



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

ESTUDIO Y SIMULACIÓN DE ALGORITMOS DE OPTIMIZACIÓN DE POSICIONAMIENTO AUTÓNOMO GNSS MULTICONSTELACIÓN

Autor: Fernando Anegón Ramos
Director: Eugenio Sillero Herrero

Madrid
Junio 2018

Fernando
Anegón
Ramos

ESTUDIO Y SIMULACIÓN DE ALGORITMOS DE OPTIMIZACIÓN DE POSICIONAMIENTO AUTÓNOMO GNSS MULTICONSTELACIÓN



AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESINAS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D. Fernando Anegón Ramos

DECLARA ser el titular de los derechos de propiedad intelectual de la obra: Estudio y simulación de algoritmos de optimización de posicionamiento autónomo GNSS multiconstelación, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

- a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar “marcas de agua” o cualquier otro sistema de seguridad o de protección.
- b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
- c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
- d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
- e) Asignar por defecto a estos trabajos una licencia Creative Commons.
- f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

- a) Que la Universidad identifique claramente su nombre como autor de la misma
- b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
- c) Solicitar la retirada de la obra del repositorio por causa justificada.
- d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

- a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
- b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
- c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e

intereses a causa de la cesión.

- d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 15 de Junio de 2018

ACEPTA

Fdo



Fernando Anegón Ramos
2018.06.15 12:53:17 +02'00'

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
“Estudio y simulación de algoritmos de optimización de posicionamiento
autónomo GNSS multiconstelación”
en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el
curso académico 2017/2018 es de mi autoría, original e inédito y
no ha sido presentado con anterioridad a otros efectos. El Proyecto no es
plagio de otro, ni total ni parcialmente y la información que ha sido tomada
de otros documentos está debidamente referenciada.



Fernando Aneón Ramos
2018.06.15 12:46:07
+02'00'

Fdo.: Fernando Aneón Ramos

Fecha: 15 / 06 / 2018

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO



Digitally signed by
Eugenio Sillero Herrero
Date: 2018.06.15
11:43:00 +02'00'

Fdo.: Eugenio Sillero Herrero

Fecha: 15 / 06 / 2018



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

ESTUDIO Y SIMULACIÓN DE ALGORITMOS DE OPTIMIZACIÓN DE POSICIONAMIENTO AUTÓNOMO GNSS MULTICONSTELACIÓN

Autor: Fernando Anegón Ramos
Director: Eugenio Sillero Herrero

Madrid
Junio 2018

Fernando
Anegón
Ramos

ESTUDIO Y SIMULACIÓN DE ALGORITMOS DE OPTIMIZACIÓN DE POSICIONAMIENTO AUTÓNOMO GNSS MULTICONSTELACIÓN



ESTUDIO Y SIMULACIÓN DE ALGORITMOS DE OPTIMIZACIÓN DE POSICIONAMIENTO AUTÓNOMO GNSS MULTICONSTELACIÓN

Autor: Anegón Ramos, Fernando.

Director: Sillero Herrero, Eugenio.

Entidad colaboradora: GMV Aerospace and Defence S.A.

RESUMEN DEL PROYECTO

I. Resumen

En este trabajo se han evaluado varias técnicas de optimización de los sistemas actuales de navegación por satélite, intentando aprovechar las ventajas que brindan los sistemas multiconstelación. Se han estudiado algoritmos de selección de satélites que sean capaces de escoger un conjunto de satélites que proporcionen una precisión adecuada y que no requieran de una carga computacional exageradamente elevada. Asimismo, se puede hacer uso de la redundancia de medidas disponible en un sistema de multiconstelación para llevar a cabo un proceso de detección y exclusión de medidas erróneas. Debido a que las situaciones en las que deben de operar los sistemas de navegación han cambiado, estos deben de ser capaces de hacer frente a la presencia de más de una medida errónea en el conjunto de datos.

Si una señal transmitida por un satélite se ve interrumpida por un obstáculo natural o artificial, se puede introducir un salto de ciclo en el lazo de seguimiento. Si no se detecta el salto de ciclo, el sistema le asigna un nuevo valor a la ambigüedad, lo que provoca una pérdida de precisión en la solución. En este trabajo se han estudiado diversas técnicas de detección de saltos de ciclo. Se han llevado a cabo varias simulaciones en el entorno matemático *Matlab* para poner a prueba los algoritmos y técnicas evaluadas. Las conclusiones extraídas de las simulaciones permiten seleccionar el enfoque preferente según las necesidades de la solución.

II. Introducción

A día de hoy, la navegación por satélite se encuentra presente en multitud de ámbitos y su uso es de vital importancia para numerosas industrias. Existen en la actualidad varios sistemas de navegación que proporcionan un servicio global: GPS, Galileo, GLONASS y BeiDou. Está demostrado que la utilización de un mayor número de satélites, y por tanto de más de un sistema de navegación, puede mejorar la navegación. Esta mejora se evidencia en la precisión que es capaz de proporcionar el sistema de navegación, debido a que contar con un mayor número de satélites implica una geometría más favorable. Además, el aumento en el número de satélites disponibles le concede al sistema mayor integridad en su funcionamiento, ya que se puede llevar a cabo la detección y exclusión de medidas erróneas con los satélites adicionales. Asimismo, la mayor disponibilidad de satélites le confiere al sistema una disponibilidad de servicio superior.

Sin embargo, el uso de un número elevado de satélites provoca un incremento en la carga computacional del sistema. Además, llegado a cierto número de satélites, la introducción de satélites adicionales no beneficia de forma significativa el desempeño del sistema de navegación. Por ello, parece necesaria la implantación de un método que sea capaz de seleccionar un subconjunto de satélites de entre los disponibles, y al mismo tiempo proporcione unos niveles de precisión adecuados. Por tanto, el primer objetivo de este proyecto será el análisis y la validación de distintos algoritmos de selección de satélites

con el fin de comparar su desempeño evaluando sus respectivas soluciones de navegación y los recursos computacionales que exigen.

Por otro lado, puede ser de interés estudiar las distintas técnicas de detección de medidas erróneas para aprovechar la abundancia de medidas proporcionadas por los satélites del sistema de navegación. El segundo objetivo de este trabajo consistirá en la revisión de varios métodos de detección de medidas erróneas. Además de la detección, varios de los algoritmos tratados en este proyecto también llevarán a cabo la exclusión de dichas medidas, lo que influirá de manera positiva en la solución de navegación. Para realizar una comprobación de la integridad que ofrecen los distintos métodos, se calcularán los niveles de protección de las técnicas que lo permitan. Dichos niveles establecen un radio dentro del cual aseguran contener la posición calculada.

Debido al uso cada vez más común de la navegación por satélite en entornos urbanos, los sistemas de posicionamiento requieren de cierta adaptación para lograr una operación adecuada en estos contextos. Las señales transmitidas en los sistemas de navegación se pueden ver interrumpidas debido a obstáculos naturales o artificiales (como pueden ser los edificios de una ciudad). Estas interrupciones pueden introducir saltos de ciclo en los lazos de seguimiento, lo que se traduce en cambios en el sesgo de la ambigüedad de fase estimada por el receptor. Cuando tiene lugar un salto de ciclo, el receptor pierde la cuenta del número de ciclo, asignándole un nuevo valor a la ambigüedad, lo que provoca una pérdida de precisión en la solución de navegación. Esta pérdida de precisión se propaga a lo largo del sistema de navegación. El tercer objetivo de este proyecto será el análisis de los distintos algoritmos de detección de saltos de ciclo. Si la detección de un salto se lleva a cabo con éxito, será posible descartar las medidas pertinentes con el fin de no perder precisión en el posicionamiento.

III. Metodología

Para cumplir con los objetivos mencionados, se llevaron a cabo simulaciones en el entorno matemático *Matlab* que comprendieron las situaciones a las que debe hacer frente un sistema de navegación en la realidad. Por tanto, todos los algoritmos y técnicas analizadas en este trabajo fueron programados en *Matlab*. Para las simulaciones correspondientes a los algoritmos de selección de satélites y de detección de medidas erróneas, se hizo uso de datos y medidas proporcionadas por un simulador de las constelaciones GPS y Galileo desarrollado por *GMV* y en el que también ha colaborado el proyectando. Este simulador permite emular la información recibida por un receptor de un sistema multiconstelación por parte de los satélites que comprende el sistema.

Se pusieron a prueba los métodos de selección de satélites y de detección de medidas erróneas con el receptor en tres disposiciones distintas: estático, siguiendo una trayectoria bidimensional y siguiendo una trayectoria tridimensional. Además, para validar el funcionamiento de las técnicas de detección actuales en un entorno multiconstelación, se introdujo más de una medida errónea en el conjunto de datos. Esto se debe a que la mayor parte de las técnicas actuales suponen la existencia de un único fallo, lo cual no se adecúa a todas las situaciones de uso de la navegación por satélite en la actualidad.

Esencialmente, para valorar los algoritmos de selección de satélites se compararon los valores del PDOP (Position Dilution Of Precision) como estimador de la precisión de la solución de navegación proporcionada y las distribuciones de los errores de posición logrados. Adicionalmente, se examinaron los tiempos requeridos por los algoritmos para llevar a cabo la selección como medida de la carga computacional.

Al igual que para la selección de satélites, se evaluaron las distribuciones de los errores de posición para estudiar la eficacia de la detección de medidas erróneas. Además, se contrastaron los niveles de protección obtenidos y se representaron los respectivos diagramas de Stanford.

Con respecto a los algoritmos de detección de saltos de ciclo, debido a que su funcionamiento no se ve modificado al operar en un sistema multiconstelación, se llevaron a cabo las simulaciones utilizando las medidas de un único satélite de la constelación GPS, usando observables provenientes de las distintas bandas de señal emitidas por el mismo. En este caso, se modificó el nivel de ruido térmico de las medidas para comprobar la inmunidad frente al ruido de los algoritmos. Además de la inmunidad frente al ruido, se evaluó el grado de detección de saltos de ciclo, es decir, la cantidad de saltos que las técnicas implementadas son capaces de detectar.

IV. Resultados

A. Selección de satélites

Se han evaluado un total de seis algoritmos: OSSA (Optimal Satellite Selection Algorithm), HEA (Highest Elevation Algorithm), MVA (Maximum Volume Algorithm), Dartboard DOP, WPDOP y EPR.

La simulación de los algoritmos de selección de satélites proporcionó información suficiente como para valorar y comparar el desempeño de los algoritmos en cuestión. Como era de esperar, el algoritmo OSSA ofreció los mejores resultados ya que para llevar a cabo la selección realiza una búsqueda exhaustiva en el espacio de observables para garantizar que se usa el conjunto que proporciona el mínimo global para el PDOP. Sin embargo, el tiempo requerido por el algoritmo para llevar a cabo la selección es extremadamente superior al del resto de técnicas, por lo que supone un método inviable para aplicaciones en tiempo real. Para llevar a cabo la selección de un subconjunto de 12 satélites de entre un conjunto de 24, este método requiere llevar a cabo 2704156 cálculos del PDOP, poniendo de manifiesto la elevada carga computacional necesitada.

Los algoritmos HEA, MVA y Dartboard DOP se posicionan como las técnicas que requieren de un menor tiempo para realizar la selección. Sin embargo, también son los que ofrecen menor precisión, especialmente el MVA ya que su inflexibilidad a la hora de seleccionar únicamente cuatro satélites le priva de los beneficios de los sistemas multiconstelación.

Por último, los algoritmos WPDOP y EPR no necesitan de un tiempo excesivo para llevar a cabo la selección del subconjunto y la precisión que ofrecen es considerablemente similar a la del OSSA. Aunque el WPDOP requiere de más tiempo para realizar la selección y proporciona una precisión ligeramente inferior que el EPR, cuenta con la ventaja de poder modificar el número de satélites seleccionados. Esta flexibilidad hace posible que el algoritmo mantenga un valor del PDOP estable y que sea capaz de adaptarse a distintos requerimientos de precisión.

Se debe mencionar que en la actualidad el algoritmo EPR no se encuentra publicado y su análisis se realiza por primera vez en este trabajo.

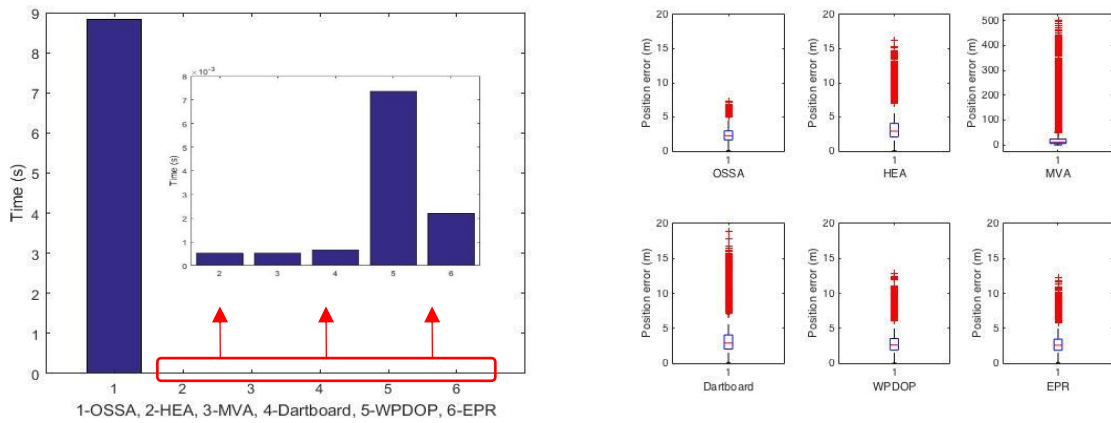


Figura 1: Tiempos medios requeridos para la selección de satélites (izquierda). Diagramas de caja del error de posición (derecha).

B. Detección de medidas erróneas

En este trabajo se han analizado 5 métodos: RAIM convencional, RAIM basado en separación de soluciones, estimador M de Huber, estimador M IGGII e IBPL (Integrity Based Protection Level).

El RAIM convencional ofrece una solución de navegación muy favorable independientemente del número de errores y del tamaño del error introducido. Esta técnica consigue detectar y excluir las medidas erróneas con éxito, lo que se ve reflejado en el error de posición, que cuenta con una distribución con una media y desviación típica menor que para el resto de técnicas evaluadas.

Por otro lado, el RAIM basado en separación de soluciones no logra detectar y excluir las medidas erróneas en una mayoría de casos. Este hecho provoca que los niveles de protección sufran decrementos puntuales, y causando que el sistema trabaje en la zona de operación engañosa del diagrama de Stanford. Además, al introducir un error con módulo creciente, el algoritmo comienza a excluir medidas de forma errónea hasta que no cuenta con un número suficiente como para calcular los niveles de protección.

Los estimadores-M evaluados en este trabajo ofrecen resultados dispares. Mientras que el estimador de Huber presenta un desempeño similar aunque algo inferior al RAIM convencional, el estimador IGGII no es capaz de hacer frente a la introducción de más de un error en las medidas y a un error cuyo módulo va aumentando.

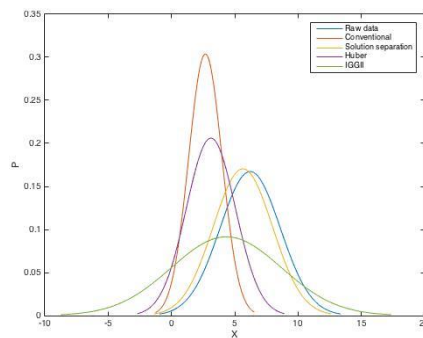


Figura 2: Funciones de densidad de probabilidad del error de posición para el caso estático con dos satélites con medidas erróneas.

Finalmente, el IBPL proporciona unos niveles de protección elevados que aseguran siempre la operación en la zona nominal del diagrama de Stanford. Sin embargo, este método no es apropiado para aplicaciones en las que se establece un límite de alerta (en el nivel de protección o en el error de posición). El IBPL se diferencia del resto de algoritmos tratados en este trabajo en que no lleva a cabo la detección y exclusión de medidas erróneas, limitándose a computar los niveles de protección, por lo que ofrece unas distribuciones del error de posición poco favorables.

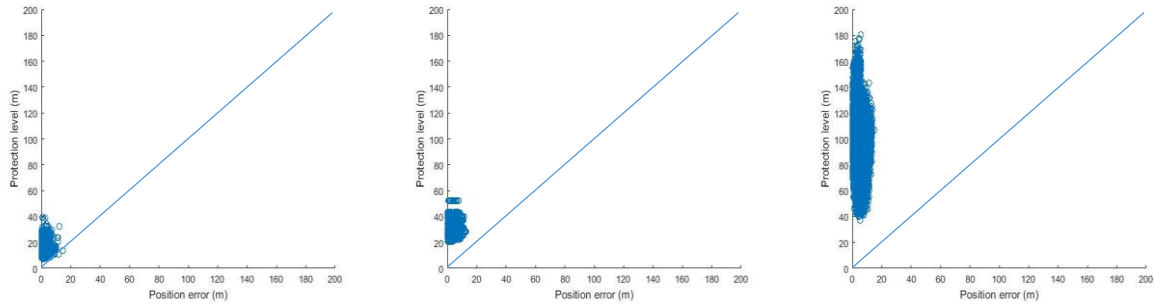


Figura 3: Componente horizontal de los diagramas de Stanford para el caso estático con dos satélites con medidas erróneas. De izquierda a derecha: RAIM convencional, RAIM basado en separación de soluciones e IBPL.

C. Detección de saltos de ciclo

Se han estudiado dos algoritmos de frecuencia simple: de ajuste polinómico y de media y desviación; y otros dos de frecuencia múltiple: combinación de geometría libre y combinación Melbourne-Wübbena.

Los dos algoritmos de detección de saltos de ciclo con frecuencia simple analizados requieren de modificaciones constantes del valor del umbral para hacer frente a distintos tamaños de saltos de ciclo, lo cual es totalmente inviable en una aplicación real ya que resulta imposible anticipar el tamaño del salto. Además, estos dos algoritmos se ven altamente perjudicados por la presencia de ruido.

El primer algoritmo de frecuencia múltiple (combinación de geometría libre) presenta un funcionamiento muy pobre cuando se lleva a cabo la diferenciación de las medidas ya que el ruido térmico también se ve aumentado. Debido al uso de frecuencia múltiple, el algoritmo es capaz de eliminar el efecto de la ionosfera sobre las medidas. El principal problema de este algoritmo reside en su incapacidad de discriminar variaciones rápidas en el retardo de ionosfera, lo que podría ser limitante en la práctica ya que provocaría falsas detecciones.

El algoritmo que utiliza la combinación de Melbourne-Wübbena utiliza frecuencia múltiple y proporciona los resultados más favorables de las técnicas analizadas en este trabajo. Presenta valores elevados tanto en el grado de detección como en la inmunidad frente al ruido. Además, es capaz de eliminar el efecto de la ionosfera y de discriminar variaciones rápidas de ionosfera de errores en las medidas. Por último, en la simulación con satélites con poca elevación (lo que provoca un ruido considerablemente superior que en los casos anteriores) el algoritmo sigue manteniendo niveles de detección de saltos y de inmunidad frente al ruido muy favorables.

