellite, the GA single optimization solution provides just a single lite pseudo-code (*nPRN*). The average execution time in all our experiments is around 3.5 seconds per satellite.

Table II presents the results of applying the methodology proposed in Sosa *et al.* (2011) to obtain the *nPRN*. In other words, we used GENESyS to minimize the single objective function $DP_A + DP_B + DP_C$, in order to generate the *nPRN*. The first, fourth, seventh and tenth columns of Table II present the satellite numbers. Then, columns second, fifth, eighth and 11th show the minimum DP of the signal when the proposed *nPRN* and the incoming data are aligned, that is, the minimum of DP_A , DP_B and DP_C .

<i>GENESyS</i>	
Population	50
Old generation replacement	Steady state
Selection scheme	Proportional
Multi-point cross-over	2
Mutation	Standard
Stop criterion	2,000 generations
Execution time ^a	3.5 seconds
<i>NSGA-II and omni-optimizer</i>	
Population	40
Crossover	0.8 [0.6, 1.0]
Mutation	0.0029 [0.001, 0.004]
Stop criterion	10,000 generations
Execution time ^b	475.2 seconds (NSGA-II) 455.6 seconds (omni-optimizer)

Notes: ^aAverage time to reach optimum; ^baverage CPU time to obtain complete Pareto front

Table I.
GAs setup parameters
and execution details

Finally, third, sixth, ninth and last columns give the maximum noise (N) for all possible cross-correlations and non-aligned auto-correlations, that is, the maximum between AN and CN .

As shown in Table II, the minimum DP is located in satellite 5 with 133 correlation units only. The maximum DP is placed in satellites 30 and 31 with 173 correlations units. In terms of noise, the maximum noise is with satellite 8 with 81 correlations units. It is mandatory to present the minimum value of the three DPs due to the worst case, in order to simplify the hardware of the reception unit of the GPS receiver. Following the convolution circuits there is a single comparator (threshold detector) to determine if there is a DP or not. If the receiver uses the $nPRN$ from GENE_sYs the detection threshold must be placed in the margin [13381]. So there exists a noise-to-signal margin of 52 correlation units.

5.3 Multi-objective solution

In this work we propose to modify the original single objective function presented in Sosa *et al.* (2011) and to apply other GA optimization engines and also multi-objective optimization to obtain a better $nPRN$. In this sense, we use the NSGA-II and omni-optimizer algorithms. The setup for both algorithms was a population of 40 individuals. The stop criterion was to reach 10,000 generations. Following the literature indications relating to both optimizers, we set the crossover probability to 0.8 and the mutation probability to $1/341 \cong 0.0029$. However, in order to obtain better solutions we altered those parameters. In case of the crossover probability we used the range [0.6, 1.0] with a step of 0.05 and the mutation probability in [0.001, 0.004] in steps of 0.0005. This tuning was done using a for loop to execute each variation (greedy strategy) as was shown in Table I. The multi-objective approaches provide the Pareto front of our problem instead of a single lite pseudo-code ($nPRN$). The average execution time in all of our experiments is around 475.2 seconds per satellite when NSGA-2 is the optimization engine and 455.6 seconds per satellite when omni-optimizer is used.

Tables III and IV provide the DPs and noises for all GPS satellites using the proposed objective functions for the NSGA-II and omni-optimizer algorithms, respectively. In both tables, the meaning of the columns is the same. The first column shows the satellite number. Then in couples, the DPs and the noises are shown for each proposed solution. The first pair, located at columns second and third, gives the data

Sat	DP	N									
1	161	77	9	153	69	17	153	63	25	169	51
2	157	69	10	157	65	18	165	63	26	169	67
3	161	57	11	157	63	19	161	59	27	153	71
4	157	81	12	145	71	20	157	67	28	161	65
5	133	73	13	161	69	21	173	63	29	165	67
6	157	61	14	169	57	22	153	63	30	173	59
7	157	63	15	169	59	23	157	61	31	173	63
8	145	81	16	149	53	24	153	61	32	161	73

Notes: DP – minimum detection peak, (minimum of DP_A , DP_B and DP_C); N – maximum noise, maximum between AN and CN

Table II.
Optimal $nPRN$ minimum detection peak (DP) when incoming subsequences are aligned and maximum noise (N) using GENE_sYs and a single-objective function

Sat ID	Sol ₁		Sol _{2a}		Sol _{2b}		Sol ₃		Sol ₄	
	<i>DP</i>	<i>N</i>	<i>DP</i>	<i>N</i>	<i>DP</i>	<i>N</i>	<i>DP</i>	<i>N</i>	<i>DP</i>	<i>N</i>
1	185	85	171	91	123	79	175	77	185	87
2	177	93	179	89	101	77	179	83	173	79
3	177	77	175	83	107	83	173	77	175	81
4	177	81	169	79	125	73	169	83	181	83
5	177	81	169	81	129	79	169	81	179	79
6	185	81	175	101	81	75	181	77	179	79
7	177	89	161	131	105	85	173	85	177	75
8	181	89	175	89	131	75	169	75	175	77
9	177	83	177	85	125	81	171	85	173	79
10	185	85	171	79	129	89	175	73	181	89
11	177	77	159	93	81	79	173	79	171	79
12	173	83	171	81	119	79	173	71	181	75
13	175	81	147	111	117	77	179	83	175	81
14	181	81	165	79	107	73	167	73	173	79
15	183	79	175	77	129	75	171	81	189	85
16	193	85	159	99	131	77	171	75	173	79
17	185	85	157	93	107	81	177	77	183	87
18	175	77	181	81	117	83	181	79	185	79
19	173	95	171	99	119	83	169	75	173	77
20	177	83	181	81	109	79	177	77	179	75
21	181	89	171	85	89	71	171	79	181	89
22	181	81	181	81	123	71	169	77	179	81
23	179	81	153	85	137	77	169	83	175	87
24	185	85	179	87	99	77	171	69	185	85
25	181	81	179	85	123	71	165	77	177	85
26	181	89	157	95	129	77	167	71	181	77
27	179	101	163	89	115	73	175	99	175	91
28	179	93	177	77	127	77	175	81	175	73
29	175	87	165	91	107	77	175	75	181	81
30	185	93	167	91	95	91	167	73	179	79
31	179	83	165	87	121	79	169	69	175	87
32	177	79	165	89	131	79	175	87	181	85

Notes: *DP* – minimum detection peak, (minimum of DP_A , DP_B and DP_C); *N* – maximum noise; maximum between *AN* and *CN*; Sol₁ – maximize $DP_A + DP_B + DP_C$; Sol_{2a} – maximize DP_A , DP_B , DP_C ; Sol_{2b} – maximize DP_A , DP_B , DP_C and minimize *AN*, *CN*; Sol₃ – maximize $DP_A + DP_B + DP_C$ and minimize *AN*, *CN*; Sol₄ – maximize $DP_A + DP_B + DP_C$ and minimize *CN*

Table III.
Optimal *nPRN* minimum
detection peaks (*DP*)
when incoming
subsequences are aligned
and maximum noise (*N*)
using NSGA-II for all
proposed objective
functions solutions

from the simulations using a single objective strategy solution no. 1. The fourth and fifth columns present the DPs and noises when using the proposed solution 2a. Columns sixth and seventh give the data from the GA optimization engine when solution 2b is used. Columns eighth and ninth exhibit the data from applying solution 3. Finally, the last two columns expose the data from solution 4.