V. Conclusiones

En este trabajo se han evaluado varias técnicas para optimizar la navegación por satélite y mejorar la integridad de los algoritmos de posicionamiento GNSS, intentando aprovechar las ventajas que proporcionan los sistemas multiconstelación. Se ha hecho un

estudio sistemático de tres técnicas que se centran en los problemas principales de la monitorización de la precisión de la solución. Los resultados del estudio han permitido extraer conclusiones que permiten seleccionar el enfoque preferente según las necesidades de la solución a implementar. Se ha demostrado que existen algoritmos de selección de satélites capaces de ofrecer una precisión similar a la obtenida mediante el método óptimo, pero que no requieren de tan elevada carga computacional. Uno de estos algoritmos se basa en la flexibilidad para modificar el número de satélites seleccionados. Mientras que el otro combina los aspectos más ventajosos de otros algoritmos.

Debido a la redundancia de medidas que ofrece un sistema multiconstelación, ha quedado patente que mediante técnicas de detección y exclusión de medidas se puede hacer frente a más de una medida errónea y asegurar la integridad del sistema sin comprometer el desempeño del mismo. Además, se ha demostrado que para el uso de estimadores robustos a este respecto, se debe tener en consideración la función del estimador a utilizar. Asimismo, se han expuesto las ventajas de los algoritmos de detección de saltos de ciclo de frecuencia múltiple frente a los de frecuencia simple. Estos algoritmos son capaces de lidiar con los efectos de la ionosfera, y en el caso del Melbourne-Wübbena, también de discriminar variaciones rápidas de ionosfera de errores en las medidas.

VI. Referencias

Gaglione, Salvatore, y otros. 2017. *Robust estimation methods applied to GPS in harsh environments*. s.l. : Navigation Conference (ENC), 14-25, 2017.

Roongpiboonsopit, Duangduen y Karimi, Hassan A. 2009. *A Multi-Constellations Satellite Selection Algorithm for Integrated Global Navigation Satellite Systems*. s.l. : Journal of Intelligent Transportation Systems: Technology, Planning, and Operations, 127-141, 2009.

Xu, Guochang. 2007. *GPS. Theory, Algorithms and Applications*. s.l. : Springer, 2007.

STUDY AND SIMULATION OF MULTICONSTELLATION GNSS AUTONOMOUS POSITIONING OPTIMIZATION ALGORITHMS

I. Abstract

In this project several techniques to optimize satellite navigation and to improve the integrity of positioning algorithms have been evaluated, aiming to take advantage of the benefits of multiconstellation systems. Algorithms that are able to select a group of satellites which provide an adequate accuracy and do not require an excessive computational load have been studied. Moreover, the redundancy of satellite measurements can be exploited in order to carry out the detection and exclusion of erroneous measures. Due to the environments in which navigation systems have to operate, these systems must be capable of facing multiple erroneous measures.

If a satellite transmitted signal was to be interrupted by a natural or artificial obstacle, a cycle slip may be introduced in the phase-lock loop. If not detected, the system assigns a new ambiguity value and the solution's accuracy is deteriorated. In this project, several techniques of cycle slip detection have been analyzed. To evaluate the performance of the algorithms and techniques, several simulations have been carried out in the mathematical environment *Matlab*. The conclusions extracted from such simulation allow to select the appropriate approach regarding the requirements of the solution.

II. Introduction

Nowadays, satellite navigation is present in multiple fields and its use is essential for numerous industries. Currently, four navigation systems provide global service: GPS, Galileo, GLONASS and BeiDou. It has been proved that the use of a greater number of satellites, and therefore of more than one navigation system, may improve navigation. This improvement is evinced in the precision the navigation system is capable of providing, as a greater number of available satellites implies a more favorable geometry. Additionally, the increase in the number of available satellites grants the system higher integrity in its operation, as the detection and exclusion of erroneous measurements can be carried out with the additional satellites. Moreover, the greater satellite availability grants a superior service availability to the system.

Nevertheless, the use of a greater number of satellites causes an increase in the computational load of the system. Furthermore, after a specific number of satellites, the introduction of additional ones does no benefit the performance of the navigation system in a significant way. The implementation of a method which is able to select a group of satellites from the available ones, and at the same time provides adequate precision levels seems necessary. Therefore, the first objective of this project is the analysis and validation of different satellite selection algorithms aiming to compare their performance through the evaluation of their respective navigation solutions and the computation load required.

Besides that, it seems interesting to study the different techniques of erroneous measurements detection to take advantage of the abundance of measurements provided by the satellites of the navigation system. The second objective of this project is to revise several methods of detection of erroneous measurements. Additionally, several of the algorithms considered carry out the exclusion of the erroneous measurements, which will influence the navigation solution in a positive manner. To conduct a validation of the integrity levels provided by each method, the protection levels will be calculated for the techniques which allow it. Protection levels establish a radius inside which the position error is assured to be contained.

Due to the ever more common use of satellite navigation in urban environments, positioning systems require adaptation to achieve an adequate operation in these contexts. The signals transmitted in the navigation systems may be interrupted by natural or artificial obstacles (such as city buildings). These interruptions can introduce cycle slips in the phase-locked loop, which means an estimated phase ambiguity bias variation. When a cycle slip takes place, the receiver loses the cycle count, and therefore assigning a new value to the ambiguity, which causes a loss of precision in the navigation solution. This precision loss propagates across the navigation system. The third objective of this project is the analysis of different cycle slip detection algorithms. If the detection of a cycle slip is successful, it would be possible to discard the pertinent measurements to avoid precision aggravation in the positioning.

III. Methodology

To achieve the aforementioned objectives, several simulations were carried out using the mathematical environment *Matlab* that comprehended the different situation a navigation system may face. Therefore, all algorithms and techniques analyzed in this project were programmed in *Matlab*. The data and measurements for the simulations of the satellite selection and erroneous measurements algorithms were provided by a GPS and Galileo simulator developed by *GMV*, in which the author of this paper collaborated. This simulator allows to mimic the information received by a receiver of multiconstellation system transmitted by the satellites of the system.

The satellite selection and erroneous measurements detection methods were tested with a receiver in three different aptitudes: static, following a two-dimensional trajectory and following a three-dimensional trajectory. Additionally, to validate the functioning of the current detection techniques, multiple erroneous measurements were introduced in the data combination. This responds to the supposition of the current techniques which were designed to detect an individual error, which does not correspond with all the satellite navigation environments faced nowadays.

Essentially, PDOP (Position Dilution Of Precision) values as an estimation of the navigation solution's precision and the position error's distributions were compared to assess the satellite selection algorithms. Additionally, the time required for each algorithm to carry out the selection were examined as a measure of computational load.

As for satellite selection, the position error's distributions were evaluated to study the efficacy of the erroneous measurement detection. Moreover, protection levels were compared and Stanford diagrams were represented.

Regarding the cycle clip detection algorithms, as their functioning is not influenced by their operation in a multiconstellation system, the simulations were carried out using measurements exclusively from one satellite of the GPS constellation, using observables from different frequency bands. In this case, the thermal noise level in the measurements was modified in order to test the resistance to noise of the algorithms. Besides to the resistance of noise, the degree of cycle slip detection was also evaluated, this is, the quantity of slips the techniques are capable of detecting.

IV. Results

A. Satellite selection

A total of six algorithms were evaluated: OSSA (Optimal Satellite Selection Algorithm), HEA (Highest Elevation Algorithm), MVA (Maximum Volume Algorithm), Dartboard DOP, WPDOP and EPR.

The simulation of such algorithms provided enough information to assess and compare their performance. As expected, the OSSA offered the better results as it carries out an exhaustive search of the selection which provides the global minimum of the PDOP. Notwithstanding, the time required by this algorithm to conduct the selection is greater by far than the rest of the techniques evaluated, which positions the OSSA as an unfeasible technique for real-time applications. To carry out the selection of a group of 12 satellites from a total of 24, this algorithm conducts 2704156 PDOP calculations, evincing the considerable computational load required.

The HEA, MVA and Dartboard DOP algorithms were proved to be the techniques which require of less time to carry out the selection. Nevertheless, they also offer the minor precision, especially the MVA as its inflexibility to select only four satellites deprives it from the benefits of multiconstellation systems.

Finally, the WPDOP and EPR algorithms do not require of excessive time to carry out the satellite selection and offer precision levels similar to that of the OSSA. Although the WPDOP requires of a greater time to conduct the selection and provides a slightly worse

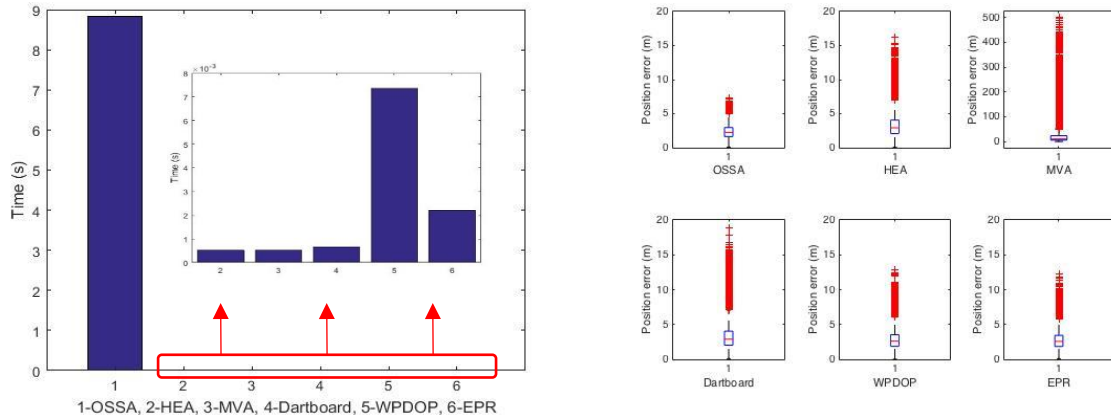


Figure 1: Mean time required for satellite selection (left). Box and whiskers plot for the position error (right).

precision than the EPR, it counts with the advantage of being able to modify the number of satellites selected. This flexibility allows the algorithm to maintain a stable PDOP value and to adapt to different precision requirements.

It must be noted that currently the EPR algorithm has not been published and its analysis has been carried out for the first time in this work.

B. Erroneous measurement detection

This Project has analyzed 5 methods: conventional RAIM, solution separation based RAIM, Huber's M-estimator, IGGII M-estimator and IBPL (Integrity Based Protection Level).

The conventional RAIM offers a navigation solution very favorable independently of the number of errors and the size of the error introduced. This technique is capable of detecting and excluding erroneous measurements successfully, this is reflected in the position error, which has a distribution with a mean value and standard deviation lower than the resto of the techniques evaluated.

Alternatively, the solution separation based RAIM is not able to detect and exclude erroneous measurements in most of the cases. This fact causes that the protection levels experience punctual decreases, and for the system to operate in the misleading area of the

Stanford diagram. Moreover, when introducing an error with an increasing absolute value, the algorithm excludes valid satellites to the point that it lacks enough satellites to calculate the protection level.

The M-estimators evaluated in this project offer dissimilar results. While Huber's estimator presents a similar performance to the conventional RAIM, the IGGII estimator is not capable of overcoming the introduction of multiple erroneous measurements and of one error with an increasing absolute value.

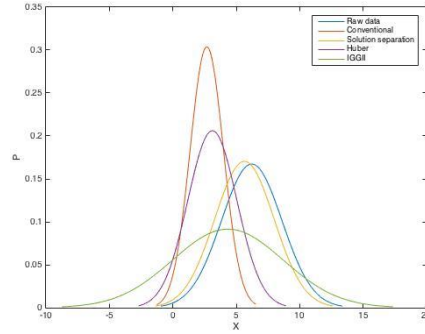


Figure 2: Probability density functions of the position error for the static case with two satellites with erroneous measurements.

Finally, the IBPL technique provides high protection levels which ensure the operation in the nominal zone of the Stanford diagram. Nevertheless, this method is not appropriate for applications which apply an alert limit (in the protection level or in the position error). The IBPL differs from the other algorithms evaluated in this project in the way that it does not carry out detection and exclusion of erroneous measurements, focusing on computing protection levels, and therefore offering unfavorable position error distributions.

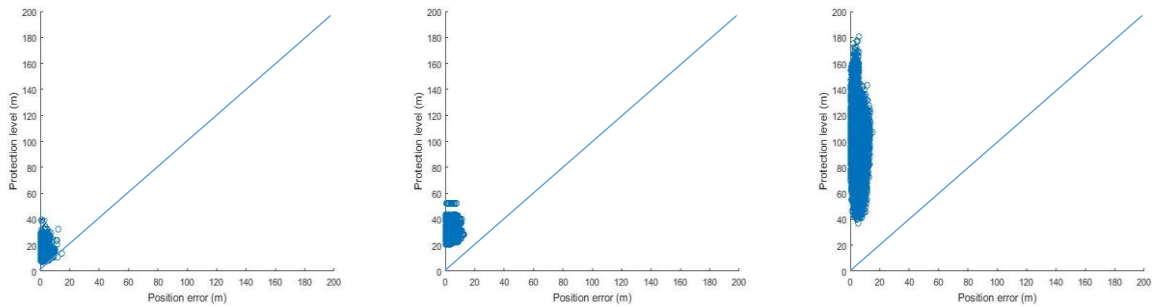


Figure 3: Stanford diagrams of the horizontal component for the static case with two satellites with erroneous measurements. From left to right: conventional RAIM, solutions separation based RAIM and IBPL.

C. Cycle slip detection

Two single frequency algorithms were studied: polynomial adjustment and mean and sigma; and other two multiple frequency algorithms: geometry-free combination and Melbourne-Wübbena combination.

Both single frequency cycle slip detection algorithms analyzed require constant modification of the threshold value to face different cycle slip sizes, which is completely

unfeasible in real applications as it is impossible to estimate the size of the slip. Additionally, these algorithms are highly influenced by the presence of noise.

The first multiple frequency algorithm (geometry-free combination) presents a poor performance when the differentiation of the measurements is carried out, as the thermal noise also increases. Due to the use of multiple frequencies, the algorithm is able to eliminate the ionosphere's effect on the measurements. The main problem of this algorithm is its incapability of discriminating quick variations in the ionospheric delay.

The algorithm based in the Melbourne-Wübbena combination uses multiple frequencies and provides the most favorable results of the techniques comprehended in this project. It presents high values of both detection degree and noise resistance. Additionally it is capable of eliminating the influence of the ionosphere and to discriminate quick variations of the ionosphere from errors in the measurements. Lastly, in the simulation using satellites with low elevation (which causes levels of noise to be considerably higher than in the previous cases) the algorithm continuous to offer high slip detection and noise resistance levels.

V. Conclusions

In this project several techniques to optimize satellite navigation and to improve the integrity of positioning algorithms have been evaluated, aiming to take advantage of the benefits of multiconstellation systems. A systematic study of three techniques which focus on the main problems of the monitoring of the solution's accuracy has been carried out. The results of such study allowed to extract conclusions which allow to select the appropriate approach regarding the requirements of the solution. It has been proved that several satellite selection algorithms are capable of offering a similar accuracy than the optimal method, but do not require of substantial computational load. One of these algorithms is based on the flexibility to modify the number of selected satellites. While the other combines the advantages of other algorithms.

Due to the measures redundancy provided by a multiconstellation system, it has become obvious that with measurement detection and exclusion techniques the system can overcome multiple erroneous measurements and ensure the integrity without compromising the performance of navigation. Additionally, it has been proved that when using robust estimators for this matters, the estimation's function must be highly considered. Furthermore, the benefits of multiple frequency cycle slip detection algorithms over single frequency ones have been exposed. These algorithms are capable to handle ionospheric effects, and in the case of the Melbourne-Wübbena combination, to discriminate quick ionospheric variations from measurement errors.

VI. References

- Gaglione, Salvatore, y otros. 2017.** *Robust estimation methods applied to GPS in harsh environments*. s.l. : Navigation Conference (ENC), 14-25, 2017.
- Roongpiboonsopit, Duangduen y Karimi, Hassan A. 2009.** *A Multi-Constellations Satellite Seleccion Algorithm for Integrated Global Navigation Satellite Systems*. s.l. : Journal of Intelligent Transportation Systems: Technology, Planning, and Operations, 127-141, 2009.
- Xu, Guochang. 2007.** *GPS. Theory, Algorithms and Applications*. s.l. : Springer, 2007.

Índice de la memoria

CAPÍTULO 1:	Introducción	3
1.1	Motivación	5
1.2	Objetivos	9
CAPÍTULO 2:	Navegación por satélite	11
2.1	Introducción	13
2.2	Segmentos	17
2.2.1	Segmento espacial	17
2.2.2	Segmento de control	17
2.2.3	Segmento de usuario	18
2.3	Observables	19
2.3.1	Pseudorange	19
2.3.2	Observable Doppler	21
2.3.3	Observable de fase	21
2.4	Errores e influencias físicas en las medidas GNSS	24
2.4.1	Errores en los satélites	25
2.4.2	Efecto de la ionosfera	25
2.4.3	Efecto de la troposfera	26
2.4.4	Error en el reloj del receptor	27
2.4.5	Ruido térmico	27
2.4.6	Multipath	27
2.5	Cálculo posición y velocidad satélites	28
2.6	Sistema de coordenadas	29
CAPÍTULO 3:	Detección de Saltos de Ciclo	32
3.1	Definición de Salto de Ciclo	34
3.2	Importancia de la Detección de los Saltos de Ciclo	36

3.3	Algoritmos de Detección de Saltos de Ciclo	37
3.3.1	Ajuste polinómico con frecuencia simple	37
3.3.2	Media y desviación con frecuencia simple	39
3.3.3	Combinación libre de geometría con frecuencia múltiple.....	40
3.3.4	Combinación Melbourne-Wübbena.....	43
3.4	Comparativa y Valoración de los Algoritmos.....	46
3.4.1	Obtención de datos.....	46
3.4.2	Ruido y errores de medida.....	48
3.4.3	Saltos de ciclo.....	49
3.4.4	Resultados	51
3.4.5	Comparativa y valoración	68
CAPÍTULO 4:	Algoritmos de selección de satélites	72
4.1	Sistemas multiconstelación	74
4.1.1	Aportación de los sistemas multiconstelación a la solución de navegación.....	74
4.1.2	Dilution Of Precision (DOP)	77
4.1.3	Necesidad de la selección de satélites	85
4.2	Descripción de los algoritmos de selección de satélites	89
4.2.1	Optimal Satellite Selection Algorithm (OSSA).....	90
4.2.2	Highest Elevation satellite selection Algorithm (HEA)	91
4.2.3	Maximum Volume Algorithm (MVA)	92
4.2.4	Dartboard DOP Algorithm	94
4.2.5	Controllable weighted PDOP threshold (WPDOP).....	97
4.2.6	EPR (Estimación Ponderada por Reducción)	100
4.3	Simulación y resultados	102
4.3.1	Caso estático	105
4.3.2	Caso 2-D.....	126

4.3.3	Caso 3-D.....	131
4.4	Comparativa y valoración.....	134
CAPÍTULO 5: Detección de medidas anómalas (RAIM).....		140
5.1	Introducción	142
5.2	RAIM (<i>Receiver Autonomous Integrity Monitoring</i>)	145
5.2.1	RAIM convencional	145
5.2.2	Niveles de protección.....	148
5.2.3	RAIM con separación de soluciones.....	154
5.2.4	RAIM IBPL (<i>Integrity-Based Protection Level</i>)	158
5.3	Estimadores robustos	161
5.3.1	Estimador-M de Huber	163
5.3.2	Estimador-M IGGII	165
5.4	Simulación y resultados	166
5.4.1	Caso estático	167
5.4.2	Caso 2D	202
5.4.3	Caso 3D	203
5.5	Comparativa y valoración.....	204
CAPÍTULO 6: Conclusiones		210
CAPÍTULO 7: Bibliografía.....		218
CAPÍTULO 8: Apéndices		226
8.1	Selección de satélites	227
8.1.1	Caso 2D	228
8.1.2	Caso 3D	233
8.2	Detección de medidas erróneas	238
8.2.1	Caso 2D	239
8.2.2	Caso 3D	261
8.3	Código fuente algoritmos detección saltos de ciclo.....	283

8.3.1	Single-frequency polynomial algorithm	283
8.3.2	Single-frequency mean-sigma algorithm.....	288
8.3.3	Multi-frequency geometry-free combination algorithm	293
8.3.4	Multi-frequency Melbourne-Wübbena combination algorithm ...	299
8.4	Código fuente algoritmos selección de satélites	305
8.4.1	Optimal Satellite Selection Algorithm (OSSA).....	305
8.4.2	Highest Elevation Algorithm (HEA)	308
8.4.3	Maximum Volume Algorithm (MVA).....	311
8.4.4	Dartboard DOP algorithm	315
8.4.5	WPDOP algorithm.....	320
8.4.6	EPR algorithm.....	326
8.5	Código fuente algoritmos detección de medidas erróneas	334
8.5.1	RAIM convencional	334
8.5.2	Nivel de protección RAIM convencional.....	338
8.5.3	RAIM basado en separación de soluciones	340
8.5.4	Nivel de protección RAIM basado en separación de soluciones 343	
8.5.5	Estimador de Huber	344
8.5.6	Estimador IGGII	347
8.5.7	Integrity Based Protection Level (IBPL)	350
8.6	Arquitectura del simulador GPS-Galileo.....	352

Índice de figuras

Figura 1: Esquema receptor GNSS. _____	18
Figura 2: Código PRN. _____	19
Figura 3: Medida tiempo de vuelo. _____	20
Figura 4: Variación de la frecuencia con el efecto Doppler. _____	21
Figura 5: Sistema de coordenadas ECEF. _____	29
Figura 6: Presencia de un salto de ciclo en el observable de diferencia de fase. _____	35
Figura 7: Combinación fase-rango para una sola frecuencia. Saltos de ciclo de cinco longitudes de onda. _____	38
Figura 8: Combinación libre de geometría. Saltos de ciclo de una longitud de onda. _____	40
Figura 9: Procedimiento de 4 diferencias. _____	41
Figura 10: Combinación libre de geometría diferenciando las medidas. Saltos de ciclo de una longitud de onda. _____	42
Figura 11: Combinación Melbourne-Wübbena. Saltos de ciclo de dos longitudes de onda. _____	44
Figura 12: Archivo RINEX GPS. _____	47
Figura 13: Detección saltos de ciclo para una única frecuencia utilizando ajuste polinómico. Saltos de ciclo de cinco longitudes de onda. _____	52
Figura 14: Detección saltos de ciclo para una única frecuencia utilizando media y desviación. Saltos de ciclo de cinco longitudes de onda. _____	55
Figura 15: Detección saltos de ciclo para dos frecuencias utilizando la combinación libre de geometría. Saltos de ciclo de dos longitudes de onda. Umbral aumentado para reducir la influencia del ruido. ____	59
Figura 16: detección de saltos de ciclo utilizando la combinación Melbourne-Wubben. Saltos de ciclo de dos longitudes de onda. _____	62
Figura 17: Valor del PDOP según el número de satélites utilizados. _____	86
Figura 18: Tiempo necesario para el cómputo de la solución de navegación según el número de satélites utilizados. _____	87
Figura 19: Diagrama polar. _____	94
Figura 20: Distribución 1 del espacio algoritmo Dartboard DOP. _____	95
Figura 21: Distribución 2 del espacio algoritmo Dartboard DOP. _____	95
Figura 22: Cantidad de satélites visibles para los algoritmos de selección de satélites. _____	103
Figura 23: Cantidad de satélites visibles según la constelación. _____	104
Figura 24: Valores del PDOP utilizando todos los satélites disponibles en cada momento. _____	105
Figura 25: Error de posición en el caso de utilizar todos los satélites disponibles en cada momento. ____	106
Figura 26: Función de densidad de probabilidad del error de posición para el caso de todos los satélites disponibles. _____	107

<i>Figura 27: Tiempo medio requerido por los algoritmos OSSA, HEA, MVA, Dartboard DOP, WPDOP y EPR para llevar a cabo la selección.</i>	108
<i>Figura 28: Cantidad de satélites visibles para un periodo de tiempo reducido para el caso del algoritmo OSSA.</i>	111
<i>Figura 29: Valores del PDOP para el algoritmo OSSA.</i>	112
<i>Figura 30: FDP del error de posición para el algoritmo OSSA.</i>	113
<i>Figura 31: Valores del PDOP para el algoritmo HEA.</i>	114
<i>Figura 32: FDP del error de posición para el algoritmo HEA.</i>	115
<i>Figura 33: Valores del PDOP para el algoritmo MVA.</i>	116
<i>Figura 34: FDP del error de posición para el algoritmo MVA.</i>	117
<i>Figura 35: Valores del PDOP para el algoritmo Dartboard DOP.</i>	118
<i>Figura 36: FDP del error de posición para el algoritmo Dartboard DOP.</i>	119
<i>Figura 37: Comparación entre los valores del PDOP y del error de posición para el algoritmo Dartboard DOP.</i>	119
<i>Figura 38: Valores del PDOP para el algoritmo Dartboard DOP modificado.</i>	120
<i>Figura 39: FDP del error de posición para el algoritmo Dartboard DOP modificado.</i>	120
<i>Figura 40: Valores del PDOP para el algoritmo WPDOP.</i>	122
<i>Figura 41: Valores del PDOP y número de satélites utilizados para el algoritmo WPDOP.</i>	123
<i>Figura 42: FDP del error de posición para el algoritmo WPDOP.</i>	123
<i>Figura 43: Valores del PDOP para el algoritmo EPR.</i>	124
<i>Figura 44: FDP del error de posición para el algoritmo EPR.</i>	125
<i>Figura 46: Solución utilizando todos los satélites para el caso de trayectoria bidimensional.</i>	126
<i>Figura 47: Valores del PDOP utilizando todos los satélites para el caso de trayectoria bidimensional.</i>	127
<i>Figura 48: Función de densidad de probabilidad del error de posición utilizando todos los satélites para el caso de trayectoria bidimensional.</i>	128
<i>Figura 49: Tiempo medio requerido por los algoritmos para llevar a cabo la selección para el caso de trayectoria bidimensional.</i>	129
<i>Figura 50: Solución utilizando todos los satélites para el caso de trayectoria tridimensional.</i>	131
<i>Figura 51: Detalle de la altitud (izquierda). Representación tridimensional de la solución (derecha).</i>	131
<i>Figura 51: Valores del PDOP utilizando todos los satélites disponibles para el caso de trayectoria tridimensional.</i>	132
<i>Figura 52: Tiempo medio requerido por los algoritmos para llevar a cabo la selección para el caso de trayectoria tridimensional.</i>	133
<i>Figura 53: Diagramas de cajas del error de posición para los distintos algoritmos de selección de satélites.</i>	136
<i>Figura 54: Niveles de protección horizontal y vertical.</i>	148
<i>Figura 55: Cálculo del nivel de protección.</i>	149
<i>Figura 56: Diagrama de Stanford.</i>	153

<i>Figura 57: Ilustración del concepto IBPL.</i>	159
<i>Figura 58: Función de Huber.</i>	163
<i>Figura 59: Pesos de la función de Huber.</i>	164
<i>Figura 60: Funciones de densidad de probabilidad del error de posición para el caso estático con un satélite con error constante en el pseudorange.</i>	167
<i>Figura 61: Niveles de protección del RAIM convencional para el caso estático con un satélite con error constante en el pseudorange.</i>	169
<i>Figura 62: Diagrama de Stanford del componente horizontal del RAIM convencional para el caso estático con un satélite con error constante en el pseudorange.</i>	170
<i>Figura 63: Diagrama de Stanford del componente vertical del RAIM convencional para el caso estático con un satélite con error constante en el pseudorange.</i>	170
<i>Figura 64: Niveles de protección del RAIM basado en la separación de soluciones para el caso estático con un satélite con error constante en el pseudorange.</i>	171
<i>Figura 65: Diagrama de Stanford del componente horizontal del RAIM basado en separación de soluciones para el caso estático con un satélite con error constante en el pseudorange.</i>	172
<i>Figura 66: Diagrama de Stanford del componente vertical del RAIM basado en separación de soluciones para el caso estático con un satélite con error constante en el pseudorange.</i>	172
<i>Figura 67: Niveles de protección del IBPL para el caso estático con un satélite con error constante en el pseudorange.</i>	173
<i>Figura 68: Diagrama de Stanford del componente horizontal del IBPL para el caso estático con un satélite con error constante en el pseudorange.</i>	174
<i>Figura 69: Diagrama de Stanford del componente vertical del IBPL para el caso estático con un satélite con error constante en el pseudorange.</i>	174
<i>Figura 70: Función de densidad de probabilidad del error de posición para el caso estático con dos satélites con errores constantes en el pseudorange.</i>	175
<i>Figura 71: Niveles de protección para el RAIM convencional para el caso estático con dos satélites con errores constantes en el pseudorange.</i>	177
<i>Figura 72: Diagrama de Stanford del componente horizontal para el RAIM convencional para el caso estático con dos satélites con errores constantes en el pseudorange.</i>	178
<i>Figura 73: Diagrama de Stanford del componente vertical para el RAIM convencional para el caso estático con dos satélites con errores constantes en el pseudorange.</i>	178
<i>Figura 74: Niveles de protección para el RAIM basado en separación de soluciones para el caso estático con dos satélites con errores constantes en el pseudorange.</i>	179
<i>Figura 75: Diagrama de Stanford del componente horizontal para el RAIM basado en separación de soluciones para el caso estático con dos satélites con errores constantes en el pseudorange.</i>	180
<i>Figura 76: Diagrama de Stanford del componente vertical para el RAIM basado en separación de soluciones para el caso estático con dos satélites con errores constantes en el pseudorange.</i>	180

<i>Figura 77: Niveles de protección para el IBPL para el caso estático con dos satélites con errores constantes en el pseudorange.</i>	181
<i>Figura 78: Diagrama de Stanford del componente horizontal para el IBPL para el caso estático con dos satélites con errores constantes en el pseudorange.</i>	182
<i>Figura 79: Diagrama de Stanford del componente vertical para el IBPL para el caso estático con dos satélites con errores constantes en el pseudorange.</i>	182
<i>Figura 80: Función de densidad de probabilidad del error de posición para el caso estático con tres satélites con errores constantes en el pseudorange.</i>	183
<i>Figura 81: Función de densidad de probabilidad del error de posición para el estimador-M IGGII para el caso estático con tres satélites con errores constantes en el pseudorange.</i>	184
<i>Figura 82: Niveles de protección para el RAIM convencional para el caso estático con tres satélites con errores constantes en el pseudorange.</i>	185
<i>Figura 83: Diagrama de Stanford del componente vertical para el RAIM convencional para el caso estático con tres satélites con errores constantes en el pseudorange.</i>	186
<i>Figura 84: Diagrama de Stanford del componente horizontal para el RAIM convencional para el caso estático con tres satélites con errores constantes en el pseudorange.</i>	186
<i>Figura 85: Niveles de protección para el RAIM basado en separación de soluciones para el caso estático con tres satélites con errores constantes en el pseudorange.</i>	187
<i>Figura 86: Diagrama de Stanford del componente horizontal para el RAIM basado en separación de soluciones para el caso estático con tres satélites con errores constantes en el pseudorange.</i>	188
<i>Figura 87: Diagrama de Stanford del componente vertical para el RAIM basado en separación de soluciones para el caso estático con tres satélites con errores constantes en el pseudorange.</i>	188
<i>Figura 88: Niveles de protección para el IBPL para el caso estático con tres satélites con errores constantes en el pseudorange.</i>	189
<i>Figura 89: Diagrama de Stanford del componente horizontal para el IBPL para el caso estático con tres satélites con errores constantes en el pseudorange.</i>	190
<i>Figura 90: Diagrama de Stanford del componente vertical para el IBPL para el caso estático con tres satélites con errores constantes en el pseudorange.</i>	190
<i>Figura 91: Número de satélites visibles y error de posición para el caso estático con error creciente.</i>	191
<i>Figura 92: Funciones de densidad de probabilidad del error de posición para el caso estático con error creciente (1).</i>	192
<i>Figura 93: Funciones de densidad de probabilidad del error de posición para el caso estático con error creciente (2).</i>	193
<i>Figura 94: Niveles de protección para el RAIM convencional para el caso estático con error creciente.</i>	194
<i>Figura 95: Número de satélites visibles y utilizados para el cálculo del nivel de protección para el RAIM convencional para el caso estático con error creciente.</i>	195
<i>Figura 96: Diagrama de Stanford del componente horizontal para el RAIM convencional para el caso estático con error creciente.</i>	196

<i>Figura 97: Diagrama de Stanford del componente vertical para el RAIM convencional para el caso estático con error creciente.</i>	196
<i>Figura 98: Niveles de protección para el RAIM basado en separación de soluciones para el caso estático con error creciente.</i>	197
<i>Figura 99: Número de satélites visibles y utilizados para el cálculo del nivel de protección para el RAIM basado en separación de soluciones para el caso estático con un error creciente.</i>	198
<i>Figura 100: Diagrama de Stanford del componente horizontal para el RAIM basado en separación de soluciones para el caso estático con error creciente.</i>	199
<i>Figura 101: Diagrama de Stanford del componente vertical para el RAIM basado en separación de soluciones para el caso estático con error creciente.</i>	199
<i>Figura 102: Niveles de protección para el IBPL para el caso estático con error creciente.</i>	200
<i>Figura 103: Diagrama de Stanford del componente horizontal para el IBPL para el caso estático con error creciente.</i>	201
<i>Figura 104: Diagrama de Stanford del componente vertical para el IBPL para el caso estático con error creciente.</i>	201
<i>Figura 105: FDP del error de posición para el algoritmo HEA para el caso de trayectoria bidimensional.</i>	228
<i>Figura 106: Valores del PDOP para el algoritmo HEA para el caso de trayectoria bidimensional.</i>	228
<i>Figura 107: Valores del PDOP para el algoritmo MVA para el caso de trayectoria bidimensional.</i>	229
<i>Figura 108: FDP del error de posición para el algoritmo MVA para el caso de trayectoria bidimensional.</i>	229
<i>Figura 109: FDP del error de posición para el algoritmo Dartboard DOP para el caso de trayectoria bidimensional.</i>	230
<i>Figura 110: Valores del PDOP para el algoritmo Dartboard DOP para el caso de trayectoria bidimensional.</i>	230
<i>Figura 111: Valores del PDOP para el algoritmo WPDOP para el caso de trayectoria bidimensional.</i>	231
<i>Figura 112: FDP del error de posición para el algoritmo WPDOP para el caso de trayectoria bidimensional.</i>	231
<i>Figura 113: Valores del PDOP para el algoritmo EPR para el caso de trayectoria bidimensional.</i>	232
<i>Figura 114: FDP del error de posición para el algoritmo EPR para el caso de trayectoria bidimensional.</i>	232
<i>Figura 118: Valores del PDOP para el algoritmo HEA para el caso de trayectoria tridimensional.</i>	233
<i>Figura 119: FDP del error de posición para el algoritmo HEA para el caso de trayectoria tridimensional.</i>	233
<i>Figura 120: Valores del PDOP para el algoritmo MVA para el caso de trayectoria tridimensional.</i>	234
<i>Figura 121: FDP del error de posición para el algoritmo MVA para el caso de trayectoria tridimensional.</i>	234

Figura 122: Valores del PDOP para el algoritmo Dartboard DOP para el caso de trayectoria tridimensional.

235

Figura 123: FDP del error de posición para el algoritmo Dartboard DOP para el caso de trayectoria tridimensional.

235

Figura 124: Valores del PDOP para el algoritmo WPDOP para el caso de trayectoria tridimensional.

236

Figura 125: FDP del error de posición para el algoritmo WPDOP para el caso de trayectoria tridimensional.

236

Figura 126: Valores del PDOP para el algoritmo EPR para el caso de trayectoria tridimensional.

237

Figura 127: FDP del error de posición para el algoritmo EPR para el caso de trayectoria tridimensional.

237

Figura 128: Funciones de densidad de probabilidad del error de posición con un satélite con error constante para el caso de trayectoria bidimensional.

239

Figura 129: Niveles de protección para el RAIM convencional con un satélite con error constante para el caso de trayectoria bidimensional.

240

Figura 130: Diagrama de Stanford del componente horizontal para el RAIM convencional con un satélite con error constante para el caso de trayectoria bidimensional.

241

Figura 131: Diagrama de Stanford del componente horizontal para el RAIM convencional con un satélite con error constante para el caso de trayectoria bidimensional.

241

Figura 132: Niveles de protección del RAIM basado en separación de soluciones para el caso de trayectoria bidimensional con un satélite con error constante en el pseudorange.

242

Figura 133: Diagrama de Stanford del componente horizontal para el RAIM basado en separación de soluciones con un satélite con error constante para el caso de trayectoria bidimensional.

243

Figura 134: Diagrama de Stanford del componente vertical para el RAIM basado en separación de soluciones con un satélite con error constante para el caso de trayectoria bidimensional.

243

Figura 135: Niveles de protección del IBPL para el caso de trayectoria bidimensional con un satélite con error constante en el pseudorange.

244

Figura 136: Diagrama de Stanford del componente horizontal para el IBPL con un satélite con error constante para el caso de trayectoria bidimensional.

245

Figura 137: Diagrama de Stanford del componente vertical para el IBPL con un satélite con error constante para el caso de trayectoria bidimensional.

245

Figura 138: Funciones de densidad de probabilidad del error de posición con dos satélites con errores constantes para el caso de trayectoria bidimensional.

246

Figura 139: Niveles de protección para el RAIM convencional con dos satélites con errores constantes para el caso de trayectoria bidimensional.

247

Figura 140: Diagrama de Stanford del componente horizontal para el RAIM convencional con dos satélites con errores constantes para el caso de trayectoria bidimensional.

248

Figura 141: Diagrama de Stanford del componente horizontal para el RAIM convencional con dos satélites con errores constantes para el caso de trayectoria bidimensional.