As shown in Table III, when solution 1 is applied to NSGA-II the minimum DP and maximum noise are 173 and 101 correlation units, respectively. The noise is higher in comparison with the GENESyS *nPRN*. In terms of DP and signal-to-noise margin it implies greater values. That is, NSGA-II provides better solutions than GENESyS in terms of signal detection but also worse noise when it is used to optimize the single objective solution 1. When solution 2a is applied, the noise of the correlations increases

Sat ID	Sol ₁		Sol _{2a}		Sol _{2b}		Sol ₃		Sol ₄	
	DP	N	DP	N	DP	N	DP	N	DP	N
1	173	77	55	95	111	89	153	77	167	75
2	185	87	63	85	119	71	153	73	167	71
3	181	81	67	85	111	81	165	77	159	73
4	169	79	91	79	117	85	155	75	165	73
5	175	83	63	91	83	89	157	77	165	71
6	183	89	59	95	107	75	159	79	171	79
7	169	81	55	95	107	77	159	75	167	75
8	177	77	55	99	79	79	167	81	163	73
9	169	81	69	83	99	87	149	73	169	81
10	183	81	69	99	79	77	165	81	175	77
11	179	85	75	79	103	75	157	69	169	75
12	167	79	63	79	95	75	151	77	163	71
13	169	85	93	81	83	91	153	75	163	71
14	153	85	53	79	73	77	147	79	159	71
15	169	81	51	91	109	85	149	75	169	73
16	171	81	59	81	123	91	155	77	171	75
17	181	89	63	91	105	81	161	81	161	75
18	181	89	91	89	87	73	151	75	169	79
19	185	81	67	91	121	87	161	81	167	81
20	173	77	61	99	75	77	159	75	169	77
21	165	83	73	73	69	89	163	75	167	83
22	173	77	59	93	119	81	163	77	167	75
23	171	81	63	93	71	77	161	79	163	75
24	181	81	59	93	75	83	163	79	177	71
25	183	81	97	85	113	83	157	73	173	73
26	175	73	71	95	119	79	165	79	157	71
27	177	79	69	85	109	83	151	73	171	77
28	169	73	71	81	79	81	165	79	159	71
29	177	77	53	111	107	77	151	71	163	73
30	171	77	59	89	119	91	157	75	163	69
31	169	79	61	95	119	93	167	79	161	71
32	181	83	77	73	123	77	157	75	159	71

Table IV.
nPRN minimum detection peak (*DP*) and maximum noise (*N*) using omni-optimizer

Notes: *DP* – minimum detection peak, (minimum of DP_A , DP_B and DP_C); *N* – maximum noise; maximum between *AN* and *CN*; Sol₁ – maximize $DP_A + DP_B + DP_C$; Sol_{2a} – maximize DP_A , DP_B , DP_C ; Sol_{2b} – maximize DP_A , DP_B , DP_C and minimize *AN*, *CN*; Sol₃ – maximize $DP_A + DP_B + DP_C$ and minimize *AN*, *CN*; Sol₄ – maximize $DP_A + DP_B + DP_C$ and minimize *CN*

dramatically up to 131 correlation units (see satellite 7 at column fourth) and the minimum signal detection is lower (131 correlation units at satellite 13 and column fifth). So, the noise margin is only 16 correlation units. Solution 2b is not practical using a GPS threshold detector because there is a minimum DP of 81 (see satellite 6 at the sixth column) and the maximum noise is located in satellite 30 (at the seventh column) with 91 correlation units. That is, cases exist with more noise than the minimum DP for other satellites. Solution 3 offers better results than previous ones in terms of minimum DP and maximum noise. The minimum DP for solution 3 is 165 in satellite 25 (see the eighth column) and a noise of 99 correlation units. Therefore, the noise-to-signal margin is 66 correlation units. Finally, when solution 4 is applied to the NSGA-II

algorithm the minimum DP is 171 and the maximum noise is 91. These results increase the noise-to-signal margin up to a minimum of 80 correlation units.

When the proposed solution 1 is applied to omni-optimizer (see the second and third columns), the DP and the maximum noise are 153 in satellite 14 and 89 in satellites 6, 17 and 18, respectively. The signal-to-noise margin is 64 correlation units. When solution 2a is used, the minimum DP falls to 51 (see satellite 15 at the fourth column) while the maximum noise grows up to 111. Consequently, the results of this solution are not applicable in threshold detection. Solution 2b, in the same way as solution 2a, gives 71 as minimum DP and 93 as maximum noise (see satellites 23 and 31 at the sixth and seventh columns, respectively). Hence, it is impossible to detect GPS satellites using threshold detection. The minimum DP when solution 3 is applied is 147 (satellite 14 at the eighth column) and the maximum noise is 81 (see satellites 8, 10, 17 and 19 at the ninth column). In this case the noise-to-signal margin is 66 correlation units. Finally, when solution 4 is used the minimum DP is 157 in satellite 26 (see the tenth column) and the maximum noise is located in satellite 21 with a value of 83 correlation units (see the last column).

5.4 Results discussion

Table V summarizes the minimum DPs and noises obtained from the optimization algorithms when the proposed objective functions are used. Each row of the table represents results from the same objective function. The first column identifies the objective function under consideration. The three following columns give the minimum DP, maximum noise and the signal-to-noise margin for NSGA-II. Columns from fifth to seventh show those three parameters but they are generated by the omni-optimizer algorithm. Finally, the last three columns show the data obtained from GENESYs. Since GENESYs is a single objective optimization engine there are no available results for multi-objective solutions.

In general, results from Table V indicate that it is better to use a multi-objective algorithm to maximize the minimum DP with minimum noise (i.e. maximum signal-to-noise margin) in comparison with the single objective solution in terms of signal-to-noise margin. However, if the DPs are included as independent maximization objectives and to minimize both noises is also targeted, the growth of the number of objectives increases the search space and the probability of finding a local minimum. The best solutions are found when the multi-objective optimization engine manages a low number of independent objective functions (not more than three objectives). The

Objective	NSGA-II			Omni-optimizer			GENESYs		
	DP	N	SN	DP	N	SN	DP	N	SN
Sol ₁	173	101	62	153	89	64	133	81	52
Sol _{2a}	147	131	16	51	111	NA		MU	
Sol _{2b}	81	91	NA	71	91	NA		MU	
Sol ₃	165	99	66	147	81	66		MU	
Sol ₄	171	91	80	159	83	76		MU	

Notes: DP – minimum detection peak, (minimum of DP_A , DP_B and DP_C); N – maximum noise; maximum between AN and CN; SN – signal-to-noise margin; NA – solution not applicable; MU – multi-objective optimization unavailable

Table V.
*n*PRN minimum
detection peak (DP) and
maximum noise (N)
comparisons using
NSGA-II, omni-optimizer
and GENESYs

greatest DPs are obtained when NSGA-II uses solution no. 4. In this sense, it is remarkable that the AN is always lower than the CN in all studied cases. The reason is, since the maximum amount of correlation is fixed at 341, that the AN (non-correlated data) is minimized. It has to be remembered that correlation is a function that measures the degree of coincidence between the subsequences and the proposed *nPRNs*, and that the optimization engine tries to maximize the auto-correlation of the aligned sequences.

On the other hand, there is unexpected behaviour of the solutions 2b and 2a, where the problem is not solved in some cases. For the problem of finding the best *nPRN* using solutions 2a and b, the search space is seen as a cube with a 341-length edge, where each cube edge is parallel to a 3D coordinate system. Each axis represents one objective function of the multi-objective optimization. In other words, the axes display the three auto-correlations DP_A , DP_B and DP_C . Figure 2 shows this search space.

In theory, in this search space there is just a single maximum which is the vector $\langle 341, 341, 341 \rangle$. Due to the pseudo-random nature of the original *PRN*, it is reasonable that this maximum is not reachable. In other words, if we obtain a *nPRN* of 341-bit length with a DP of 341 for the three incoming subsequences, this implies that the three subsequences are the same. Therefore, the original *PRN* is not a *PRN*. As a consequence, it is impossible to obtain the vector $\langle 341, 341, 341 \rangle$. Therefore, the global problem of searching the best *nPRN* is turned into the search of the Pareto front of the best *nPRN*.

On the other hand, we are interested in the *nPRN* close to the cube diagonal that goes from the origin to the theoretic maximum (shows in Figure 2 with a dotted line) and also close to theoretic maximum. Those points represent the *nPRN* where the DP values are the best for the three incoming subsequences.

In Figure 2, we locate the optimal point offered by GENESYs near to the center of the search space (remarked with an * symbol). The progression of the search using solution 2a and NSGA-II and omni-optimizer are labelled with the + and x symbols,

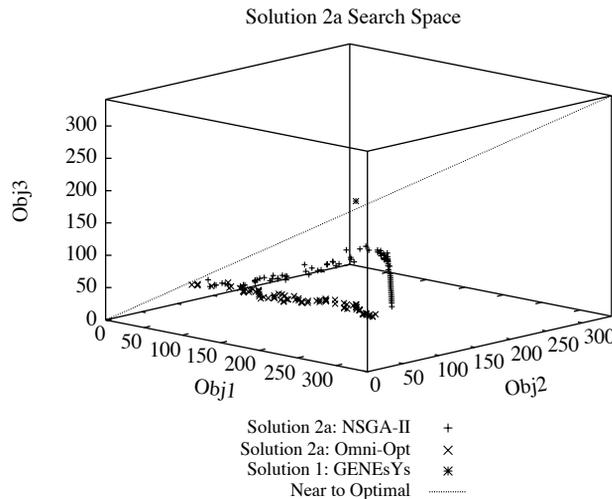


Figure 2.
Complete search
space (cube)

Note: Optimal *nPRN* for solution 2a using NSGA-II and OMNI-optimizer and solution 1 from GENESYs

respectively. Based on the arguments given above, we assume that the optimization engines must keep their solutions, in every generation, close to the diagonal of the cube and as near as possible to the vector solution $\langle 341, 341, 341 \rangle$. At the beginning both optimization engines move close the cube diagonal. However, as soon as the number of generations grows, both optimization engines diverge for several generations. Finally, both multi-objective algorithms approach to the same area. Unfortunately, the multi-objective algorithms fall into an area where there exist a local minimum and they cannot leave this area to find better solutions. As shown in Figure 2, the optimization increases substantially objective 1 keeping around the same value objectives 2 and 3 when the omni-optimizer is used. On the other hand, the NSGA-II algorithm starts increasing the value of the objective but reaches a point where it only optimizes objective 1 and reduces the values of objectives 2 and 3.

Figure 3 shows the Pareto front curve using NSGA-II and multi-objective function 1 that is Objective 1: $Maximize(Corr_A^{sat})$, Objective 2: $Maximize(Corr_B^{sat})$ and Objective 3: $Maximize(Corr_C^{sat})$ for the new PRN code of the satellite 0. Figure 3(a), (b) and (c) show the dependency among objectives using two dimensional plots. In all three figures, we include the cube diagonal (labeled as “near to optimal”). The most upper-right points of the graph and close to this diagonal belong to the Pareto front. This figure shows the typical Pareto front obtained in all of our experiments. From the comparison of Figure 3(a) and (b), we can observe that Objective 1 versus Objective 3 follows the same behavior as Objective 1 versus Objective 2 in terms of the Pareto

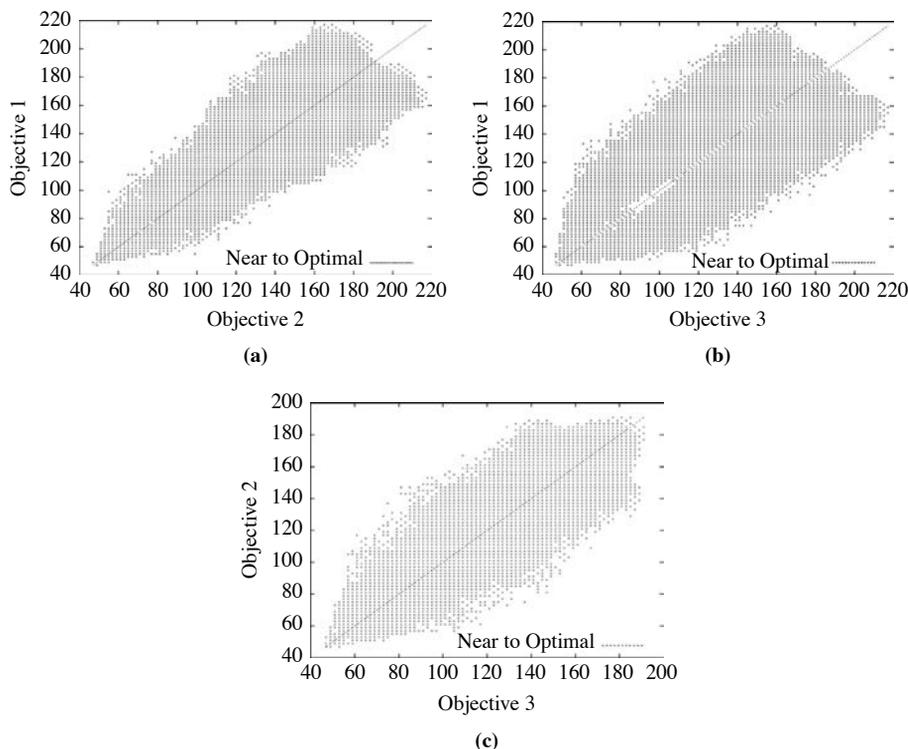


Figure 3.
Pareto front
representation for the
search of the new PRN
code for satellite 0 using
NSGA-II and
multiobjective function 1

front. However, Figure 3(c) shows a difference between previous comparisons and the Objective 2 versus Objective 3 graph. The GA optimization engines are trying to maximize the three DPs (DP_A , DP_B and DP_C). When the Pareto front appears like Figure 3(a) and (b) this means that there exist several sets of optimal solutions. On the other case shown in Figure 3(c), there exist only a few set of solutions that belong to the Pareto front. In terms of $nPRN$ codes, this behavior means that two of the DPs are too close and the third DP is not as near as the first ones.

Results from tested single/multi-objective functions and optimization engines indicate that it is possible to obtain sometimes an optimal $nPRN$ in terms of the DP but non-optimal in terms of the signal-to-noise margin. In addition, there is not any single algorithm in combination with a single/multi-objective function which provides the best $nPRN$. In this sense, Table VI presents the best results in terms of optimal DP and best signal-to-noise margin. The first column gives the satellite number. Then, from the second column to the sixth one the results for highest DPs are presented. In the same manner, the following columns show the results for the greatest signal-to-ratio margins. The minimum DP in this table is 177 correlation units (see satellites 7 and 9). The maximum noise is generated in satellite 27 with a value of 101 correlation units. So, the signal-to-noise margin is only 76 correlations when we select each $nPRN$ with the maximum DP criteria. Satellite 27 alone is responsible of this high value of maximum noise. So, in order to decrease the maximum noise we can pick another optimal $nPRN$ with lower DP, always greater or equal to 177, and also lower maximum noise. In this case, the omni-optimizer optimization engine based on solution 1 provides a DP of 177 and a maximum noise of 98. Certainly, this way the maximum noise is reduced, although the satellite 27 is still responsible of the maximum noise.

Moreover, column 6 in Table VI presents the objective used to reach the high quality $nPRN$. The multi-objective optimization solution produces better lite pseudo-codes in 12 satellites. That implies that the problem of “obtaining lite pseudo-codes” is a multi-objective optimization problem. Thus, it demonstrates that solutions obtained with the single-objective approach could belong to a local minimum.

In summary, there exists a trade-off between the selection of the DP and the signal-to-noise margin that allows, in several cases, to increase or decrease the maximum DP or signal-to-noise signal to obtain better global performance of all selected $nPRNs$. This scenario defines a new problem which is how to choose the best $nPRN$ set from the Pareto front curves of each independent $nPRN$. This new scenario is out of the scope of this work, which is focused in how to obtain the independent Pareto front of the satellite PRN.

Optimal $nPRN$ values presented in Table VI confirm that solutions 2a and b produce non-optimal solutions as described when obtained data were compared in terms of noise-to-signal and the number of required objectives. In addition, it is remarkable that NSGA-II generates the 87.5 percent of the selected $nPRN$ when the selection is done in terms of DP and a 60.71 percent of these $nPRNs$ were obtained with a single-objective function (solution 1). omni-optimizer $nPRNs$ are better than the NSGA-II ones in only 13.5 percent of the cases. In these cases, the omni-optimizer set of $nPRNs$ is not far of the NSGA-II solution.