248

<i>Figura 142: Niveles de protección del RAIM basado en separación de soluciones para el caso de trayectoria bidimensional con dos satélites con errores constantes en el pseudorango.</i>	249
<i>Figura 143: Diagrama de Stanford del componente horizontal para el RAIM basado en separación de soluciones con dos satélites con errores constantes para el caso de trayectoria bidimensional.</i>	250
<i>Figura 144: Diagrama de Stanford del componente vertical para el RAIM basado en separación de soluciones con dos satélites con errores constantes para el caso de trayectoria bidimensional.</i>	250
<i>Figura 145: Niveles de protección del IBPL para el caso de trayectoria bidimensional con un satélite con error constante en el pseudorango.</i>	251
<i>Figura 146: Diagrama de Stanford del componente horizontal para el IBPL con dos satélites con errores constantes para el caso de trayectoria bidimensional.</i>	251
<i>Figura 147: Diagrama de Stanford del componente vertical para el IBPL con dos satélites con errores constantes para el caso de trayectoria bidimensional.</i>	252
<i>Figura 148: Funciones de densidad de probabilidad del error de posición con tres satélites con errores constantes para el caso de trayectoria bidimensional.</i>	253
<i>Figura 149: Función de densidad de probabilidad del error de posición para el estimador IGGII con tres satélites con errores constantes para el caso de trayectoria bidimensional.</i>	254
<i>Figura 150: Niveles de protección para el RAIM convencional con tres satélites con errores constantes para el caso de trayectoria bidimensional.</i>	255
<i>Figura 151: Diagrama de Stanford del componente horizontal para el RAIM convencional con tres satélites con errores constantes para el caso de trayectoria bidimensional.</i>	256
<i>Figura 152: Diagrama de Stanford del componente vertical para el RAIM convencional con tres satélites con errores constantes para el caso de trayectoria bidimensional.</i>	256
<i>Figura 153: Niveles de protección para el RAIM basado en separación de soluciones con tres satélites con errores constantes para el caso de trayectoria bidimensional.</i>	257
<i>Figura 154: Diagrama de Stanford del componente horizontal para el RAIM basado en separación de soluciones con tres satélites con errores constantes para el caso de trayectoria bidimensional.</i>	258
<i>Figura 155: Diagrama de Stanford del componente vertical para el RAIM basado en separación de soluciones con tres satélites con errores constantes para el caso de trayectoria bidimensional.</i>	258
<i>Figura 156: Niveles de protección para el IBPL con tres satélites con errores constantes para el caso de trayectoria bidimensional.</i>	259
<i>Figura 157: Diagrama de Stanford del componente horizontal para el IBPL con tres satélites con errores constantes para el caso de trayectoria bidimensional.</i>	260
<i>Figura 158: Diagrama de Stanford del componente vertical para el IBPL con tres satélites con errores constantes para el caso de trayectoria bidimensional.</i>	260
<i>Figura 159: Funciones de densidad de probabilidad del error de posición con un satélite con error constante para el caso de trayectoria bidimensional.</i>	261
<i>Figura 160: Niveles de protección para el RAIM convencional con un satélite con error constante para el caso de trayectoria bidimensional.</i>	262

<i>Figura 161: Diagrama de Stanford del componente horizontal para el RAIM convencional con un satélite con error constante para el caso de trayectoria bidimensional.</i>	263
<i>Figura 162: Diagrama de Stanford del componente horizontal para el RAIM convencional con un satélite con error constante para el caso de trayectoria bidimensional.</i>	263
<i>Figura 163: Niveles de protección del RAIM basado en separación de soluciones para el caso de trayectoria bidimensional con un satélite con error constante en el pseudorange.</i>	264
<i>Figura 164: Diagrama de Stanford del componente horizontal para el RAIM basado en separación de soluciones con un satélite con error constante para el caso de trayectoria bidimensional.</i>	264
<i>Figura 165: Diagrama de Stanford del componente vertical para el RAIM basado en separación de soluciones con un satélite con error constante para el caso de trayectoria bidimensional.</i>	265
<i>Figura 166: Niveles de protección del IBPL para el caso de trayectoria bidimensional con un satélite con error constante en el pseudorange.</i>	266
<i>Figura 167: Diagrama de Stanford del componente horizontal para el IBPL con un satélite con error constante para el caso de trayectoria bidimensional.</i>	266
<i>Figura 168: Diagrama de Stanford del componente vertical para el IBPL con un satélite con error constante para el caso de trayectoria bidimensional.</i>	267
<i>Figura 169: Funciones de densidad de probabilidad del error de posición con dos satélites con errores constantes para el caso de trayectoria bidimensional.</i>	268
<i>Figura 170: Niveles de protección para el RAIM convencional con dos satélites con errores constantes para el caso de trayectoria bidimensional.</i>	269
<i>Figura 171: Diagrama de Stanford del componente horizontal para el RAIM convencional con dos satélites con errores constantes para el caso de trayectoria bidimensional.</i>	269
<i>Figura 172: Diagrama de Stanford del componente horizontal para el RAIM convencional con dos satélites con errores constantes para el caso de trayectoria bidimensional.</i>	270
<i>Figura 173: Niveles de protección del RAIM basado en separación de soluciones para el caso de trayectoria bidimensional con dos satélites con errores constantes en el pseudorange.</i>	271
<i>Figura 174: Diagrama de Stanford del componente horizontal para el RAIM basado en separación de soluciones con dos satélites con errores constantes para el caso de trayectoria bidimensional.</i>	271
<i>Figura 175: Diagrama de Stanford del componente vertical para el RAIM basado en separación de soluciones con dos satélites con errores constantes para el caso de trayectoria bidimensional.</i>	272
<i>Figura 176: Niveles de protección del IBPL para el caso de trayectoria bidimensional con un satélite con error constante en el pseudorange.</i>	273
<i>Figura 177: Diagrama de Stanford del componente horizontal para el IBPL con dos satélites con errores constantes para el caso de trayectoria bidimensional.</i>	273
<i>Figura 178: Diagrama de Stanford del componente vertical para el IBPL con dos satélites con errores constantes para el caso de trayectoria bidimensional.</i>	274
<i>Figura 179: Funciones de densidad de probabilidad del error de posición con tres satélites con errores constantes para el caso de trayectoria bidimensional.</i>	275

<i>Figura 180: Función de densidad de probabilidad del error de posición para el estimador IGGII con tres satélites con errores constantes para el caso de trayectoria bidimensional.</i>	276
<i>Figura 181: Niveles de protección para el RAIM convencional con tres satélites con errores constantes para el caso de trayectoria bidimensional.</i>	277
<i>Figura 182: Diagrama de Stanford del componente horizontal para el RAIM convencional con tres satélites con errores constantes para el caso de trayectoria bidimensional.</i>	277
<i>Figura 183: Diagrama de Stanford del componente vertical para el RAIM convencional con tres satélites con errores constantes para el caso de trayectoria bidimensional.</i>	278
<i>Figura 184: Niveles de protección para el RAIM basado en separación de soluciones con tres satélites con errores constantes para el caso de trayectoria bidimensional.</i>	279
<i>Figura 185: Diagrama de Stanford del componente horizontal para el RAIM basado en separación de soluciones con tres satélites con errores constantes para el caso de trayectoria bidimensional.</i>	279
<i>Figura 186: Diagrama de Stanford del componente vertical para el RAIM basado en separación de soluciones con tres satélites con errores constantes para el caso de trayectoria bidimensional.</i>	280
<i>Figura 187: Niveles de protección para el IBPL con tres satélites con errores constantes para el caso de trayectoria bidimensional.</i>	281
<i>Figura 188: Diagrama de Stanford del componente horizontal para el IBPL con tres satélites con errores constantes para el caso de trayectoria bidimensional.</i>	281
<i>Figura 189: Diagrama de Stanford del componente vertical para el IBPL con tres satélites con errores constantes para el caso de trayectoria bidimensional.</i>	282

Índice de tablas

Tabla 1: Clasificación saltos de ciclo aplicados según la frecuencia.	50
Tabla 2: Indicadores detección salto de ciclo. Algoritmo de ajuste polinómico con frecuencia simple.	53
Tabla 3: Indicadores detección salto de ciclo. Algoritmo de media y desviación con frecuencia simple.	56
Tabla 4: Indicadores detección salto de ciclo. Algoritmo de combinación de geometría libre con frecuencia múltiple.	57
Tabla 5: Indicadores detección salto de ciclo. Algoritmo de combinación de geometría libre con frecuencia múltiple. Umbral de detección duplicado.	58
Tabla 6: Indicadores detección salto de ciclo. Algoritmo de combinación de geometría libre con frecuencia múltiple. Sin diferenciar la combinación de geometría libre.	60
Tabla 7: Indicadores detección salto de ciclo. Algoritmo de combinación Melbourne-Wübbena con frecuencia múltiple.	63
Tabla 8: Indicadores detección salto de ciclo. Algoritmo de combinación de geometría libre con frecuencia múltiple. Satélite con poca elevación.	65
Tabla 9: Indicadores detección salto de ciclo. Algoritmo de combinación de geometría libre con frecuencia múltiple. Sin diferenciar la combinación de geometría libre. Satélite con poca elevación.	66
Tabla 10: Indicadores detección salto de ciclo. Algoritmo de combinación Melbourne-Wübbena con frecuencia múltiple. Satélite con poca elevación.	67
Tabla 11: Número de satélites visibles según distintas combinaciones de sistemas GNSS.	75
Tabla 12: Valor del GDOP según distintas combinaciones de sistemas GNSS.	76
Tabla 13: Número de multiplicaciones y sumas en comparación con el caso de 10 satélites.	87
Tabla 14: Número de cambios del conjunto de satélites seleccionados.	109
Tabla 15: Valores máximos, mínimos y medios para el tiempo que permanecen los algoritmos sin modificar la selección.	110
Tabla 16: Número de cambios del conjunto de satélites seleccionados para el caso de trayectoria bidimensional.	129
Tabla 17: Valores máximos, mínimos y medios para el tiempo que permanecen los algoritmos sin modificar la selección para el caso de trayectoria bidimensional.	130
Tabla 18: Valores medios del PDOP para los distintos algoritmos de selección de satélites.	137
Tabla 19: Tasas de exclusión de medidas erróneas para el caso estático con un error constante.	168
Tabla 20: Comparación del caso estático de las técnicas de detección de medidas erróneas.	207
Tabla 21: Resumen valoración algoritmos de detección de saltos de ciclo.	213
Tabla 22: Resumen valoración algoritmos de selección de satélites.	214
Tabla 23: Resumen valoración algoritmos de detección de medidas erróneas.	216

CAPÍTULO 1:

Introducción

En este primer capítulo se van a introducir las razones que motivan el desarrollo de este trabajo fin de master. Además, se describirán los objetivos que se pretenden lograr.

1.1 Motivación

Hoy en día, los sistemas de navegación por satélite están presentes en multitud de ámbitos cotidianos. Para cada aplicación, se requiere que el sistema de navegación sea capaz de proporcionar un nivel de precisión concreto, además de poder asegurar la integridad de la solución.

Existen en la actualidad varios sistemas de navegación que dan servicio global: GPS, Galileo, GLONASS y BeiDou. Cada uno de estos sistemas cuenta con su propia flota de satélites. Está demostrado que el uso de los distintos sistemas de forma conjunta puede mejorar la navegación (Teng & Wang, New Characteristics of Geometric Dilution of Precision (GDOP) for Multi-GNSS Constellations, 2014).

Utilizando varios sistemas de navegación, se tiene acceso a un mayor número de satélites visibles. Esta situación es de interés para entornos que cuentan con múltiples obstáculos, como puede ser un entorno urbano. Un mayor número de satélites visibles implica una mejor geometría lo que redundará en un incremento en la precisión de la solución.

Se define una figura de mérito para la disposición de los satélites utilizados en el cálculo de la solución de posicionamientos GNSS, el DOP (Dilution Of Precision). El DOP se puede descomponer en una serie de factores que afectan a la precisión alcanzable en el posicionamiento o en la finura de la sincronización de la solución con la base de tiempos del sistema de satélites utilizado. Los factores del DOP relacionan la geometría espacial que conforman los satélites utilizados en la navegación con la precisión que es capaz de proporcionar dicho sistema. Aumentar el número de satélites disponibles conlleva una mejoría en los factores del DOP. Esta mejoría es considerablemente menos pronunciada cuando ya se cuenta un número aceptable de satélites. En cambio, continuar aumentando el

número de satélites utilizados en la navegación incrementa de forma significativa la carga computacional del sistema. En otras palabras, existe un momento en el cual agregar más satélites implica una carga computacional exagerada y no beneficia especialmente la precisión de la solución.

Con esto, parece necesario el uso de un método que seleccione un subconjunto de satélites de los que se encuentren disponibles que proporcione un nivel de precisión adecuado. La minimización de los factores del DOP para la selección procura el subconjunto óptimo de satélites, pero este método implica llevar a cabo el cálculo del DOP para todas combinaciones posibles de satélites. Para muchas aplicaciones de navegación esta carga computacional no puede ser asumida considerando una ejecución en tiempo real con una tasa de refresco elevada, especialmente cuando se requieren plataformas pequeñas y baratas que ofrezcan un rápido refresco de la solución con un mínimo consumo energético, como pueden ser las aplicaciones actualmente en boga de Internet Of Things o navegación de pequeñas aeronaves no tripuladas.

Por otro lado, el contar con un mayor número de satélites visibles también tiene un efecto positivo en cuanto a la integridad del sistema de navegación. La redundancia de las medidas disponibles se utiliza para llevar a cabo comprobaciones de integridad. Las técnicas RAIM son unas técnicas orientadas a la detección y exclusión de medidas erróneas. Las técnicas RAIM operan mediante la comparación de estadísticos y estimadores contruidos a partir de las observaciones individuales para cada satélite procesado.

Los algoritmos RAIM se desarrollaron suponiendo la presencia de una única medida errónea, ya que se centraban en monitorizar la integridad de los sistemas de aviación donde la aparición de más de una medida errónea es sumamente improbable. Esta suposición no se sostiene para muchas de las aplicaciones en la que se hace uso de la navegación por satélite hoy en día.

Por ello, resulta de interés estudiar como las metodologías de selección de satélites pueden aprovecharse de un mayor grado de redundancia gracias al uso

de los sistemas multi GNSS. Además, se debe analizar si son capaces de asegurar un nivel de integridad cuando se encuentran en un entorno propicio a la aparición de más de una medida errónea.

Con el estudio de los métodos de selección de satélites y de detección de medidas erróneas se pretende lograr una optimización del desempeño de los sistemas de navegación. Otra cuestión con potencial para mejorar el desempeño es la detección de saltos de ciclo. Cuando la señal que transmite un satélite se ve interrumpida, se produce un offset en la medida de fase. Si esto no se detecta, el error se propaga por el sistema y resulta en una pérdida de precisión temporal. Existe poca información acerca de los algoritmos de detección de saltos de ciclo ya que la mayoría de estos algoritmos se integran en los esquemas propietarios de procesamiento GNSS de los fabricantes de receptores.

La mayoría de los errores que afectan al cálculo efectivo de la distancia del receptor a los satélites se pueden corregir mediante el modelado del error o combinaciones de las diferentes señales transmitidas. No obstante, el ruido en la medida no puede ser suprimido. Afortunadamente existe un modo de mitigar este ruido, se puede usar la medida de la variación de la fase de la portadora para estimar la variación de la distancia. La fase de la portadora se obtiene con mucha precisión y poco ruido pero está afectada por la ambigüedad que introduce desconocer el número entero de longitudes de onda de señal que caben en la distancia recorrida por la señal. No obstante, los circuitos de seguimiento de señal son muy estables y permiten tener una medida precisa de la variación de fase entre observaciones, siempre que no pierdan el seguimiento de la señal y la ambigüedad sea la misma en ambas observaciones.

Por tanto, la detección de saltos de ciclo es una herramienta crucial para no perder el seguimiento de la señal y mantener el valor de la ambigüedad constante. La detección de saltos de ciclo permite hacer uso de la precisión de la medida de la variación de la fase de la señal portadora y mitigar el ruido de las medidas.

1.2 Objetivos

Este trabajo plantea la consecución de una serie de objetivos estrechamente relacionados con lo expuesto en la sección anterior. En primer lugar, debido a la importancia de los saltos de ciclo en las soluciones obtenidas por los sistemas de navegación, resulta primordial hacer lo posible por llevar a cabo su detección para limitar su efecto sobre la solución. Por ello, se evaluarán y validarán distintos algoritmos para detectar los saltos de ciclo.

En segundo lugar, debido al aumento de la carga computacional que conlleva el uso de un mayor número de satélites, es necesario realizar una selección de entre los satélites disponibles para computar una solución óptima sin tener que hacer uso de todos los satélites visibles. Para esto, se valorarán distintas técnicas de selección de satélites evaluando el nivel de carga computacional requerido.

En tercer lugar, debido a la considerable redundancia de medidas que proporcionan los sistemas multiconstelación, se estudiarán diversos métodos de detección y exclusión de medidas erróneas.

Para llevar a cabo los análisis mencionados se contará con una serie de datos provenientes de simulaciones de los sistemas GPS y Galileo. Dichas simulaciones tienen en cuenta todas las influencias físicas que actúan sobre un sistema de navegación real con el objetivo de asemejarse en medida de lo posible a la realidad.

CAPÍTULO 2: Navegación por satélite

En este capítulo se plantean y se explican los fundamentos de la navegación por satélite. Los conceptos descritos a continuación son fundamentales para comprender la aportación que supone este trabajo y que se describirá en los capítulos que siguen. Además de introducir las nociones básicas de los sistemas de navegación, se especifican los 3 observables de los sistemas GNSS y se detallan los distintos errores e influencias físicas que afectan a las señales de estos sistemas.

2.1 Introducción

El término GNSS (*Global Navigation Satellite System*) comprende los distintos sistemas de navegación por satélite presentes en la actualidad. Las constelaciones GNSS proporcionan capacidades e resolver la posición y velocidad del receptor, además de proporcionar un mecanismo muy preciso de sincronización a una escala común de tiempo. Estas capacidades se explotan en aplicaciones de navegación, transporte, geodésicos, hidrográficos o agrícolas. Si se conoce la posición de los satélites en todo momento y se es capaz de estimar la distancia del receptor a ellos mediante la medida de las señales recibidas, es posible calcular la posición del receptor mediante técnicas de optimización. Además, los sistemas de navegación también proporcionan una manera de sincronizar los relojes de los distintos dispositivos (que cuentan con sistema de navegación) con una exactitud considerable (del orden del nanosegundo).

Fundamentalmente, un sistema de navegación está compuesto por una serie de satélites que portan un reloj atómico y se encuentran orbitando la Tierra siguiendo unas órbitas conocidas. Dichos satélites transmiten señales que son recibidas en la Tierra. Si el receptor es capaz de conocer el momento en el que se envió el mensaje y el momento en el que fue recibido, es posible obtener una distancia (denominada pseudodistancia ya que hace referencia al rango entre el satélite en el momento de emisión y el receptor en el momento de recepción) que se utilizará para computar la posición del receptor.

Se requiere de 3 satélites de una misma constelación para obtener las 3 coordenadas de un receptor. Una parte importante de los sistemas de navegación es la sincronización de los relojes presentes tanto en los satélites como en los receptores. Esta sincronización permite obtener el tiempo necesario de los mensajes transmitidos por los satélites en alcanzar el receptor. Este tiempo se suele denominar tiempo de propagación, y es de vital importancia para el cálculo de la distancia satélite-receptor. Para asegurar esta sincronización, se debe contar con otro satélite (4 satélites en total) para obtener también el tiempo como una variable.

Los receptores deben estar exáctamente sincronizados al tiempo GNSS correspondiente ya que la distancia al satélite se calcula comparando las marcas de tiempo asociadas al flanco de los mensajes recibidos con la copia interna (sincronizada) que tiene el receptor del oscilador que marca el tiempo de la constelación. Mediante la comparación del instante de recepción del mensaje con el tiempo de reloj dado por el receptor se calcula el tiempo de propagación. No obstante, esta estimación se ve afectada por una serie de errores que se describen en este capítulo.