The 68.75 percent of the best $nPRNs$ in terms of DP are also the best in terms of signal-to-noise margin. Finally, note that GENEsYs still provides the best solution for some cases (18.75 percent) when the signal-to-noise margin criterion is used. This

Sat	DP	N	Optimal DP			Optimal <i>signal-to-noise margin</i>				
			SN	Algorithm	Solution	DP	N	SN	Algorithm	Solution
1	185	85	100	NSGA-II	1					*
2	185	87	98	OMNI-opt	1					*
3	181	87	94	OMNI-opt	1	161	57	104	GENEsYs	1
4	181	83	98	NSGA-II	4					*
5	179	79	100	NSGA-II	4					*
6	185	81	104	NSGA-II	1					*
7	177	75	102	NSGA-II	4					*
8	181	89	92	NSGA-II	1	177	77	100	OMNI-opt	1
9	177	83	94	NSGA-II	1					*
10	185	85	100	NSGA-II	1	183	81	102	OMNI-opt	1
11	179	85	94	NSGA-II	1					*
				OMNI-opt						
12	181	75	106	NSGA-II	4					*
13	179	83	96	NSGA-II	3					*
14	181	81	100	NSGA-II	1					*
15	189	85	104	NSGA-II	4	169	59	110	GENEsYs	1
16	193	85	108	NSGA-II	1					*
17	185	85	100	NSGA-II	1					*
18	185	79	106	NSGA-II	4					*
19	185	81	104	OMNI-opt	1					*
20	179	75	104	NSGA-II	3					*
21	181	89	92	NSGA-II	1	173	63	110	GENEsYs	1
				NSGA-II	4					
22	181	81	100	NSGA-II	1					*
23	179	81	98	NSGA-II	1					*
24	185	85	100	NSGA-II	1	177	71	106	OMNI-opt	4
				NSGA-II	4					
25	183	81	102	OMNI-opt	1	169	51	118	GENEsYs	1
26	181	77	104	NSGA-II	4					*
27	179	101	78	NSGA-II	1	171	77	94	OMNI-opt	4
28	179	93	86	NSGA-II	1					*
29	181	81	100	NSGA-II	4					*
30	185	93	92	NSGA-II	1	173	59	114	GENEsYs	1
31	179	83	96	NSGA-II	1	173	63	110	GENEsYs	1
32	181	83	98	OMNI-opt	1					*

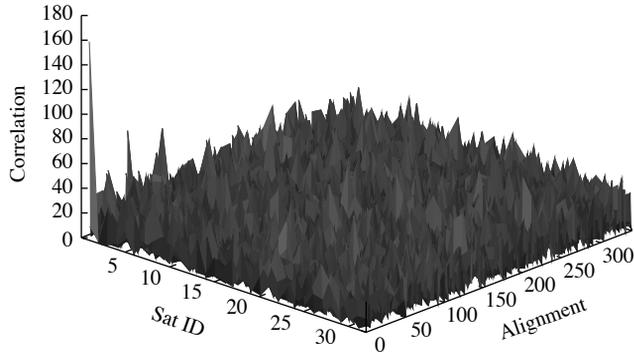
Table VI.
Best *nPRN* selection from
the three optimization
engines and solutions

Notes: DP – minimum detection peak; N – maximum noise; SN – signal-to-noise margin; * – optimal DP is also optimal SN; ** – OMNI-optimizer gives 177, 79 and 98 (DP, N and SN)

indicates that the other GA optimization engines fall into a local minimum in those cases.

Figure 4 shows the typical correlation executed in a GPS receiver when the searched satellite is present in the incoming signal. This curve shows all possible auto-correlations and cross-correlations of the *nPRN* provided by GENEsYs for satellite 1 with respect to the rest of satellites. In this figure, along the X and Y axes are displayed the satellite correlated and the alignment between the *nPRN* sequence and the incoming data. In these cases the DP is located at alignment 0 and satellite 1. The DP is 161 correlation units. The noise is represented by different hills and multiple values with a maximum value of 77 correlation units.

Figure 4.
Correlation between GPS
incoming data and proposed
nPRN for satellite 1



Demonstrated the existence of the reduced PRN that we called *nPRN*, we are able to evaluate the total required computational effort using proposed new and reduced pseudo random numbers by equation (2) to determine the visible satellites, where $size(ID^{sat})$ and $size(D^{rx})$ are 341:

$$size(ID^{sat}) \times Size(D^{rx}) \times bins \times satellites = 152,560,672 \quad (28)$$

this represents a 11.11 percent of the computation effort to determine the visible GPS satellites among all using traditional GPS methodology.

6. Conclusions

Nowadays, GPS applications are spreading in devices where it is desired the lowest power consumption as possible. In the case of GPS applications this power consumption is reduced by decreasing the computational effort. In this paper, we study in detail the problem of working with new and shorter PRNs for GPS receivers derived from the original well-known ones. These new PRNs are generated using NSGA-II and omni-optimizer with five different single/multi-objective solutions. Data from the simulation of our proposals are compared with results in the literature based on GENEsYs with a single-objective optimization. The main keys for the comparisons are the number of objectives to maximize/minimize in the single/multi-objective optimization engine, considering the different noise sources (auto- and cross-correlations) and their influence on the whole optimization process.

The new PRNs obtained with the proposed single/multi-objective solutions are better always than those in the literature when the highest DP is required in the GPS receiver. On the other hand, the proposed solutions based on our approach are a 81.25 percent better in comparison with others in the literature. In the cases where the proposed solution does not override the results in the literature, the deviation between those and our solution is lower than 5 correlation units. In average, our studied proposals increase the minimum DP from 133 to 173 for NSGA-II and 157 for omni-optimizer. On those cases, the maximum noise moves from 81 to 101 and 83, respectively.

Finally, we discuss about the selection criteria of the *nPRN* sequences and we propose a new problem that consists in how to choose the best new PRN based in a multi-criteria objective based in the Pareto front obtained. In addition, we evidence the

existence of this new problem giving an example that keeps the same DP maximum and decreasing the maximum noise from 101 to 98 correlation units just changing one $nPRN$ from the optimal in terms of maximum DP to a near but non-optimal $nPRN$.

References

- Back, T. (1992), *A Users Guide to Genesys 1.0*, Department of Computer Science, University of Dortmund, Systems Analysis Research Group, Dortmund.
- Borre, K. (2006), *A Software Defined GPS and Galileo Receiver: A Single-Frequency Approach*, Birkhuser, Boston, MA.
- Chih-Hung, W., Wei-Han, S. and Ya-Wei, H. (2011), "A study on GPS GDOP approximation using support-vector machines", *IEEE Transactions on Instrumentation and Measurement*, Vol. 60 No. 1, pp. 137-145.
- Comer, D.J. (1994), *Digital Logic and State Machine Design*, 3rd ed., Oxford University Press, Oxford, December 1.
- Deb, K. and Tiwari, S. (2008), "Omni-optimizer: a generic evolutionary algorithm for single and multi-objective optimization", *European Journal of Operational Research*, Vol. 185 No. 3, pp. 1062-1087.
- Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002), "A fast and elitist multi-objective genetic algorithm: NSGA-II", *IEEE Transactions on Evolutionary Computation*, Vol. 6 No. 2, pp. 182-197.
- Duncan, M.J., Badland, H.M. and Mummery, W.K. (2009), "Applying GPS to enhance understanding of transport-related physical activity", *Journal of Science and Medicine in Sport*, Vol. 12 No. 5, pp. 549-556.
- Flores, A., de Arellano, J.V.-G., Gradinarsky, L.P. and Rius, A. (2001), "Tomography of the lower troposphere using a small dense network of GPS receivers", *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 39 No. 2, pp. 439-447.
- Gunawardena, S. and van Graas, F. (2006), "Split-sum correlator simplifies range computations in GPS receiver", *Electronics Letters*, Vol. 42 No. 25, pp. 1469-1471.
- Hamza, G., Zekry, A. and Motawie, I. (2009), "Implementation of a complete GPS receiver using simulink", *IEEE Circuits and Systems Magazine*, Vol. 9 No. 4, pp. 43-51.
- Herring, T.A. (1999), "Geodetic applications of GPS", *Proceedings of the IEEE*, Vol. 87 No. 1, pp. 92-110.
- Hsin-Min, P., Chia-Shu, L. and Jeng-Kuang, H. (2005), "Performance testing of time comparison using GPS-smoothed P3 code and IGS ephemerides", *IEEE Transactions on Instrum and Measurement*, Vol. 54 No. 2, pp. 825-828.
- Jerrett, D. and Nash, J. (2001), "Potential uses of surface based GPS water vapour measurements for meteorological purposes", *Physics and Chemistry of the Earth International Journal, Part A: Solid Earth and Geodesy*, Vol. 26 Nos 6-8.
- Kaplan, E. and Hegarty, C. (1996), *Understanding GPS Principles and Applications*, Artech House Publishers, Norwood, MA.
- Klukas, R., Lachapelle, G., Ma, C. and Jee, G.-I. (2003), "GPS signal fading model for urban centres", *IEEE Proceedings on Microwaves, Antennas and Propagation*, Vol. 150 No. 4, pp. 245-252.
- Larson, K.M., Braun, J.J., Small, E.E., Zavorotny, V.U., Gutmann, E.D. and Bilich, A.L. (2010), "GPS multipath and its relation to near-surface soil moisture content", *IEEE Journal of Selected Topics in Appl. Earth Observations and Remote Sensing*, Vol. 3 No. 1, pp. 91-99.