Dentro del concepto de GNSS existen varios tipos sistemas de navegación, atendiendo a la zona que dan cobertura. Existen sistemas de navegación globales que proporcionan servicio en todo el mundo. También existen sistemas regionales que se limitan a proporcionar servicio en ciertas regiones geográficas. Por último, existen los denominados sistemas de aumentación que informan sobre la calidad de las señales y son capaces de mejorar el posicionamiento de los otros sistemas. Este trabajo fin de master se va a centrar en el primer grupo.

Cada sistema de navegación tiene su propia constelación de satélites, y estos siguen órbitas diferentes a las de otras constelaciones. Además, cada constelación de satélites transmite mensajes en distintas frecuencias, y estos mensajes contienen la información relevante para cada sistema de navegación particular.

Es importante mencionar que cada sistema define su propia referencia de tiempos, aunque se busca vincularla a una referencia continua fija estandarizada, el tiempo atómico internacional. A pesar de que estas referencias se vinculan al mismo estándar, las desviaciones son propias para cada sistema, por lo que tienen que calcularse por separado cuando se procesan datos de varias constelaciones. Por este motivo, si se desea utilizar satélites de distintas constelaciones se debe tener un satélite adicional por cada constelación adicional que se vaya a utilizar. Es decir, si se utiliza únicamente GPS, se podrá navegar utilizando un mínimo 4 satélites; mientras que si se utiliza GPS y Galileo, serán necesarios al menos 5 satélites.

Algunos de estos sistemas de navegación ofrecen un servicio civil y uno encriptado o militar, que se encuentra cifrado para controlar su acceso y uso. Aunque este trabajo se va a centrar en el servicio civil, las conclusiones son fácilmente extrapolables al servicio encriptado, ya que se ve afectado por los mismos errores.

De los sistemas de navegación existentes en la actualidad se pueden destacar tres:

a) GPS

En el presente, el GPS (Global Positioning System) es el sistema de navegación más utilizado. El GPS pertenece al gobierno de los Estados Unidos y se necesitan 24 satélites para asegurar la operabilidad. En la actualidad está preparado para ofrecer 32 satélites funcionales, rebasando con creces la capacidad mínima de disponibilidad.

b) GLONASS

El sistema de navegación GLONASS (Global-naya Navigatsionnaya Sputnikova Systema) desarrollado por la federación rusa también necesita 24 satélites operativos aun que cuenta con 31.

c) Galileo

Galileo es el sistema de navegación europeo y en la actualidad no se encuentra plenamente operativo. Se ha diseñado para contar con 30 satélites, de los cuales 18 sí que se encuentran operativos. Galileo se ha preparado para ser compatible con otros sistemas de navegación por satélite.

2.2 Segmentos

Los sistemas de navegación por satélite se componen de tres segmentos: espacio, control y usuario. Esta descomposición facilita la comprensión del funcionamiento del sistema completo.

2.2.1 Segmento espacial

El segmento espacial está compuesto por la constelación de satélites que tienen como fin transmitir la información necesaria para obtener la posición del receptor. El mensaje que envían dichos satélites está codificado con una señal PRN (*Pseudo Random Noise*). Esta señal modula el mensaje y tiene como objetivo la identificación de información perteneciente a cada satélite en particular.

Las señales PRN hacen referencia a la codificación de los mensajes del sistema GPS. El sistema Galileo modula con otro código y transmite las señales utilizando igual frecuencia que el sistema GPS. Sin embargo, este no es el caso de GLONASS, el sistema ruso no comparte el medio modulando la señal de cada satélite con un código propio ortogonal a los demás, en su lugar asigna una frecuencia ligeramente diferente a cada satélite. En cuestión de años el sistema ruso pasará a trabajar igual que los sistemas de GPS y Galileo.

2.2.2 Segmento de control

El segmento de control es responsable del seguimiento y el mantenimiento de los satélites en funcionamiento. Este segmento se ocupa de analizar el estado de comunicaciones de los satélites en órbita y monitorizar su trayecto orbital. Si se diese el caso de la avería de uno de los satélites que se encuentran en funcionamiento, el segmento de control se encargaría de activar los satélites de repuesto para mantener la operatividad del sistema.

Además, el segmento de control se encarga de actualizar el reloj del satélite, la efemérides y otros componentes del mensaje de navegación. La efemérides contiene una serie de parámetros físicos particulares para cada satélite y su

órbita que serán necesarios para la obtener la posición de dicho satélite, y por tanto la posición del receptor.

Este segmento se encarga de calcular los parámetros orbitales que deben transmitir los satélites. Estos parámetros son distribuidos por los satélites en el mensaje de efemérides. Los receptores hacen uso de estas efemérides para calcular la posición de los satélites que están observando.

2.2.3 Segmento de usuario

El único elemento del segmento de usuario es el receptor. El receptor procesa el mensaje transmitido por los satélites en órbita y utiliza la información extraída para obtener la solución de posición.

El receptor es el responsable de proporcionar los observables de la navegación por satélite, que se explicarán en la sección siguiente. La Figura 1 muestra un esquema simplificado de un receptor GNSS. Se debe tener en cuenta que todos los componentes del receptor añaden ruido a la medida.

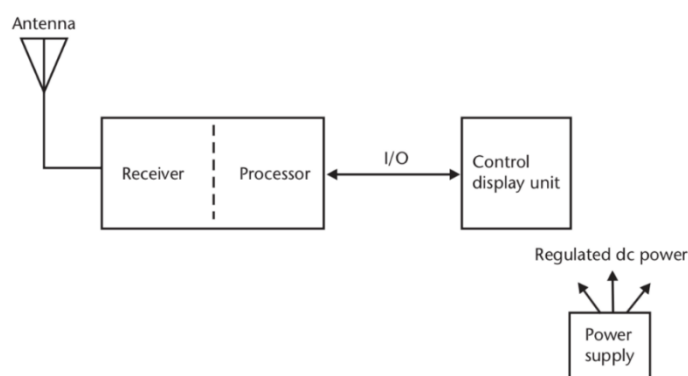


Figura 1: Esquema receptor GNSS.

2.3 Observables

Los sistemas de navegación por satélite hacen uso de tres observables que se obtienen de las transmisiones de los satélites en órbita. El primero de ellos es la distancia viajada por la señal computado desde el satélite en el momento de la emisión del mensaje hasta el receptor en el momento en el que recibe dicho mensaje. Por esto, este observable se denomina pseudorange o pseudodistancia. La frecuencia de la portadora a la que se reciben las señales no es la misma que a la que se transmite debido a la velocidad relativa entre el emisor y el receptor. Esta variación de frecuencia se recoge en el observable Doppler que se utiliza para calcular la velocidad del receptor. Por último, el observable de fase proporciona una medida considerablemente más limpia que el pseudorange, pero contiene una ambigüedad. Para obtener este observable se realiza un seguimiento de la señal recibida del satélite.

2.3.1 Pseudorange

La transmisión de los mensajes de los satélites se lleva a cabo multiplexando dicho mensaje con un código PRN. Los códigos PRN son secuencias conocidas que se repiten de forma periódica, esto facilita que puedan ser replicados en el receptor.

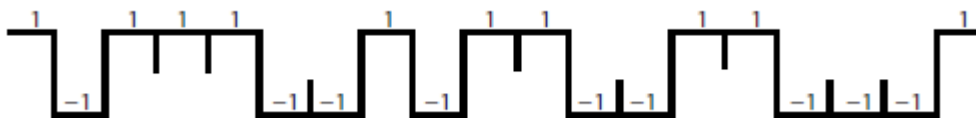


Figura 2: Código PRN.

El tiempo que tarda la transmisión en llegar al receptor se denomina tiempo de propagación. Los receptores GNSS cuentan con un dispositivo llamado PLL (*Phase Lock Loop*) que se encarga de asistir al correlador para computar el tiempo de propagación comparando la señal recibida con un réplica de la señal PRN generada localmente. Por tanto, resulta necesario en primer lugar llevar a

cabo una adquisición de la señal GNSS para luego realizar una correlación con una réplica de la señal recibida.

Cuando el PLL encuentra una correlación significativa entre la señal recibida y la generada, estima el tiempo de propagación. Gracias a la sincronización de relojes de los satélites y el receptor se puede obtener el tiempo transcurrido entre la transmisión del mensaje y la recepción del mismo.

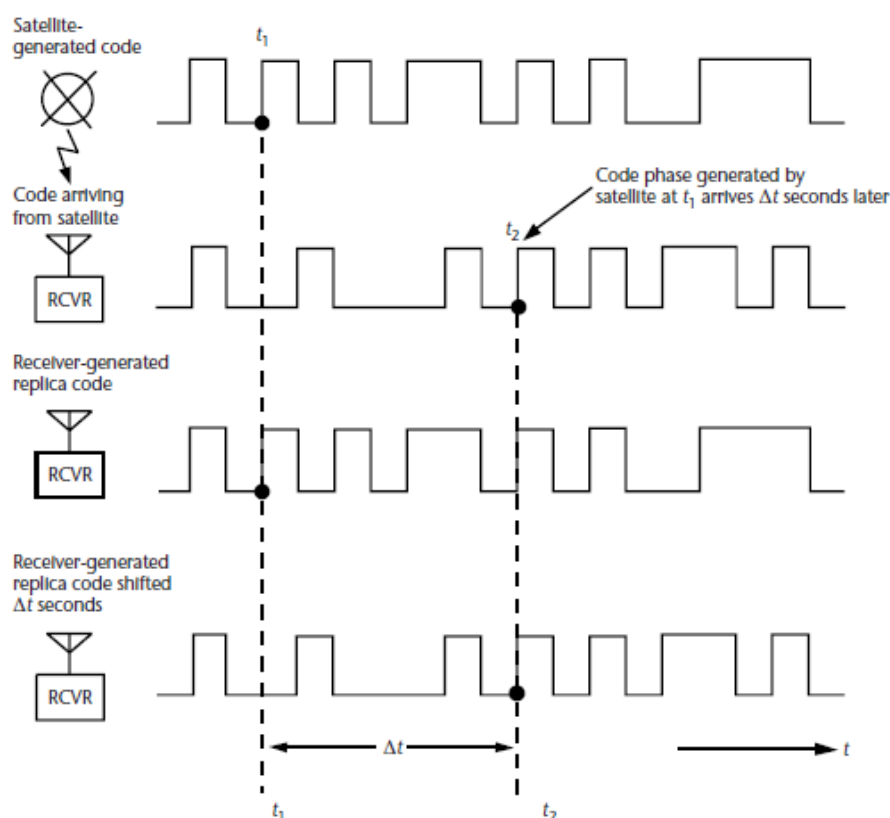


Figura 3: Medida tiempo de vuelo.

Multiplicando el tiempo de propagación por la velocidad de la luz obtenemos una pseudodistancia que hace referencia a la posición del satélite en el momento de transmisión del mensaje y la posición del receptor en el momento de la recepción del mensaje.

La precisión que ofrece el observable de rango puede no ser suficiente para todas las aplicaciones. Esto se debe a que el PLL tiene una cierta tolerancia de alineamiento, por lo que cuando detecta una correlación significativa establece

el alineamiento, pudiendo existir una pequeña incertidumbre en el instante concreto.

2.3.2 Observable Doppler

El efecto Doppler supone una variación de la frecuencia de una señal electromagnética debido al movimiento relativo entre el emisor y el receptor.

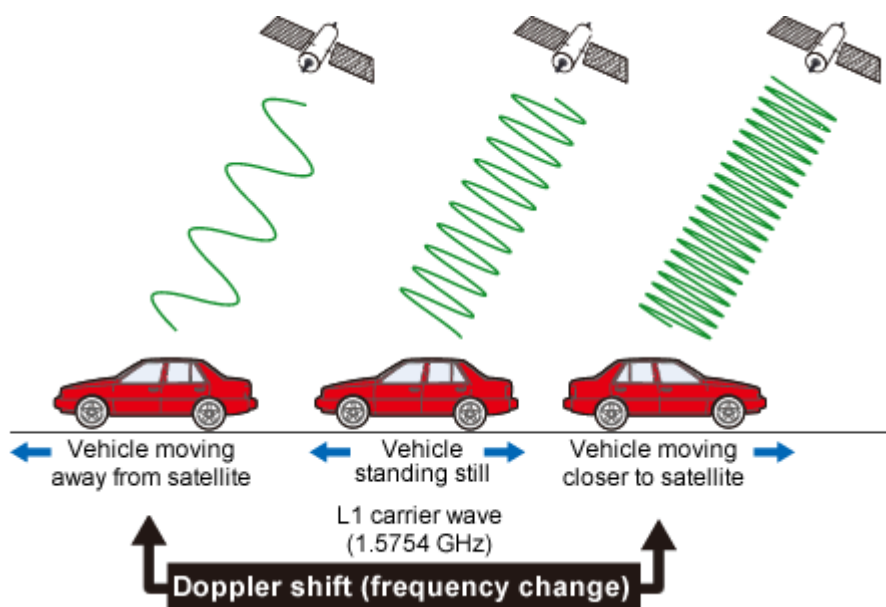


Figura 4: Variación de la frecuencia con el efecto Doppler.

Como se puede observar en la Figura 4, cuando el receptor se aleja del satélite la frecuencia de la señal será menor que cuando se acerca. Debido a este efecto, la frecuencia de la señal recibida no es la misma que a la que fue transmitida.

Los receptores GNSS pueden conocer con una precisión considerable la frecuencia a la que reciben los mensajes de navegación. Mediante la integración del observable Doppler es posible obtener la fase de la señal recibida.

2.3.3 Observable de fase

Con la integración del observable Doppler se puede obtener de forma precisa la fase de la señal recibida. Para ello, se debe llevar a cabo un seguimiento o *tracking* de la señal recibida. Como no es posible conocer el número de

longitudes de onda entre el emisor y el receptor, la medida de la fase contiene una ambigüedad.

Si se utiliza la diferencia entre dos medidas secuenciales, se elimina esta ambigüedad ya que tendrá el mismo valor en las dos medidas. Al realizar esta diferenciación el ruido de la medida se incrementa. Aun así, con el observable de fase se obtiene una mayor precisión que con el pseudorango.

Gracias a la precisión que proporciona el observable de fase es posible corregir el observable de rango. La medida del pseudorango se suaviza utilizando un algoritmo denominado filtro de Hatch.

Dados un observable de rango $R(s, n)$ y un observable de fase $\phi(s, n)$ sobre s satélites y n muestras, se obtiene un pseudorango suavizado mediante:

Ecuación 2.1

$$R'(s, k) = \frac{1}{n} \cdot R(s, k) + \frac{n-1}{n} \cdot [R'(s, k-1) + (\phi(s, k) - \phi(s, k-1))]$$

El algoritmo se inicializa con:

Ecuación 2.2

$$R'(s, 1) = R(s, 1)$$

En la práctica, se aplica este algoritmo utilizando una ventana de N muestra. Con el valor de N oscilando entre 20 y 100, según la aplicación.

El uso del filtro de Hatch debe reservarse para los casos en los que las medidas son correctas, y debe reiniciarse si se cuenta con medidas conteniendo errores, como es el caso de los saltos de ciclo. Ya que un escalón espurio en la medida de fase introduciría un sesgo en la estimación del pseudorango.

2.4 Errores e influencias físicas en las medidas GNSS

Los sistemas de navegación por satélite se ven afectados por una serie de errores y efectos físicos que se deben de tener en cuenta en los cálculos de navegación. Hay errores presentes en los tres segmentos comentados en la sección 2.3.

Existen errores debidos a los relojes de los satélites y del receptor, errores causados por la ionosfera y la troposfera; errores originados por el *multipath*, errores debidos al efecto relativista y por último, errores causados por el ruido.

Incluyendo la existencia de estos errores, los observables de rango y fase toman la siguiente forma:

Ecuación 2.3

$$R(t_r, t_e) = \rho(t_r, t_e) - (\delta t_r - \delta t_e) \cdot c + \delta_{ion} + \delta_{tro} + \delta_{tide} + \delta_{mul} + \delta_{rel} + \varepsilon$$

Donde ρ es la distancia entre emisor y receptor en el momento de emisión y recepción de la señal, respectivamente. El resto de términos hacen referencia a los errores mencionados anteriormente.

Ecuación 2.4

$$\lambda \cdot \phi(t_r) = \rho(t_r, t_e) - (\delta t_r - \delta t_e) \cdot c + \lambda \cdot N - \delta_{ion} + \delta_{tro} + \delta_{tide} + \delta_{mul} + \delta_{rel} + \varepsilon$$

Donde λ es la longitud de onda y N es la ambigüedad.

En cuanto al observable Doppler, como este se mide directamente sobre la portadora no se ve influenciado por los efectos en la transmisión. Por tanto, a este observable únicamente le afecta el ruido térmico.

2.4.1 Errores en los satélites

Puede existir un error en el cálculo de la posición del satélite debido a imprecisiones en las efemérides. Para eliminarlo se acude a los sistemas de aumentación mencionados anteriormente.

También suele aparecer un error en el reloj del satélite, pero este es posible modelarlo para paliar su efecto. Por último, se combate el efecto relativista en el reloj del satélite mediante una modelización precisa a partir de los parámetros orbitales.

En el mensaje de la efemérides se envían los parámetros necesarios para corregir el efecto relativista, los errores instrumentales y la desviación del reloj del satélite.

2.4.2 Efecto de la ionosfera

El efecto ionosférico puede causar el retraso o adelanto de una señal de GNSS de hasta 20 metros en un día. La ionosfera es un medio parcialmente ionizado debido a los rayos X y ultravioleta contenidos en la radiación solar y a la incidencia de partículas cargadas.

La velocidad de propagación de una señal electromagnética en un medio de este tipo depende de la densidad de electrones, la cual varía según el momento del día. La ionosfera es un medio dispersivo, lo que quiere decir que la velocidad de propagación de las ondas depende de la frecuencia. Esto será de utilidad para eliminar el efecto de la ionosfera en las medidas, ya que si se cuenta con más de una frecuencia se puede suprimir el impacto de la ionosfera en los observables.

Debido a la diferencia entre la velocidad de fase y la velocidad de grupo de las señales en el medio ionosférico, tiene lugar un fenómeno denominado divergencia ionosférica. La divergencia ionosférica produce un retraso en la información de la señal (código PRN e información de navegación) y un adelanto en la fase de la señal portadora. Esta es la razón de que en el observable de

rango el efecto ionosférico aparezca sumado y en el observable de fase el término de la ionosfera aparezca restado.

Los sistemas de navegación GPS y Galileo modelan el efecto de la ionosfera mediante el modelo ionosférico de *Klobuchar* y el modelo ionosférico *NeQuick* respectivamente. Aunque para GLONASS no se utiliza ningún modelo de la ionosfera, se pueden aplicar los modelos de GPS y Galileo aplicando un factor de corrección relativo a sus frecuencias. Estos modelos son aplicables únicamente si se cuenta con más de una frecuencia.

2.4.3 Efecto de la troposfera

La troposfera se encuentra en la parte inferior de la atmósfera y a diferencia de la ionosfera, no supone un medio dispersivo para las señales GNSS. Es decir, las señales transmitidas por los sistemas de navegación no se ven afectadas por la frecuencia.

El retraso introducido por la troposfera depende en la temperatura, presión, humedad y en la posición de las antenas del transmisor y el receptor. El efecto de la ionosfera se puede descomponer en dos elementos:

a) Componente hidrostático

Debido a los gases secos presentes en la troposfera (N_2 , O_2 , Ar, etc).

b) Componente húmedo

Causado por el vapor y condensado del agua en forma de nubes.

El efecto de la troposfera se puede modelar mediante modelos categorizados entre según su orientación: geodésica o de navegación. Los modelos pertenecientes al grupo de orientación geodésica proporcionan resultados de mayor precisión pero requieren de información meteorológica. Los del segundo grupo presentan menor precisión pero no necesitan de datos meteorológicos.

2.4.4 Error en el reloj del receptor

El error del reloj del receptor se incluye como una incógnita en las ecuaciones de navegación, por lo que no es necesario modelarlo ni contrarrestarlo de ninguna otra manera ya que entra dentro de la optimización de los observables.

2.4.5 Ruido térmico

El ruido que se adhiere a la señal a lo largo de todo el proceso de transmisión y recepción es inevitable.

2.4.6 Multipath

El efecto de multipath puede aparecer en entornos urbanos cuando la señal rebota en edificios retrasando la recepción de la misma por parte del receptor. Con esto, el valor del observable de pseudorange es mayor que el real, y por tanto incorrecto. Este efecto es más frecuente cuando los satélites cuentan con poca elevación.

2.5 Cálculo posición y velocidad satélites

Con las posiciones de varios satélites (4 mínimo, como se explicó en un apartado anterior) y sus distancias al receptor, es posible obtener la posición de dicho receptor.

El cálculo de la posición y la velocidad de los satélites dependen de la constelación, específicamente del contenido de las efemérides contenida en los mensajes de navegación. Los cálculos para los sistemas de GPS y Galileo son prácticamente iguales, mientras que el procedimiento para GLONASS es completamente distinto.

Con el objetivo de no alargar excesivamente la memoria no se incluyen los cálculos ni las ecuaciones pertinentes, aunque están contenidos en el simulador que se utilizará en el trabajo. Existen varias técnicas de optimización para llevar a cabo su cálculo, siendo las más frecuentes la minimización por ajuste de mínimos cuadrados o el filtro de Kalman. Estas técnicas son estándar y se encuentran ampliamente recogidas en la literatura por lo que quedan fuera del alcance de la exposición de esta memoria.

2.6 Sistema de coordenadas

Para computar la posición de un receptor es conveniente utilizar un sistema de referencia que esté centrado en la Tierra. Además, resulta interesante que este sistema rote con la Tierra. El sistema ECEF (*Earth-Centered Earth-Fixed*) cumple estas condiciones. Por tanto, los ejes (x,y,z) del sistema de coordenadas no describen direcciones fijas en el espacio, si no que rotan con la Tierra.

En el sistema ECEF, el eje X apunta en la dirección de 0° de longitud, mientras que el eje Y apunta en la dirección de 90° de longitud. El eje Z es perpendicular al plano ecuatorial y apunta al polo norte geográfico.

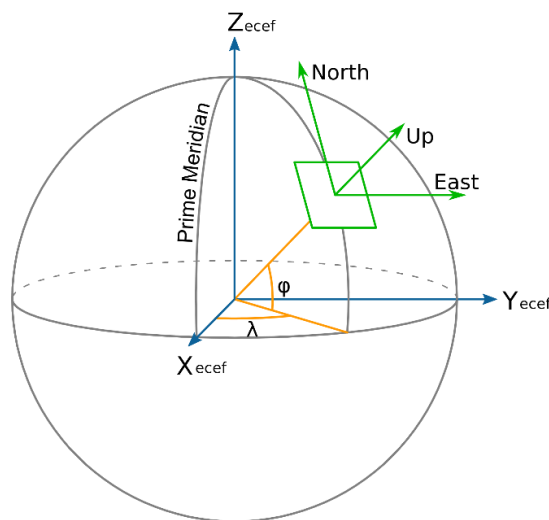


Figura 5: Sistema de coordenadas ECEF.

El cálculo de la posición de los receptores (x_u , y_u , z_u) se computa en el sistema ECEF.

CAPÍTULO 3: Detección de Saltos de Ciclo

En este capítulo se va a tratar la aparición de los llamados “saltos de ciclo” en los sistemas de navegación por satélite. En primer lugar se describirá en que consiste un salto de ciclo y en qué condiciones se da su ocurrencia. Seguidamente, se abordará la importancia de detectar estos saltos de ciclo y las consecuencias de no llevar a cabo dicha detección.

Para analizar la detección de los saltos se presentan 4 algoritmos distintos que persiguen localizar los saltos de maneras desiguales.

Por último se llevará a cabo una valoración de los resultados obtenidos mediante los algoritmos en cuestión. Además, se realizará una comparativa entre los algoritmos de forma cuantitativa y cualitativa.

3.1 Definición de Salto de Ciclo

El objetivo de la detección de los saltos de ciclo es evitar sus efectos sobre los sistemas de navegación. Resulta necesario conocer el origen de los saltos de ciclo y su manifestación en las medidas GNSS. Como se comentó en el capítulo anterior, existen 3 tipos de observables en la navegación por satélite: la pseudodistancia, la diferencia de fase y el observable Doppler. El observable de fase es una medida considerablemente precisa pero cuenta con una ambigüedad desconocida. Los saltos de ciclo afectan directamente a esta ambigüedad por lo que sus efectos son visibles en el observable de fase.

Cuando la señal que recibe la antena del receptor se ve interrumpida por cualquier motivo, el receptor pierde la cuenta del número de ciclo. Al reestablecerse la señal debe comenzar la cuenta, asignando un nuevo valor a la ambigüedad. Como se mencionó en el capítulo anterior, la fase de la portadora es una medida con mucha precisión que puede mitigar el ruido de las señales, pero que se ve afectada por una ambigüedad desconocida. La aparición de saltos de ciclo provoca la asignación de un nuevo valor a la ambigüedad, lo que impediría la mitigación del ruido.

La interrupción de la señal se puede deber a diversas causas:

- a) Pérdida de la señal enviada por el satélite debido a obstáculos (edificios, montañas, árboles, etc.)
- b) Baja calidad de la señal debido a factores físicos o baja elevación del satélite.
- c) Fallo de sincronismo.

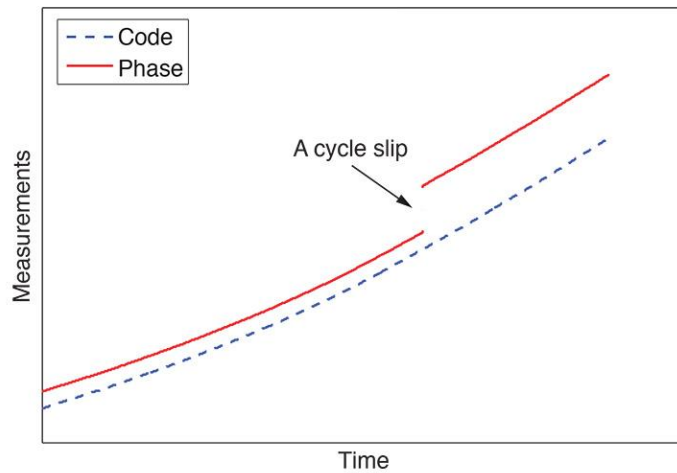


Figura 6: Presencia de un salto de ciclo en el observable de diferencia de fase.

La Figura 6 muestra cómo afecta la presencia de un salto de ciclo al observable de fase, en el que se produce un cambio brusco. Este salto en el observable provoca una pérdida de precisión en la solución de navegación. En definitiva, un salto de ciclo provoca variaciones en el offset de la ambigüedad.

3.2 Importancia de la Detección de los Saltos de Ciclo

Como se ha explicado en el apartado anterior, un salto de ciclo causa un escalón en la medida de fase. Si esta se usa para suavizar el pseudorange, se produce una pérdida de precisión en la posición proporcionada por el sistema de navegación. Esta pérdida de precisión es temporal, ya que el efecto desaparece porque típicamente los filtros de Hatch se hacen con una ventana temporal. Aun así, la pérdida de precisión puede no ser aceptable en ciertas aplicaciones en las que se requiere un grado de exactitud elevado.

Por tanto, resulta evidente la necesidad de detectar los saltos de ciclo. Habiendo detectado el salto, el sistema de navegación obviará la observación en la que ha detectado el salto y una serie de observaciones consiguientes. De esta forma se evita la pérdida de precisión en la solución de navegación.

3.3 Algoritmos de Detección de Saltos de Ciclo

La mayor parte de los receptores de la actualidad tienen la capacidad de recibir más de una banda de frecuencia. Esta característica resulta interesante por diversos motivos. Después de realizar la detección de saltos de ciclo, se combinarán ambas frecuencias en una única medida para eliminar el efecto de la ionosfera (que es dependiente de la frecuencia). Por otro lado, para deshacerse de efectos independientes de la frecuencia, se utilizan combinaciones de observables para suprimir estos errores.

Para este trabajo se han considerado dos algoritmos de saltos de ciclo que funcionan con una única frecuencia y otros dos algoritmos que utilizan 2 frecuencias para la detección. A continuación, se describen dichos algoritmos.

3.3.1 Ajuste polinómico con frecuencia simple

Tanto este algoritmo como el que se describirá a continuación, utilizan la combinación observable de fase-observable de rango para la detección de los saltos de ciclo. Los errores no dependientes de la frecuencia se cancelan gracias a la combinación mencionada, resultando en:

Ecuación 3.1

$$d = \phi - R = \lambda \cdot N - 2 \cdot I + K + \varepsilon$$

Siendo, λ la longitud de onda, N la ambigüedad, I el efecto que introduce la ionosfera, K el error instrumental y ε el ruido.

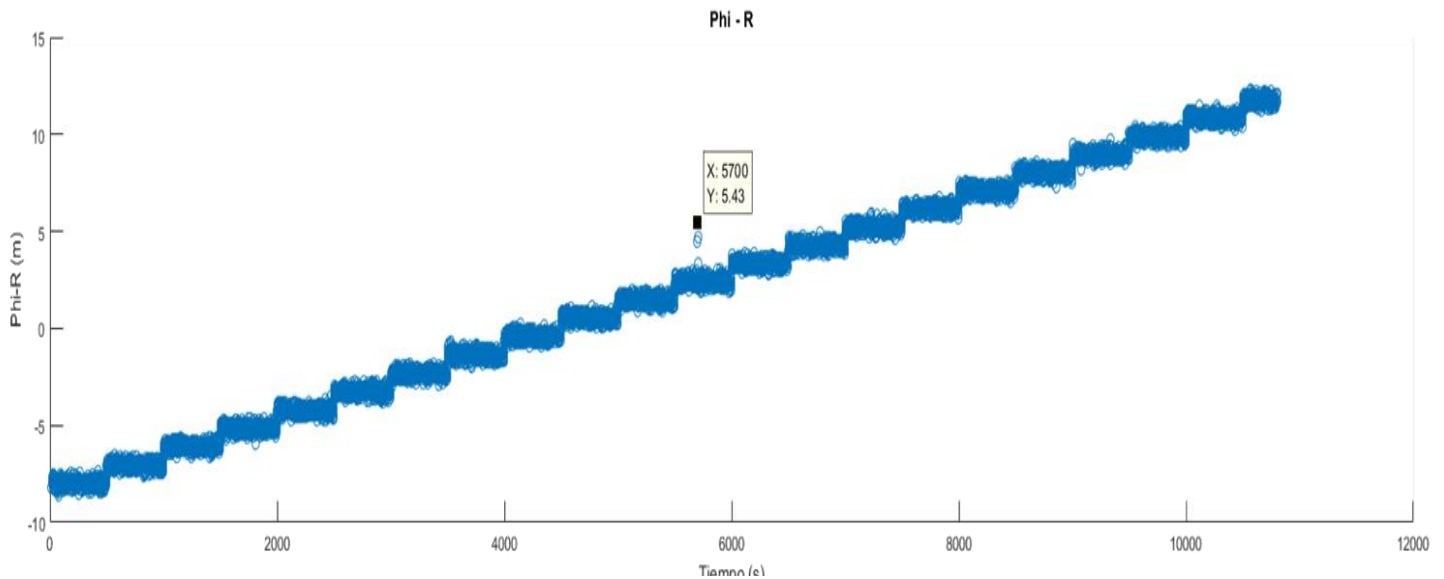


Figura 7: Combinación fase-rango para una sola frecuencia. Saltos de ciclo de cinco longitudes de onda.

La Figura 7 muestra la combinación fase rango obtenida a partir de los observables de las medidas utilizadas. Se pueden apreciar los distintos saltos de ciclo además de variaciones en la ionosfera (por ejemplo: variaciones rápidas de la ionosfera) en el segundo 5700.

La detección con este primer algoritmo se basa en el ajuste polinómico de una ventana de n muestras de la combinación fase-rango. Se compara la muestra actual con el valor que suministra el polinomio para la muestra $n+1$. Si la diferencia entre estos dos valores es superior a un umbral predefinido, se determina que ha ocurrido un salto de ciclo.

A continuación se describe el algoritmo paso a paso para k muestras de un satélite:

- Si existiese una ausencia de muestras considerable (2-3 segundos) se declara la presencia de un salto de ciclo.
- Si no es así:
 - Utilizando las n muestras anteriores a k ($[d(k-N), \dots, d(k-1)]$), se ajusta un polinomio de orden 2 ($P(x)$).
 - Si el valor absoluto de la diferencia entre $P(k)$ y $d(k)$ es mayor que el umbral definido, se declara la presencia de un salto de ciclo en k .

$$\text{if } (|d(k) - P(k)|) > \text{Umbral} \rightarrow \text{Salto de ciclo en } k$$

3.3.2 Media y desviación con frecuencia simple

Este algoritmo también hace uso de la combinación fase-rango descrita en el apartado anterior.

Para detectar saltos de ciclo, el algoritmo computa la media y la desviación de una ventana de n muestras de la combinación d . Para cada muestra, se calcula la media y la desviación de las n muestras anteriores.

Si el valor absoluto de la diferencia entre el valor actual de la combinación $d(k)$ y el valor medio de las n muestras anteriores $Md(k)$ es mayor que el valor de la desviación $Sd(k)$ escalado mediante una constante η ; se declara la presencia de un salto de ciclo en k .

A continuación se describe el algoritmo paso a paso para k muestras de un satélite:

- Si existiese una ausencia de muestras considerable (2-3 segundos) se declara la presencia de un salto de ciclo.
- Si no es así:
 - Utilizando las n muestras anteriores a k ($[d(k-N), \dots, d(k-1)]$), se calculan:

Ecuación 3.2

$$\begin{aligned} Md(k) &= \frac{1}{N} \cdot \sum_{i=1}^N d(k-i) \\ Md^2(k) &= \frac{1}{N} \cdot \sum_{i=1}^N d(k-i)^2 \\ Sd(k) &= \sqrt{Md^2(k) - Md(k)^2} \end{aligned}$$

- Si el valor absoluto de la diferencia entre $d(k)$ y $Md(k)$ es mayor que $\eta \cdot Sd(k)$, se declara la presencia de un salto de ciclo en k .

$$\text{if } (|d(k) - Md(k)| > \eta \cdot Sd(k)) \rightarrow \text{Salto de ciclo en } k$$

3.3.3 Combinación libre de geometría con frecuencia múltiple

En este caso, se hace uso de la combinación del observable de fase a dos frecuencias distintas (L1, L2) para detectar los saltos de ciclo. Se hace referencia a L1 (19 cm de longitud de onda) y L2 (24.4 cm de longitud de onda) ya que se va a utilizar una señal GPS en las simulaciones.

Ecuación 3.3

$$\varnothing = \varnothing_{L1} - \varnothing_{L2}$$

Esta combinación permite librarse de efectos indeseados en la señal. Se eliminan los errores de los relojes y los no dependientes en la frecuencia.

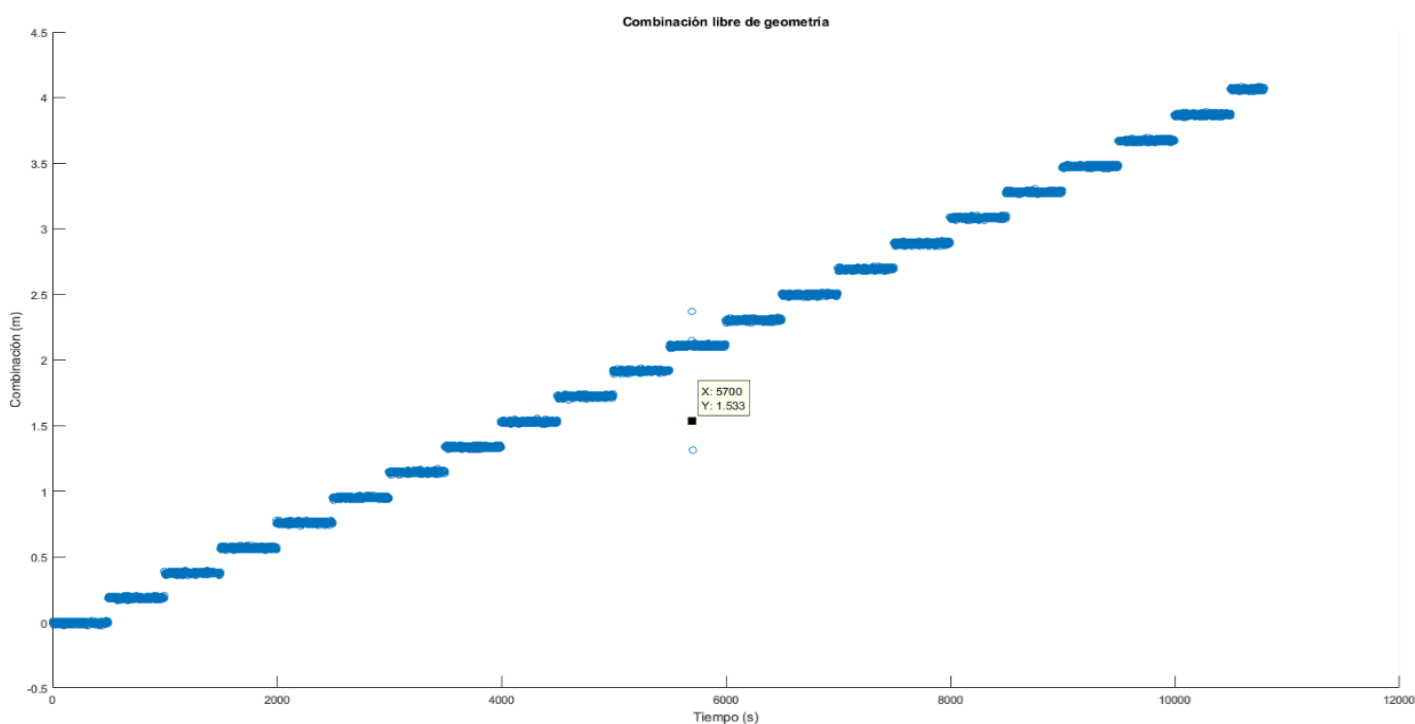


Figura 8: Combinación libre de geometría. Saltos de ciclo de una longitud de onda.

La combinación libre de geometría se muestra en la Figura 8. Los saltos de ciclo se diferencian de forma más clara que en la combinación fase-rango. De la misma forma que en la combinación fase-rango, se aprecia el efecto introducido por las variaciones en la ionosfera en el segundo 5700.