- Lin, T.-H., Hsu, N.C., Tsay, S.-C. and Huang, S.-J. (2011), "Asian dust weather categorization with satellite and surface observations", *International Journal on Remote Sens*, Vol. 32.
- Michalski, A. and Czajewski, J. (2004), "The accuracy of the global positioning systems", *IEEE Instrumentation & Measurement Magazine*, Vol. 7 No. 1, pp. 56-60.
- Prasad, R. and Ruggieri, M. (2005), *Applied Satellite Navigation Using GPS, GALILEO, and Augmentation Systems*, Artech House Publishers, Norwood, MA.
- Saab, S. and Kassas, Z. (2006), "Power matching approach for GPS coverage extension", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 7 No. 2, pp. 156-166.
- Seung-Woo, L., Jeongrae, K., Myeong-Sook, J. and Young Jae, L. (2011), "Monitoring atomic clocks on board GNSS satellites", *Advances in Space Research*, Vol. 47 No. 10, pp. 1654-1663.
- Sosa, J., Montiel-Nelson, J.A. and Nooshabadi, S. (2011), "Low power GPS pseudo random numbers using genetic algorithms", paper presented at Evolutionary and Deterministic Methods for Design, Optimisation and Control with Applications to Industrial and Societal Problems, EUROGEN11, Capua, Italy.
- Sosa, J., Montiel-Nelson, J.A., Navarro, H. and García, J.C. (2005), "Minimum power consumption optimization using genetic algorithms", paper presented at Evolutionary and Deterministic Methods for Design, Optimisation and Control with Applications to Industrial and Societal Problems, EUROGEN05, Munich, Germany.
- Toledo-Moreo, R., Zamora-Izquierdo, M.A., Úbeda-Miarro, B. and Gómez-Skarmeta, A.F. (2007), "High-integrity IMM-EKF-based road vehicle navigation with low-cost GPS/SBAS/INS", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 8 No. 3, pp. 491-511.
- USAF. Navstar GPS (2003), "Where am I? Are we there yet?", *Air & Space Power Journal*, Vol. 6 No. 2, pp. 182-197.
- Zavorotny, V.U., Larson, K.M., Braun, J.J., Small, E.E., Gutmann, E.D. and Bilich, A.L. (2010), "A physical model for GPS multipath caused by land reflections: toward bare soil moisture retrievals", *IEEE Journal of Selected Topics in Appl. Earth Observations and Remote Sensing*, Vol. 3 No. 1, pp. 100-110.

About the authors

Javier Sosa received the MS degree in telecommunication engineering and the PhD degree from the University of Las Palmas de Gran Canaria, Gran Canaria, Spain, in 1999 and 2006, respectively. In 2000, he was working at Vitesse Semiconductor Corporation (Camarillo, USA) in the Division of Design and optimization of very high-speed digital communication systems. In 2009, he was a Visiting Professor at the Gwangju Institute of Science and Technology (Gwangju, South Korea) in VLSI multimedia signal processing in the Department of Information and Communications. From 2001 to 2008, he was a Lecturer in the Department of Electronic Engineering at the University of Las Palmas de Gran Canaria, where he is currently an Assistance Professor. Since 1999, he has been with the Institute for Applied Microelectronics (IUMA) working on CAD and VLSI design of emergent technologies, performance estimation and optimization of circuits and systems, and functional and formal verification. His current research fields include very high-speed integrated circuit (VHSIC) design, performance estimation and optimization of digital integrated circuits and systems; and verification of circuits and systems. Javier Sosa is the corresponding author and can be contacted at: jsosa@iuma.ulpgc.es

Daniel Alcaraz Real-Arce received the MS degree in electronics and telecommunication from the University of Las Palmas de Gran Canaria in 2008. Then he received the MS degree in renewable energies from the San Pablo CEU University and the MS degree in telecommunication

technologies from the Institute for Applied Microelectronics (IUMA) in 2010 and 2011, respectively. Currently, he is working toward the PhD degree in telecommunication technologies at the Microelectromechanical Systems division of the same institution. His research interests include GNSS systems, signal processing, microelectromechanical systems, and energy harvesting.

Tomás Bautista received the MS degree in telecommunications engineering in 1992 from the University of Las Palmas de Gran Canaria, Gran Canaria, Canary Is., and the PhD degree in 1999 from the same University. In 1989 he was a Visiting Scientist in the Fraunhofer Institut für Mikroelektronische Schaltungen und Systeme in Duisburg, Germany, and in 2000 in the Philips Research Laboratories in Eindhoven, The Netherlands. Since 2002, he is an Associate Professor in the Department of Electronic Engineering at the University of Las Palmas de Gran Canaria. Since the foundations he joined the Institute for Applied Microelectronics focusing his research in the development of methodologies for building RISC microprocessors. Later on, he also participated in the development of multimedia systems with concurrent reception and transmission, and smart camera systems for the mobiles and automotive segments. His current research interests are targeted mainly to platform based design, global navigation satellite systems and ADCs.

Juan A. Montiel-Nelson received the MS degree in electrical engineering and the PhD degree from the University of Las Palmas de Gran Canaria, Gran Canaria, Spain, in 1991 and 1994, respectively. From 1996 to 1997, he was a Visiting Scientist at the Centre for Very High-Speed Microelectronic Systems of Edith Cowan University, Western Australia, Australia, and in the Department of Computer and Communication Engineering of the same university. From 1997 to 2003, he was a Titular Professor in the Department of Electronic Engineering at the University of Las Palmas de Gran Canaria, where he is currently Full Professor. Since 1990, he has been with the Institute for Applied Microelectronics (IUMA) working on CAD and VLSI design of emergent technologies, performance estimation and optimization of circuits and systems, and functional and formal verification. He has authored or coauthored more than 50 papers in edited books, international journals, and conference proceedings. His current research fields include VHSIC design (circuit, logic, and module design; full-custom design and ASIC synthesis) in GaAs, SiGe, InP, and very deep submicrometer CMOS technologies; performance estimation and optimization of digital integrated circuits and systems (including power-delay tradeoff, timing, and interconnect analysis); and verification of circuits and systems (satisfiability, functional verification, and formal checking).

S. Garcia-Alonso, PhD in industrial engineering from the University of Navarra in San Sebastián, Spain. He has been working with industrial companies in their technical departments in different sectors from 1997 to 2001. Then, he joined LABEIN, a technological research and innovation centre, settled in Bilbao, Spain. Currently, since 2005, he is member of the Department of Electronic and Automatic Engineering at the University of Las Palmas de Gran Canaria, ULPGC, Spain, and focuses his research in MEMS, in the Micromechanical Systems Division of the Institute for Applied Microelectronics (IUMA).

José M. Monzón-Verona, PhD in industrial engineering from the University of University of Las Palmas de Gran Canaria, ULPGC, Spain. He is Titular Professor at the Industrial Engineering School and Professor of Circuit Theory in the Electric Engineering Department of the ULPGC. He is Member of the Micromechanical Systems Division of the Institute for Applied Microelectronics (IUMA) His current research fields include MEMS.

Saeid Nooshabadi received an MTech and a PhD in electrical engineering from the India Institute of Technology, Delhi, in 1986 and 1992, respectively. Saeid Nooshabadi has a joint appointment as a professor in high-performance computer architecture, embedded systems and VLSI signal processing in the Department of Electrical and Computer Engineering and the Department of Computer Science at Michigan Technological University. Prior to his current appointment, he was a Professor in VLSI multimedia signal processing in the Department of Information and Communications at the Gwangju Institute of Science and Technology, in the

EC
30,8

1192

Republic of Korea. In 1992, Nooshabadi was a Research Scientist at the CAD Laboratory, of the Indian Institute of Science in Bangalore, working on the design of VLSI chips for TV ghost cancellation in digital TV. In 1996 and 1997, he was a visiting faculty member and researcher at the Center for Very High Speed Microelectronic Systems, of Edith Cowan University in Western Australia, working on high-performance integrated circuits; and at Curtin University of Technology, in Western Australia, working on the design of high-speed high-frequency modems. From 2000 to 2007, he was with the School of Electrical Engineering and Telecommunications at the University of New South Wales, in Sydney, where he currently holds an adjunct appointment.

Sosa, J., Bautista, T., Alcaraz, D., García-Alonso, S., & Montiel-Nelson, J. A. (2015). Generation of New Detection Codes for GPS Satellites Using NSGA-II. In D. Greiner, B. Galván, J. Périaux, N. Gauger, K. Giannakoglou, & G. Winter (Eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences* (pp. 511–520). *Computational Methods in Applied Sciences*, 36. Springer International Publishing. ISBN: 978-3-319-11541-2.

Chapter 34

Generation of New Detection Codes for GPS Satellites Using NSGA-II

J. Sosa, Tomás Bautista, Daniel Alcaraz, S. García-Alonso and Juan A. Montiel-Nelson

Abstract In this paper we obtain new detection codes, to determine whether a GPS satellite in particular is visible, using NSGA-II as multi-objective optimization engine. Our approach takes into consideration the length of the code and the sampling frequency in comparison with other approaches found in the literature that fix those design parameters. The obtained new detection codes produce an improvement of the 19% in terms of CPU execution time. Results demonstrate that both design parameters must be taken in consideration to obtain high quality detection codes.

Keywords Genetic algorithms · GNSS · Gold codes · Low computational effort · Multi-objective optimization

34.1 Introduction

Nowadays, the detection of GPS signals for performing location tasks is one of the most commonly demanded applications [1]. In particular, the fast expansion of the mobile telephony, the increasing of the CPU capabilities and the reduced

J. Sosa (✉) · T. Bautista · D. Alcaraz · S. García-Alonso · J.A. Montiel-Nelson
Department of Electronic Engineering and Automation (DIEA),
Institute for Applied Microelectronics (IUMA),
University of Las Palmas de Gran Canaria (ULPGC),
35015 Las Palmas de Gran Canaria, Spain
e-mail: jsosa@iuma.ulpgc.es

T. Bautista
e-mail: bautista@iuma.ulpgc.es

D. Alcaraz
e-mail: dalcaraz@iuma.ulpgc.es

S. García-Alonso
e-mail: sgarcia@iuma.ulpgc.es

J.A. Montiel-Nelson
e-mail: montiel@iuma.ulpgc.es

battery charging times provide a wide market where GPS applications are potentially explored [2, 3]. However, in this scenario, since the GPS receiver just becomes one application running concurrently with some others in a single device, the requirements of low computational effort and low power consumption are mandatory [4]. Reducing the computational effort to determine whether a GPS satellite is visible or not is a hot topic in this research area [5–7]. In the existing literature three basic approaches are presented to cope with this problem. The first one consists in reusing most of the computation with additional hardware. This approach is called *split-sum methodology* [8]. Other authors propose to obtain a single detection code that allows to know if more than one satellite is visible or not [9]. Finally, in another approach [10] authors present a methodology to obtain GPS detection codes of 341 bits achieving a lower computational effort. In this paper we explore the idea of obtaining reduced length detection codes for GPS satellites presented in [10] using a multi-objective approach and we introduce as new optimization variables the length of the reduced code and the sampling frequency.

34.2 Problem Definition

In order to detect whether a satellite is visible or not, receivers compare the incoming GPS identifiers (at a frequency L1 of 1,575.42 MHz), with all the possible GPS satellite identifiers [11]. A satellite identifier is a Pseudo Random Number (PRN). Basically, a PRN is an array of binary digits where each digit is called *chip*. The length of the PRN array is 1,023 chips [12]. Each satellite has assigned a unique PRN as identifier. Every satellite transmits its own PRN identifier every millisecond. The comparison function is as follows:

$$\begin{aligned} \text{Comp}(\text{PRN}^{\text{sat}}, D^{\text{rx}}) &= [a_1, a_2, \dots, a_L] & (34.1) \\ a_i &= \sum_{j=1}^L \text{PRN}_{\text{mod}_L(i+j)}^{\text{sat}} * D_i^{\text{rx}} \end{aligned}$$

where L is the length of the PRN, that is, 1,023 chips for each GPS satellite. D^{rx} is the incoming radio frequency data that is acquired by the GPS analog front-end receiver. PRN^{sat} is the PRN identification of the GPS satellite (*sat*). This function takes in consideration all the possible alignments between the incoming GPS identification data and the compared PRN. Therefore, this is the reason to obtain an array of values and not only a single value. The size of array $\text{Comp}(\text{PRN}^{\text{sat}}, D^{\text{rx}})$ is L , one value for each possible alignment between the incoming data and the compared PRN.

Following the theory, when a satellite is visible the array $\text{Comp}(\text{PRN}^{\text{sat}}, D^{\text{rx}})$ has a unique maximum. This maximum value is called Detection Peak (D_P). The location/index where the D_P is placed in the comparison array is called code-phase. The code-phase determines the starting chip of the PRN sequence. The other values

in this array are lower than the D_P value. These lower values are called Noise (N). If the compared satellite (PRN^{sat}) is not visible, all values in array $Comp(PRN^{sat}, D^{rx})$ are noise.

Nowadays, in the literature there exist multiple approaches to implement the detection Eq. 34.1. For instance, it is quite easy to translate this Eq. 34.1 from the time domain to the frequency domain and use the Fast Fourier Transform (FFT) and its inverse (IFFT) to obtain the same results. However, in practical GPS receivers there exist only two basic detection techniques [13]. Their main key in comparison with other approaches are their implementation simplicity and the required computational effort.

In one of these approaches the L1 incoming signal is oversampled, that is, every chip of the PRN is sampled more than once, so the D_P grows with the increasing sampling frequency. In terms of Eq. 34.1, implementing the oversampling only requires to set the correct value to L .

The other solution, instead of increasing the sampling frequency, increases the sampling period. As a consequence, the recorded incoming GPS data contains more than one complete PRN sequence. Therefore, the Detection Peak increases its value in proportion to the increment of the sampling period, that is, the signal recording time is increased.

Stepping up the sampling frequency and/or the sampling period increases the sensibility of a GPS receiver when the GPS signal-to-noise ratio is too low. However, ordinary applications like open-sky navigation systems, i.e., typical GPS receiver for car tracking, only takes one of both methodologies with reduced increasing factors.

In this research, we propose to obtain new detection codes to determine whether a satellite is visible or not. The main feature of those new codes is their reduced length in comparison with the original PRN. Proposed new codes require lower computational effort than the traditional PRN [15]. Moreover, we are based on the approach presented in [10] where authors introduce a novel methodology to obtain new PRN detection codes using GA. In this previous approach, the research is focused in determining what kind of multi/single objective algorithm is more suitable for this type of application. However, in order to narrow the search space of the problem, authors fix the length of the new detection codes to a submultiple of the original length. This submultiple is 341 chips ($341 \times 3 = 1,023$).

Our proposal in this work is redefining the problem presented in [10] with two new variables to optimize. The first is the length of the detection code and the second is the sampling ratio. We define the sampling ratio as the size of the new detection code divided by 1,023 chips. Since we reduce the sampling ratio below the unity, we will use the term dropping ratio as a more adequate concept or definition. The approach presented in [10] has a length of 341 and a dropping ratio of 3 (that is, to take 1 sample and drop 2 for every 3 samples of the incoming signal). This results in a sampling ratio of 1/3 as shown in Fig. 34.1.