Además, se pretende amplificar el efecto del salto de ciclo en la señal recibida utilizando un procedimiento de n diferencias ($n=4$). Como se puede observar en la Figura 9, gracias a este procedimiento se obtiene una amplificación del error de hasta 3 veces.

t_i	$y(t_i)$	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
t_1	0				
		0			
t_2	0		0		
		0		ϵ	
t_3	0		ϵ		-3ϵ
		ϵ		-2ϵ	
t_4	ϵ		$-\epsilon$		3ϵ
		0		ϵ	
t_5	ϵ		0		$-\epsilon$
		0		0	
t_6	ϵ		0		
		0			
t_7	ϵ				

Figura 9: Procedimiento de 4 diferencias.

Este procedimiento hace posible la detección de saltos de ciclo de diversos tamaños. Es importante indicar que el procedimiento de diferencias trae consigo también la amplificación del ruido, lo que condicionará sin duda el desempeño del algoritmo.

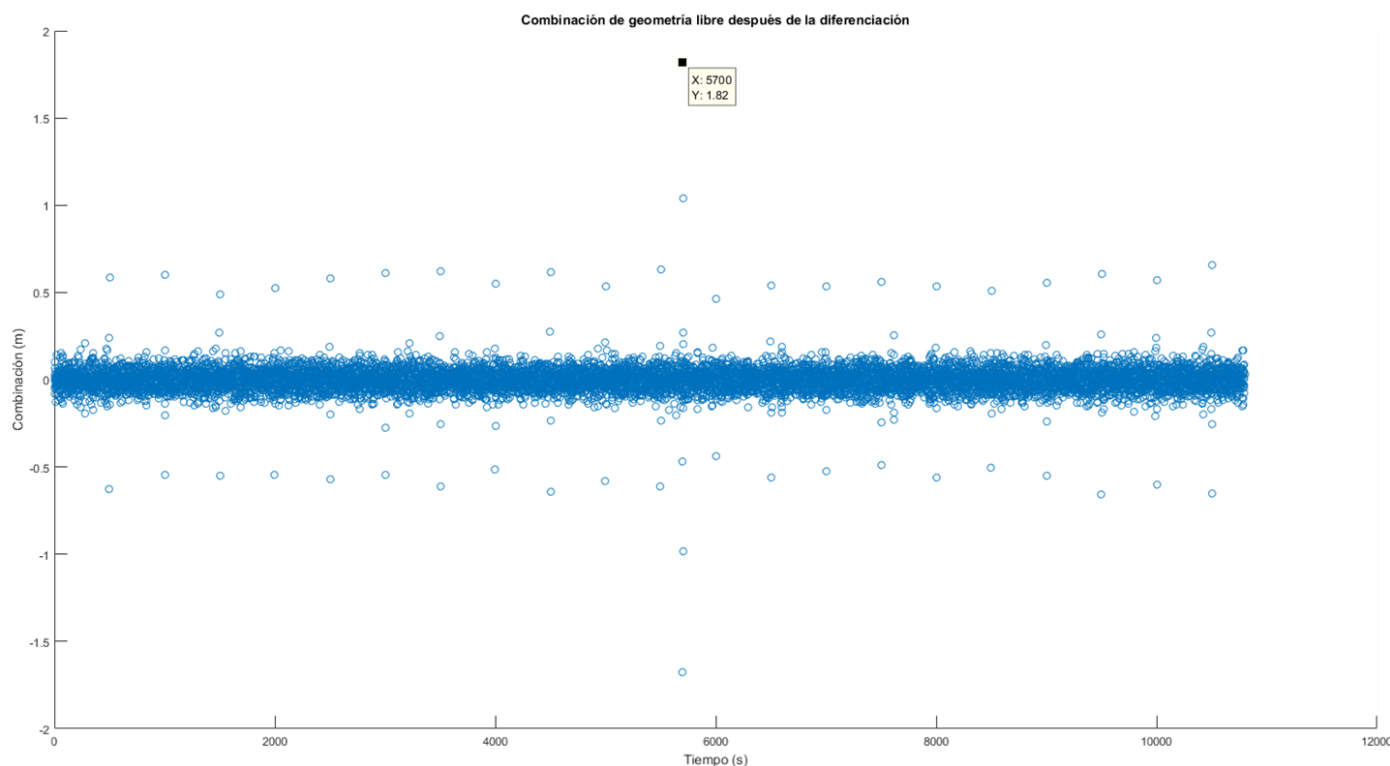


Figura 10: Combinación libre de geometría diferenciando las medidas. Saltos de ciclo de una longitud de onda.

En la Figura 10 se puede observar la combinación libre de geometría después de aplicar el procedimiento de las diferencias comentado. Se puede apreciar claramente las muestras con saltos de ciclo. También es evidente que el efecto por las variaciones en la ionosfera es más pronunciado. Por último, el ruido presente en las medidas también se ve intensificado, lo que afectará de forma negativa al desempeño del algoritmo.

Después de realizar la diferenciación, se lleva a cabo un ajuste polinómico similar al realizado en el algoritmo de la sección 1.4.1. El valor que toma el polinomio en k se compara con el de la combinación $\emptyset(k)$. Si el valor absoluto de esta diferencia es mayor a un umbral previamente definido, se declara la presencia de un salto de ciclo.

El valor del umbral se define de la forma siguiente:

Ecuación 3.4

$$Umbral = a_0 - a_1 \cdot e^{-\frac{\Delta t}{T_0}}$$

Siendo,

Ecuación 3.5

$$a_0 = \frac{3}{2} \cdot (\lambda_1 - \lambda_2)$$

$$a_1 = \frac{a_0}{2}$$

Δt es el tiempo de reposo del algoritmo después de detectar un salto de ciclo.

T_0 es el tiempo de decorrelación de la ionosfera. Se suele fijar en 60 segundos.

A continuación se describe el algoritmo paso a paso para k muestras de un satélite:

- Si existiese una ausencia de muestras considerable (2-3 segundos) se declara la presencia de un salto de ciclo.
- Si no es así:
 - Utilizando las n muestras anteriores a k ($[\emptyset(k-N), \dots, \emptyset(k-1)]$), se ajusta un polinomio de orden 2 ($P(x)$).
 - Si el valor absoluto de la diferencia entre $P(k)$ y $\emptyset(k)$ es mayor que el umbral definido, se declara la presencia de un salto de ciclo en k .

$$if (|\emptyset(k) - P(k)|) > Umbral \rightarrow \text{Salto de ciclo en } k$$

3.3.4 Combinación Melbourne-Wübbena

La combinación Melbourne-Wübbena sigue la siguiente ecuación:

Ecuación 3.6

$$Bw = \phi_w - R_N$$

Siendo,

Ecuación 3.7

$$\phi_w = \frac{f_1 \cdot \phi_1 - f_2 \cdot \phi_2}{f_1 - f_2}$$

Ecuación 3.8

$$R_N = \frac{f_1 \cdot R_1 + f_2 \cdot R_2}{f_1 + f_2}$$

Donde los subíndices 1,2 hacen referencia a las dos frecuencias utilizadas en este caso.

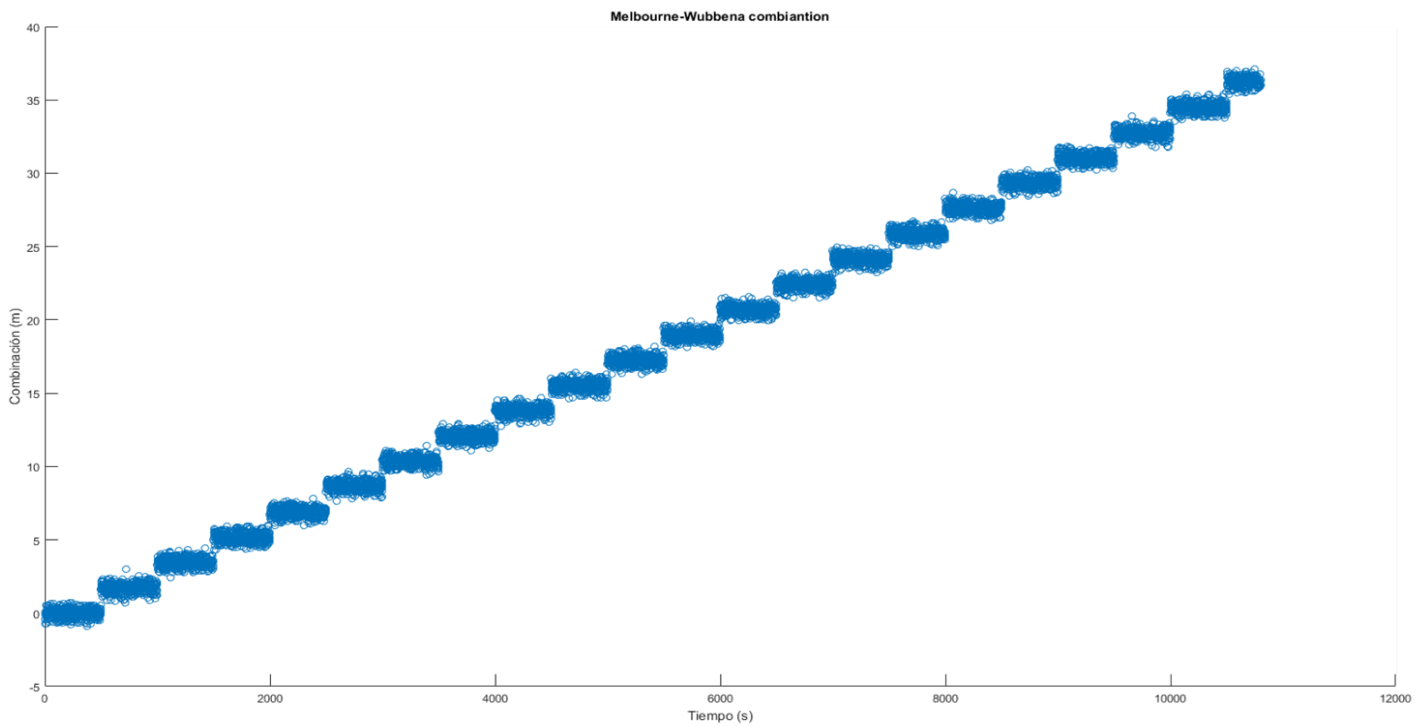


Figura 11: Combinación Melbourne-Wübbena. Saltos de ciclo de dos longitudes de onda.

La Figura 11 muestra la combinación Melbourne-Wübbena. Es posible apreciar los saltos de ciclo introducidos en las muestras. Se debe destacar, que en este caso, el efecto introducido por las variaciones en la ionosfera no está presente

en la combinación. Por ello, este algoritmo no declarará erróneamente un salto de ciclo debido a la ionosfera.

De forma similar al algoritmo descrito en la sección 1.3.2, para cada muestra k se calcula la media y la desviación de la muestras anteriores desde el último salto de ciclo detectado (en este caso no se utiliza un ventana de n muestras).

Si el valor absoluto de la diferencia entre $B_w(k)$ y la media obtenida $m_{B_w}(k)$ es mayor la desviación obtenida $S_{B_w}(k)$ escalado con un factor $K=4$, se declara la presencia de un salto de ciclo.

A continuación se describe el algoritmo paso a paso para k muestras de un satélite:

- Si existiese una ausencia de muestras considerable (2-3 segundos) se declara la presencia de un salto de ciclo.
- Si no es así:
 - Si el valor absoluto de la diferencia entre $B_w(k)$ y $m_{B_w}(k)$ es mayor que $K \cdot S_{B_w}(k)$, se declara la presencia de un salto de ciclo en k .

$$\text{if } (|B_w(k) - m_{B_w}(k)| > \eta \cdot K \cdot S_{B_w}(k) \rightarrow \text{Salto de ciclo en } k$$

- Utilizando las n muestras anteriores a k ($[d(k-N), \dots, d(k-1)]$), se calculan:

Ecuación 3.9

$$m_{B_w}(k) = \frac{k-1}{k} \cdot m_{B_w}(k-1) + \frac{1}{k} \cdot B_w(k)$$

$$S_{B_w}^2(k) = \frac{k-1}{k} \cdot S_{B_w}^2(k-1) + \frac{1}{k} \cdot (B_w(k) - m_{B_w}(k-1))^2$$

3.4 Comparativa y Valoración de los Algoritmos

3.4.1 Obtención de datos

Para poner a prueba el desempeño de los algoritmos descritos en el apartado anterior, se partió de un vector de datos obtenido a partir de un archivo RINEX de un satélite GPS.

Gracias al archivo RINEX, se obtuvo la posición del dicho satélite en el espacio en un determinado momento. Se calculó su posición durante las siguientes 3 horas para cada segundo. Obteniendo finalmente, la distancia entre un receptor situado en la superficie terrestre y el satélite para 10800 segundos.

Para llevar a cabo las operaciones mencionadas, se programaron dichas funcionalidades utilizando el lenguaje C++. Se desarrollaron dos módulos que contienen una serie de funciones para la extracción de datos del archivo RINEX y para el cálculo de la distancia.

Los archivos RINEX están compuestos en un formato estandarizado y contienen la información necesaria para obtener la posición y velocidad de satélites. El formato del archivo varía según el sistema de navegación, es decir, la información presentada en un RINEX de GPS no será la misma que la que se puede encontrar en un RINEX de GLONASS.

La configuración típica de un archivo RINEX de muestra en la Figura 12. El archivo consta en primer lugar de un encabezado que contiene información general de la constelación de satélites. A continuación aparece la información individual de cada satélite comprendido en la constelación.

<pre> 2 NAVIGATION DATA RINEX VERSION / TYPE CCRINEXN V1.6.0 UX CDDIS 08-JAN-09 10:54 PGM / RUN BY / DATE IGS BROADCAST EPHEMERIS FILE COMMENT 0.7451E-08 -0.1490E-07 -0.5960E-07 0.1192E-06 ION ALPHA 0.9011E+05 -0.6554E+05 -0.1311E+06 0.4588E+06 ION BETA -0.372529029846E-08-0.444089209850E-14 589824 1513 DELTA-UTC: A0,A1,T,W 14 LEAP SECONDS END OF HEADER </pre>	Encabezado
<pre> 2 09 1 8 0 0 0.0 0.162754207849E-03-0.193267624127E-11 0.000000000000E+00 0.550000000000E+02 0.430000000000E+02 0.491484758027E-08 0.161220258749E+01 0.219047069550E-05 0.890624953899E-02 0.984035432339E-05 0.515360116005E+04 0.345600000000E+06 0.109896063805E-06 0.209825615380E+01 0.521540641785E-07 0.941753737355E+00 0.178156250000E+03 0.269590463683E+01-0.797211778499E-08 -0.670385067086E-09 0.100000000000E+01 0.151300000000E+04 0.000000000000E+00 0.200000000000E+01 0.000000000000E+00-0.172294676304E-07 0.550000000000E+02 0.338418000000E+06 0.400000000000E+01 0.000000000000E+00 0.000000000000E+00 </pre>	Satélite 1
<pre> 3 09 1 8 0 0 0.0 0.335775781423E-03 0.534328137292E-11 0.000000000000E+00 0.610000000000E+02 0.156250000000E+01 0.583667159049E-08 0.126460770886E+01 0.726431608200E-07 0.115746114170E-01 0.579841434956E-05 0.515371566200E+04 0.345600000000E+06-0.165775418282E-06 0.951693665907E+00 0.134110450745E-06 0.925898792133E+00 0.251437500000E+03 0.887259589786E+00-0.871500560606E-08 0.635740765587E-10 0.100000000000E+01 0.151300000000E+04 0.000000000000E+00 0.280000000000E+01 0.000000000000E+00-0.419095158577E-08 0.610000000000E+02 0.338418000000E+06 0.400000000000E+01 0.000000000000E+00 0.000000000000E+00 </pre>	Satélite 2

Figura 12: Archivo RINEX GPS.

En el encabezado encontramos, en primer lugar, la versión del formato y el tipo de archivo. A continuación se encuentra información acerca de la creación del archivo en sí. El archivo RINEX contiene información para facilitar el cálculo del efecto ionosférico además de los parámetros necesarios para calcular el tiempo UTC (tiempo universal coordinado).

En cuanto a la información referida a cada satélite particular, se presentan una serie de parámetros físicos que permiten el cálculo de la posición y velocidad del satélite en cuestión.

Mediante la información obtenida de los archivos RINEX se han calculado dos de los observables que son de interés para la detección de los saltos de ciclo y que se han explicado en la sección 2.3: el pseudorango y la diferencia de fase.

3.4.2 Ruido y errores de medida

Resulta necesario añadir los efectos del ruido y de la ionosfera a estos datos antes de poner en práctica los algoritmos. El ruido presente en los sistemas de navegación proviene de distintas fuentes: el equipamiento en sí mismo y de fuentes naturales.

El ruido térmico es la forma más básica de ruido y se produce por el movimiento aleatorio de electrones en cualquier conductor con una temperatura superior al cero absoluto (0°K ó -273°C). Se suele denominar a este ruido como “ruido blanco”.

La antena de un receptor de navegación por satélite recoge cierto ruido en forma de radiación electromagnética producida de forma natural. Esta radiación proviene del cielo, el suelo y cualquier objeto en las cercanías de la antena.

El otro componente del ruido del sistema es el que surge como combinación de las pérdidas del cableado e internamente generado por el receptor. Se puede aproximar que la aportación del ruido en la medida del rango es de 33 centímetros y en de 0.2 milímetros en la medida de fase.

Para la simulación llevada a cabo en este trabajo, se va suponer que el ruido sigue una distribución normal de media nula; y desviación típica 0.02 veces la longitud de onda para el observable de fase y 2 veces la longitud de onda para el observable de rango.

Por otro lado, se debe tener en cuenta también los efectos de la ionosfera sobre las señales de navegación. La velocidad de propagación de las señales electromagnéticas en la ionosfera depende de la densidad de electrones. La ionosfera es un medio dispersivo, lo que significa que la propagación por el

mismo depende de la frecuencia. Gracias a esta característica, será posible eliminar el efecto ionosférico en los casos en los que contemos con más de una frecuencia.

Se va tener en cuenta también la presencia del fenómeno conocido como “centelleo de ionosfera”, como ejemplo de variación en la ionosfera. El centelleo de la ionosfera ocurre debido a irregularidades en la densidad y distribución de los electrones en el trayecto que realiza la señal. Estas irregularidades pueden estar presentes en estructuras de hasta decenas de kilómetros.

Para tener en cuenta el efecto de la ionosfera en la señales, se va a añadir una componente de 4 metros que varíe 0.1 metros por segundo con un pequeño rizado. Esta componente se restará en caso del observable de fase y se sumará en caso del observable de rango.

Al ser el centelleo de la ionosfera un fenómeno temporal, este se incorporará durante 4 segundos con una amplitud de 0.6 metros.

3.4.3 Saltos de ciclo

Se añadirán saltos de ciclo artificiales a las medidas con los errores mencionados en el apartado anterior. Estos saltos se agregarán a los observables de fase en forma de múltiplos de la longitud de onda.

Para poder observar la reacción de los distintos algoritmos a los saltos de ciclo, se realizarán pruebas implantando en las medidas saltos de ciclo de distintos tamaños.

La Tabla 1 muestra los distintos saltos de ciclo aplicados los observables de fase según su frecuencia.

<i>Múltiplo longitud de onda</i>	L1 (1575.42 MHz/19 cm)	L2 (1227.6 MHz/24.4 cm)
<i>1</i>	19 cm	24.4 cm
<i>2</i>	38 cm	48.8 cm
<i>5</i>	95 cm	122 cm
<i>10</i>	190 cm	244 cm

Tabla 1: Clasificación saltos de ciclo aplicados según la frecuencia.