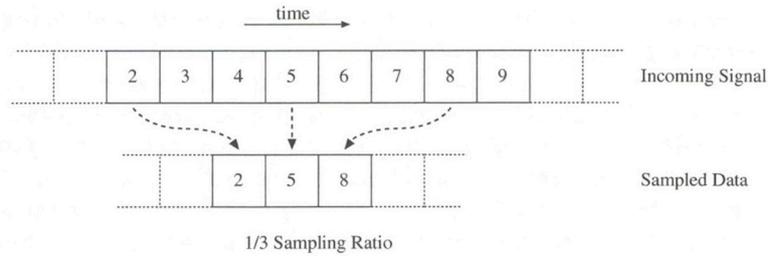


Fig. 34.1 Example of 1/3 sampling ratio

34.3 Problem Codification

Based on the approaches found in the literature, we choose the NSGA-II [14] as GA multi-objective optimization engine. The codification for the individual is as follows. Each individual represents a new detection code. The individual is made by an array of Boolean values (see Fig. 34.2). The status of each Boolean value can be only a logic zero and a logic one. The unknown or error states are forbidden in this engine. The length of the individual array determines the length of the new detection code. Finally, each element of the individual array corresponds with an element of the new detection code; this makes that the index is the same in both arrays.

Unsurprisingly, the cost function is basically the Eq. 34.1. As before mentioned, this cost function is an array of comparison values. However we are not interested in all these values but our attention is focused on the Detection Peak and the Noise. Moreover, we know that the dropping ratio determines the total number of Detection Peaks in the comparison array. For example, if the length of the new detection code is set to 341 and the dropping ratio is set to 3, then there exist 3 different Detection Peaks. So the comparison array contains as many Detection Peaks as set the dropping ratio. Other values on this comparison array are Noise values.

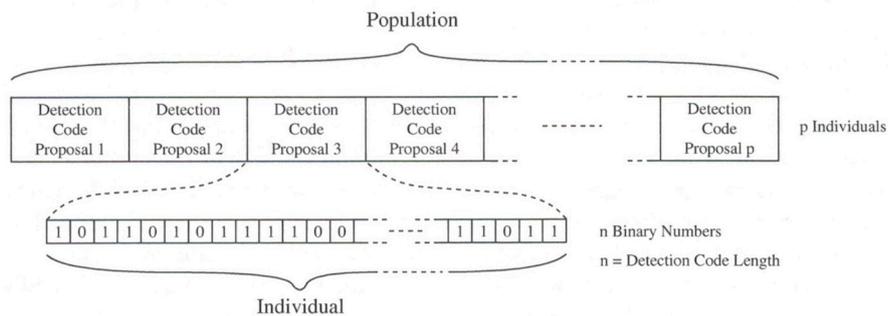


Fig. 34.2 Problem codification

We are interested in obtaining new reduced GPS detection codes with a low Noise and high Detection Peaks. In order to obtain those values, after evaluating each individual with Eq. 34.1, the evaluation function extracts the minimum Detection Peak and the maximum Noise from the computed comparison array.

34.4 Experiments

We assume that the search space of this problem grows with the length of the new detection codes and dropping ratio used. If we provide all the degrees of freedom on all our problem variables, the required computational effort is increased in great matter. The magnitude of this computational effort produces non-practical CPU execution times. In this sense, in order to obtain the solutions in a practical CPU time, we include the following rule:

$$|Proposed\ length \times Dropping\ Ratio - 1,023| < \xi \quad (34.2)$$

where ξ is the maximum allowed deviation between the original PRN and new proposed PRN.

Table 34.1 presents the NSGA-II optimization engine setup. We define two objective functions to optimize. The first one is to maximize D_P (minimize $-D_P$) and the second is to minimize the Noise (N). The NSGA-II optimization engine is controlled by a program written in C that proposes lengths and sizes randomly using the rule introduced in Eq. 34.2.

In other words, our application (C program) proposes different lengths and dropping ratios and the NSGA-II optimization engine obtains new detection codes. That is, the application starts with a proposal of a new length and dropping ratio to obtain a new detection code. Then the NSGA-II optimization engine takes the control and produces a Pareto-Front curve for those parameters. Once the optimization is finished, our application extracts the best solution of the Pareto-Front. This procedure starts over and over with several lengths and dropping ratios. In addition, after extracting the best solution our application evaluates the convergence of the proposed

Table 34.1 NSGA-II setup in our experiments

Parameter	Value
Num. objectives	2
OBJ_1	$-D_P$
OBJ_2	N
Population	40
Generations	10 k/100 k
Crossover	0.8
Mutation	1/PRN length
Seed	Random/uniform distribution

parameters. If the application determines that the solution can be refined/improved the NSGA-II is executed again with more generations (from 10k to 100k).

In order to measure the quality of new detection codes we introduce some concepts. We define the Detection Gap (D_G) as follows:

$$D_G = \min(D_P) - \max(N) \quad (34.3)$$

This means that the Detection Gap is the distance between the minimum Detection Peak ($\min(D_P)$) and the maximum Noise ($\max(N)$). Bigger Detection Gaps give better detection codes.

We also need to evaluate the required mathematical operations. For this purpose, we label the number of required multiplication/addition operations to execute Eq. (34.1) as CPU Operations (CPU_O). In this case, fewer CPU Operations require lower resources for hardware implementation.

Moreover, we define the CPU Effort (CPU_E) as the CPU Operations per Detection Gap, that is:

$$CPU_E = \frac{CPU_O}{D_G} = \frac{\text{num. required mul/add}}{\min(D_P) - \max(N)} \quad (34.4)$$

In our experiments we compare our proposal with the traditional methodology [15]. This traditional methodology consists of implementing Eq. (34.1) directly without any of our proposed improvements.

Table 34.2 presents some results of our application when we look for reduced codes of GPS satellite ID 1. In this experiment, we set ξ to a maximum of 400 chips. The first and second columns of the table contain the proposed code length and the dropping ratio. The number of generations are shown in the third column (in times of 1k generations). The fourth column gives the value of the rule defined in Eq. (34.2). The following two columns exhibit the best optimized minimum Detection Peak ($\min(D_P)$) and maximum Noise ($\max(N)$). The seventh column indicates the Detection Gap (D_G). The following column measures the CPU Operations (CPU_O). Ninth column evaluates the CPU Effort CPU_E . Finally, the last column presents the difference (CPU_{Diff}) between our proposal and the traditional methodology for the CPU Effort ($CPU_E(our) - CPU_E(traditional)$). In addition, please note that the last two rows in this Table 34.2 show the best solution obtained in [10] and also using the traditional methodology, respectively.

As expected, results from Table 34.2 are better as greater is the total number of generations in terms of maximum Noise and/or minimum Detection Peak. Last column presents a comparison between the traditional methodology and our new reduced detection code. The comparison is done in terms of required computational effort and detection gap to determine whether a satellite is visible or not. The negative values in this column denote better solutions than using the traditional methodology.

From Table 34.2, we observed that, for instance, the combination 320×3 (length \times dropping ratio) with 100k generations and 212×3 with 10k generations have similar values in last column, -73 and -76 , respectively. The first one has the double

Table 34.2 NSGA-II experiments for Satellite ID 1 and $\xi < 400$

Code length	Dropping ratio	Generations ($\times 1k$)	Rule value	$min(D_P)$	$max(N)$	D_G	CPU_O	CPU_E	CPU_{Diff}
384	2	10	768	206	64	142	147,456	1038.4	-59
448	2	10	896	236	66	170	200,704	1180.6	84
480	2	10	960	250	70	180	230,400	1280.0	183
512	2	10	1,024	276	72	204	262,144	1285.0	188
212	3	10	636	90	46	44	44,944	1021.5	-76
288	3	10	864	116	54	62	82,944	1337.8	241
304	3	10	912	126	56	70	92,416	1320.2	223
320	3	10	960	134	56	78	102,400	1312.8	216
336	3	10	1,008	140	60	80	112,896	1411.2	314
341	3	10	1,023	141	59	82	116,281	1418.1	321
256	4	10	1,024	90	52	38	65,536	1724.6	628
512	2	100	1,024	276	70	206	262,144	1272.5	176
288	3	100	864	136	54	82	82,944	1011.5	-85
304	3	100	912	148	56	92	92,416	1004.5	-92
320	3	100	960	156	56	100	102,400	1024.0	-73
341	3	100	1,023	177	57	120	116,281	969.0	-128
256	4	100	1,024	98	52	46	65,536	1424.7	328
341	3	40	1,023	177	89	88	116,281	1321.4	224 ^a
1,023	1	-	1,023	1,023	69	954	1,046,529	1097.0	0 ^b

$min(D_P)$ minimum Detection Peak, $max(N)$ maximum noise, D_G Detection Gap, CPU_O CPU Operations, CPU_E CPU Effort, CPU_{Diff} CPU Effort Difference

^aData from [10]

^bTraditional methodology [15]

of detection gap than the second one, but the second proposal requires a half of the CPU Operations to evaluate Eq. (34.1). Therefore, this comparison demonstrates that there exist several combinations of code lengths and dropping ratios that have similar ratios of CPU effort and detection gap.

The results obtained with our application, as shown on Table 34.2, are always better than the presented in a previous work [10] in terms of maximum Noise. In case of the Detection Peaks our approach obtains at least the same or better values than those in the referred work [10].

Table 34.3 presents the summary of the results for all checked lengths. The first column shows the PRN length. The second, the third and the fourth columns provide the maximum, minimum and average computational effort difference (CPU difference) between our proposal presented in this document and the traditional methodology. The last column in this table gives the obtained improvement in percentage. A negative value here means that our proposal is better than the traditional methodology in the case referred to.

Table 34.3 Obtained results from experiments for all satellites

PRN length	CPU _{Diff}			Improvement (%)
	Max	Min	Avg	
256	114.26	-159.25	-44.38	4.05
264	27.13	-339.43	-202.94	18.50
272	-40.09	-342.06	-208.95	19.05
280	-65.42	-328.37	-170.86	15.57
288	55.00	-267.56	-173.64	15.83
296	26.28	-270.43	-135.58	12.36
304	303.24	-241.30	-116.91	10.66
312	-38.91	-227.86	-115.25	10.51
320	40.78	-229.20	-98.23	8.95
328	47.51	-215.16	-80.58	7.35
341	114.26	-159.25	-44.38	4.05

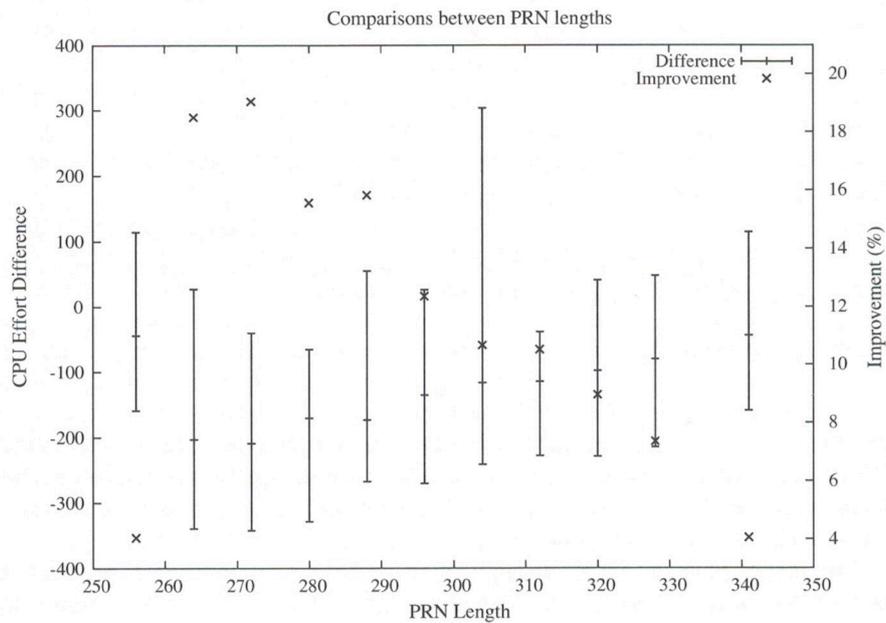


Fig. 34.3 Results comparison in terms of PRN length and CPU effort improvement

Figure 34.3 display graphically the data on Table 34.3. On the left vertical axis is represented the computational effort difference units. In addition, on the right vertical axis the average improvement in percentage is shown. The horizontal axis gives the detection code length (proposed PRN length). In this Fig. 34.3, each vertical bar represents the maximum, minimum and average obtained improvement (left vertical axis). Finally, the × symbol identifies the improvement in percentage (right vertical axis).

The ideal goal in our application is to obtain a set of detection codes where the obtained improvement is equal for all satellites. In addition, due to hardware restrictions, all detection codes must have with the same length. These are moved to Fig. 34.3 looking for a bar (set of detection codes with same length) and its maximum and minimum as close as possible to the bottom of the figure. In this sense, there are two solutions very close with similar results, lengths 264 and 272, where the improvement in average is 18.50 and 19.05 % respectively. The advantage of length 272 set is not only the better percentage, but also this set provides improvements in all its detection codes (maximum, minimum and average improvements are negative values).

34.5 Conclusions

In this paper a methodology is proposed to take into account new design parameters to obtain reduced GPS detection codes to determine whether a satellite is visible or not. Those new design parameters are the length of the reduced detection code and the dropping ratio. New optimal detection codes are obtained using NSGA-II as optimization engine. In addition, we present a new metric to evaluate the performance of new detection codes in terms of required CPU effort. Results demonstrate that new detection codes exist that exhibit both similar or better performance in terms of CPU effort, detection gap, code length and/or dropping ratio.

Acknowledgments This work is patent pending and was funded under project BATTLEWISE (TEC2011-29148-C02-01) of the Ministry of Economy and Competitiveness.

References

1. USAF Navstar GPS (2003) Where am I? Are we there yet? *Air Space Power J* 6(2):182–197
2. Kaplan E, Hegarty C (1996) *Understanding GPS principles and applications*. Artech House, Norwood
3. Duncan MJ, Badland HM, Mummery WK (2009) Applying GPS to enhance understanding of transport-related physical activity. *J Sci Med Sport* 12(5):549–556
4. Prasad R, Ruggieri M (2005) *Applied satellite navigation using GPS, GALILEO, and augmentation systems*. Artech House, Norwood
5. Borre K (2006) *A software defined gps and galileo receiver: a single-frequency approach*. Birkhuser, Boston
6. Hyoungmin S, Haeyoung J, Changdon K (2007) A new GNSS signal acquisition algorithm based on cross-correlation sequence with reduced signal-receiving time. In: *International conference on control, automation and systems (ICCAS'07)*, pp 2563–2567
7. Chih-Hung W, Wei-Han S (2011) A study on GPS GDOP approximation using support-vector machines. *IEEE Trans Instrum Meas* 60(1):137–145
8. Gunawardena S, van Graas F (2006) Split-sum correlator simplifies range computations in GPS receiver. *Electron Lett* 42(25):1469–1471
9. Jan S-S, Lin Y-C (2009) A new multi-C/A code acquisition method for GPS. *GPS Solutions* 13(4):293–303

10. Sosa J, Montiel-Nelson JA, Nooshabadi S (2011) Low power GPS pseudo random numbers using genetic algorithms. In: Evolutionary and deterministic methods for design, optimisation and control with applications to industrial and societal problems (EUROGEN11), Capua, Italy
11. Lee S-W, Kim J, Jeong M-S, Lee YJ (2011) Monitoring atomic clocks on board GNSS satellites. *Adv Space Res* 47(10):1654–1663
12. Michalski A, Czajewski J (2004) The accuracy of the global positioning systems. *IEEE Instrum Meas Mag* 7(1):56–60
13. Hamza G, Motawie I (2009) Implementation of a complete GPS receiver using Simulink. *IEEE Circuits Syst Mag* 9(4):43–51
14. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comp* 6:182–197
15. Dunn MJ (2012) Navstar GPS space segment/navigation user interfaces (IS-GPS-200). U.S. Coast Guard Navigation Center

Este trabajo fue desarrollado en el marco del proyecto SolSubC200801000282 subvencionado por la Agencia Canaria de Investigación, Innovación y Sociedad de la Información.