3.4.4 Resultados

Las simulaciones correspondientes se llevaron a cabo mediante la herramienta matemática *Matlab*. Debido al uso de la función generadora de números aleatorios de tipo normal, se realizó cada caso de simulación en bucle para posteriormente calcular la media de los resultados.

Se obtendrán dos indicadores de las simulaciones: probabilidad de detección de saltos de ciclo e inmunidad frente al ruido.

El grado de detección de saltos de ciclo se refiere al porcentaje de los saltos de ciclo introducidos en las medidas que el algoritmo en cuestión es capaz de detectar. Se repartirán 21 saltos de ciclo colocados cada 500 muestras.

Ecuación 3.10

$$\begin{aligned} &\text{Grado detección saltos de ciclo (\%)} \\ &= \frac{N^{\circ} \text{ saltos de ciclo detectados correctamente}}{21} \cdot 100 \end{aligned}$$

La inmunidad frente al ruido hace referencia a la cantidad de saltos de ciclo inexistentes que el algoritmo detecta debido a los efectos del ruido. Un algoritmo con baja inmunidad frente al ruido provocará que se rechacen medidas recibidas de los satélites incorrectamente. La inmunidad frente al ruido se medirá como el porcentaje de falsas detecciones indicadas por el algoritmo de entre las medidas disponibles.

Ecuación 3.11

$$\begin{aligned} &\text{Inmunidad frente al ruido (\%)} \\ &= \frac{N^{\circ} \text{ falsos saltos de ciclo}}{10800 - (21 * (\text{Ventana muestras} + \text{Muestras rechazadas tras detección}))} \cdot 100 \end{aligned}$$

Fundamentalmente, el grado de detección de saltos de ciclo es el error por no detectar un salto de ciclo. Mientras que la inmunidad frente al ruido es el error por detectar un salto de ciclo inexistente.

Para poder observar el desempeño de cada algoritmo en relación al nivel de ruido introducido en la simulación, se llevarán a cabo 3 simulaciones para cada algoritmo y tamaño de salto:

- 1) Sin ruido.
- 2) Con el término del ruido reducido a la mitad.
- 3) Con el término del ruido.

A continuación se presentan los resultados derivados de las simulaciones, así como una serie de gráficas para intentar ilustrar de forma más clara los resultados obtenidos.

3.4.4.1 Ajuste polinómico con frecuencia simple

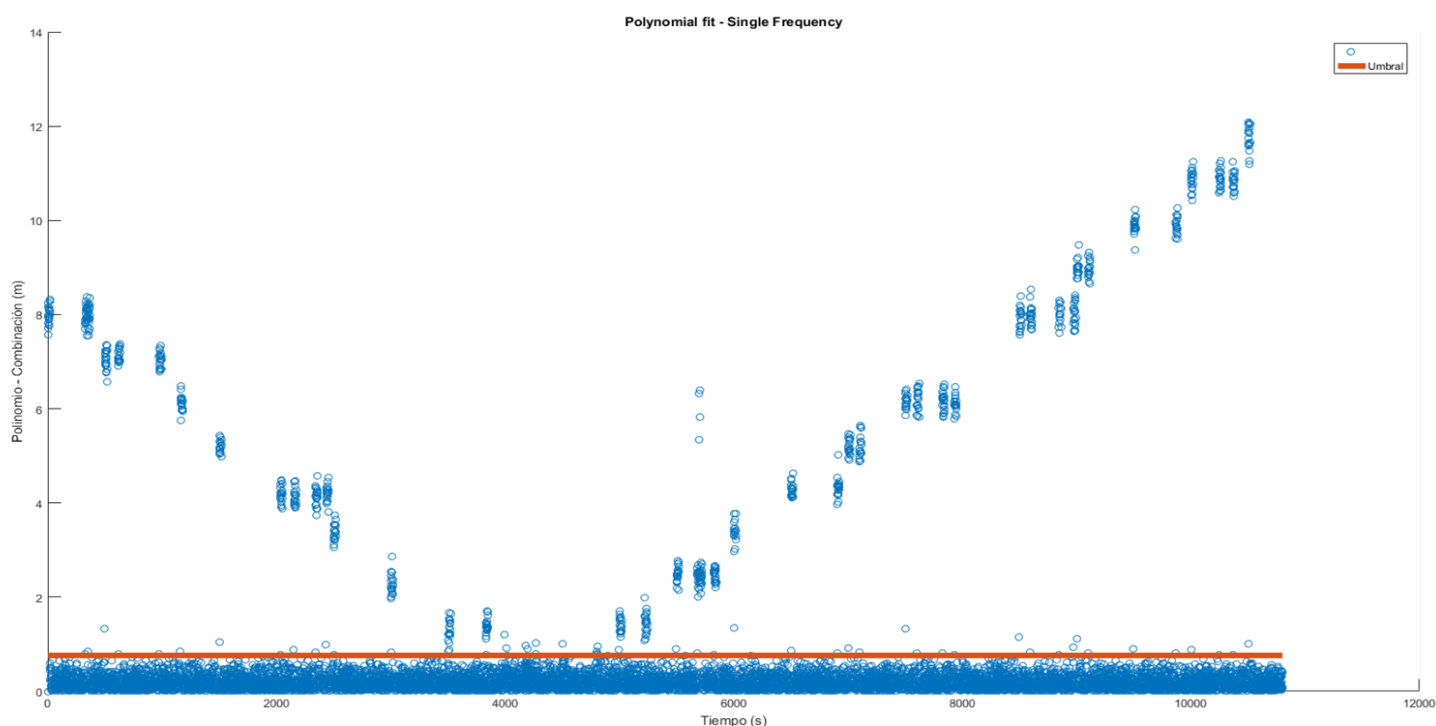


Figura 13: Detección saltos de ciclo para una única frecuencia utilizando ajuste polinómico. Saltos de ciclo de cinco longitudes de onda.

En la Figura 13 se puede observar la detección de salto de ciclo mediante un ajuste polinómico para una sola frecuencia. Para la simulación de la figura, se implantaron saltos de ciclo de tamaño 5 veces la longitud de onda y se utilizó un umbral de 4 veces la longitud de onda. Cuando la diferencia entre el valor

obtenido por el polinomio y la combinación fase-rango queda por encima del umbral establecido, se declara salto de ciclo.

Se puede comprobar, que en ciertos casos el ruido provoca una detección errónea. Además, se puede observar la presencia del centelleo ionosférico en el segundo 5700.

Tamaño saltos de ciclo	Umbral	Grado ruido	Nº saltos detectados	Grado de detección	Nº falsas detecciones	Inmunidad frente al ruido
1	1*Lambda	0	9.50	45.24	1.00	0.19
		0.5	1.00	4.76	4.85	0.93
		1	0.80	3.81	502.50	96.82
2	1*Lambda	0	21.00	100.00	1.00	0.19
		0.5	1.60	7.62	484.90	93.43
		1	0.70	3.33	503.40	96.99
	2*Lambda	0	10.30	49.05	1.00	0.19
		0.5	2.50	11.90	387.40	74.64
		1	1.00	4.76	483.90	93.24
5	2*Lambda	0	21.00	100.00	1.00	0.19
		0.5	5.90	28.10	387.60	74.68
		1	1.30	6.19	484.90	93.43
	4*Lambda	0	21.00	100.00	1.00	0.19
		0.5	15.60	74.29	25.20	4.86
		1	3.70	17.62	388.80	74.91
10	4*Lambda	0	21.00	100.00	1.00	0.19
		0.5	19.70	93.81	27.70	5.34
		1	5.90	28.10	385.60	74.30
	8*Lambda	0	21.00	100.00	1.00	0.19
		0.5	19.20	91.43	1.00	0.19
		1	14.90	70.95	28.90	5.57

Tabla 2: Indicadores detección salto de ciclo. Algoritmo de ajuste polinómico con frecuencia simple.

Como se puede observar en la Tabla 2, el algoritmo de ajuste polinómico para una sola frecuencia se ve considerablemente afectado por el nivel de ruido y el umbral utilizado para la detección.

También es evidente que al aumentar el tamaño del salto de ciclo el desempeño del algoritmo en cuestión mejora. De esta forma, para saltos de ciclo de 5 o 10 veces la longitud de onda el algoritmo detecta una gran mayoría de los saltos incluidos en las medidas. Además, presenta una mejor inmunidad frente al ruido. Para los saltos de ciclo de menor tamaño, el grado de detección es menor y la detección falsa de saltos, mayor.

Para lograr los resultados mostrados en la Tabla 2, se ha tenido que ir variando el umbral de detección según el tamaño del salto. Sin esta variación, los indicadores ofrecerían resultados más pobres. Además, en un caso de uso real sería muy complicado estimar a priori el nivel del umbral.

3.4.4.2 Media y desviación con frecuencia simple

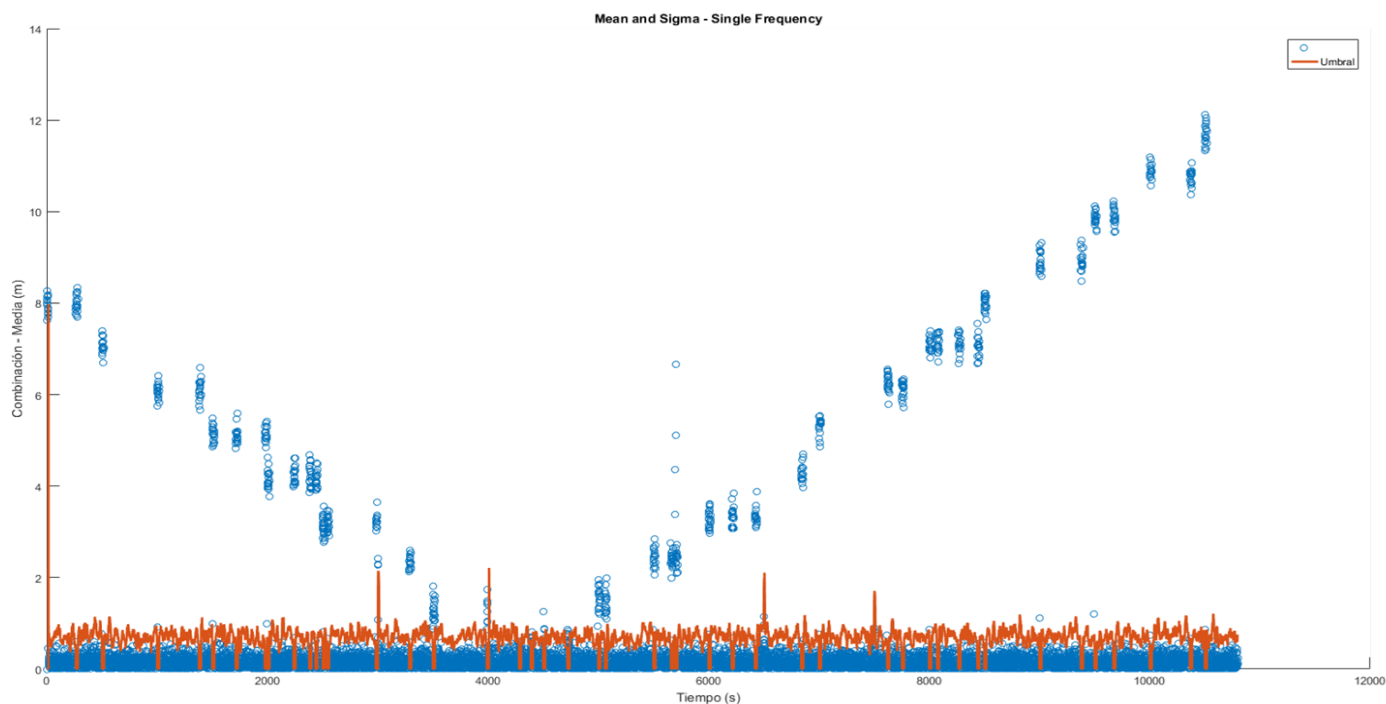


Figura 14: Detección saltos de ciclo para una única frecuencia utilizando media y desviación. Saltos de ciclo de cinco longitudes de onda.

La Figura 14 muestra la detección de saltos de ciclo utilizando la media y la desviación con una sola frecuencia. Para la simulación de la figura, se implantaron saltos de ciclo de tamaño 5 veces la longitud de onda y se utilizó un umbral de 4 veces la desviación típica de la combinación fase-rango.

Se puede apreciar claramente el efecto del centelleo ionosférico en el segundo 5700. Además, debido al ruido ciertas muestras se encuentran por encima del umbral de detección lo que podría dar lugar a detecciones falsas.

Tamaño saltos de ciclo	Umbral	Grado ruido	Nº saltos detectados	Grado de detección	Nº falsas detecciones	Inmunidad frente al ruido
1	2*Sd	0	7.60	36.19	346.00	66.67
		0.5	1.60	7.62	350.00	67.44
		1	1.20	5.71	352.70	67.96
2	2*Sd	0	7.10	33.81	348.30	67.11
		0.5	4.10	19.52	351.50	67.73
		1	1.80	8.57	350.90	67.61
	4*Sd	0	20.00	95.24	23.10	4.45
		0.5	1.00	4.76	22.00	4.24
		1	0.10	0.48	22.50	4.34
5	4*Sd	0	20.10	95.71	23.20	4.47
		0.5	16.60	79.05	24.50	4.72
		1	2.30	10.95	23.40	4.51
	6*Sd	0	21.00	100.00	1.50	0.29
		0.5	7.40	35.24	2.20	0.42
		1	0.50	2.38	1.70	0.33
10	6*Sd	0	21.00	100.00	2.20	0.42
		0.5	20.70	98.57	1.80	0.35
		1	7.00	33.33	1.50	0.29
	8*Sd	0	21.00	100.00	1.00	0.19
		0.5	19.80	94.29	1.00	0.19
		1	1.70	8.10	0.80	0.15

Tabla 3: Indicadores detección salto de ciclo. Algoritmo de media y desviación con frecuencia simple.

En la Tabla 3 se puede observar como el algoritmo que utiliza la media y la variación con una sola frecuencia ofrece resultados similares al algoritmo anterior en cuanto a grado de detección. El desempeño del algoritmo se ve también afectado por el ruido y el tamaño del salto de ciclo.

Por otro lado, la inmunidad frente al ruido mejora considerablemente respecto al algoritmo anterior. Excepto para saltos de ciclo con una longitud de onda de tamaño, el algoritmo es capaz de ofrecer menos de 5% de detecciones falsas.

Por último, la dependencia de este algoritmo en la variación del umbral de detección también es importante. Ambos algoritmos que utilizan únicamente una frecuencia para la detección requieren de la modificación del umbral para lograr la detección de saltos de diferentes tamaños.

3.4.4.3 Combinación de geometría libre con frecuencia múltiple

Tamaño saltos de ciclo L2	Tamaño saltos de ciclo L1	Grado ruido	Nº saltos detectados	Grado de detección	Nº falsas detecciones	Inmunidad frente al ruido
0	1	0	7.80	37.14	330.30	63.64
		0.5	3.40	16.19	439.40	84.66
		1	2.40	11.43	484.30	93.31
	2	0	6.90	32.86	331.90	63.95
		0.5	3.20	15.24	440.70	84.91
		1	2.40	11.43	484.60	93.37
	5	0	8.50	40.48	334.90	64.53
		0.5	4.80	22.86	439.60	84.70
		1	2.10	10.00	484.20	93.29
	10	0	7.80	37.14	330.80	63.74
		0.5	4.20	20.00	437.40	84.28
		1	2.70	12.86	483.90	93.24
1	1	0	6.10	29.05	328.70	63.33
		0.5	2.30	10.95	439.70	84.72
		1	0.80	3.81	485.40	93.53
	2	0	8.30	39.52	333.80	64.32
		0.5	2.90	13.81	441.00	84.97
		1	2.10	10.00	448.70	86.45
	5	0	8.10	38.57	333.40	64.24
		0.5	4.10	19.52	441.00	84.97
		1	1.70	8.10	483.80	93.22
	10	0	7.70	36.67	329.50	63.49
		0.5	4.50	21.43	438.00	84.39
		1	1.7	8.10	485.10	93.47
2	2	0	7.2	34.29	335.70	64.68
		0.5	3.3	15.71	440.20	84.82
		1	1.5	7.14	485.50	93.55
	5	0	6.8	32.38	333.70	64.30
		0.5	2.9	13.81	440.50	84.87
		1	2.1	10.00	485.10	93.47
	10	0	7.4	35.24	331.90	63.95
		0.5	4.5	21.43	441.40	85.05
		1	1.6	7.62	484.40	93.33
5	5	0	7.6	36.19	331.70	63.91
		0.5	5	23.81	438.10	84.41
		1	2.3	10.95	484.00	93.26
	10	0	8.2	39.05	333.20	64.20
		0.5	3.7	17.62	442.10	85.18
		1	1.8	8.57	483.90	93.24
10	10	0	7.8	37.14	331.50	63.87
		0.5	3.4	16.19	442.10	85.18
		1	2.7	12.86	483.90	93.24

Tabla 4: Indicadores detección salto de ciclo. Algoritmo de combinación de geometría libre con frecuencia múltiple.

Los resultados mostrados en la Tabla 4 corresponden al algoritmo que utiliza la combinación de geometría libre con dos frecuencias para realizar la detección. Es evidente que los resultados son nefastos e incluso peores que los presentados por los algoritmos que utilizan una sola frecuencia.

El algoritmo es solamente capaz de ofrecer un grado de detección aceptable sin la presencia de ruido. La inmunidad frente al ruido resultante es inferior a la aceptable. El número de falsas detecciones es desmedido y puede ser la causa de que el grado de detección sea tan reducido.

Debido a los resultados pobres obtenidos con la implementación de este algoritmo, se han propuesto dos modificaciones para mejorar su desempeño. En primer lugar se incrementará el umbral de detección con el objetivo de reducir el número de falsas detecciones. Por otro lado se eliminará la diferenciación de la combinación de libre geometría del algoritmo. De esta forma se conseguirá un ruido menor y se reducirá el número de falsas detecciones.

Con estas dos variaciones se persigue reducir el número de falsas detecciones y por tanto se aumenta la posibilidad de que el algoritmo detecte los saltos de ciclo añadidos a las medidas.

Tamaño saltos de ciclo L2	Tamaño saltos de ciclo L1	Grado ruido	Nº saltos detectados	Grado de detección	Nº falsas detecciones	Inmunidad frente al ruido
0	1	0	19.90	94.76	18.60	3.58
		0.5	12.40	59.05	209.00	40.27
		1	4.70	22.38	404.60	77.96
	2	0	20.30	96.67	19.90	3.83
		0.5	11.60	55.24	206.90	39.87
		1	5.30	25.24	407.30	78.48
	5	0	20.50	97.62	21.50	4.14
		0.5	13.00	61.90	211.50	40.75
		1	5.50	26.19	405.70	78.17
	10	0	20.50	97.62	21.40	4.12
		0.5	12.60	60.00	215.50	41.52
		1	4.80	22.86	405.20	78.07
1	1	0	2.20	10.48	40.90	7.88
		0.5	2.50	11.90	225.10	43.37
		1	1.00	4.76	407.30	78.48
	2	0	18.90	90.00	23.10	4.45
		0.5	12.90	61.43	202.80	39.08
		1	3.40	16.19	408.00	78.61
	5	0	20.20	96.19	20.30	3.91
		0.5	11.80	56.19	214.10	41.25
		1	6.20	29.52	404.30	77.90
	10	0	20.60	98.10	22.00	4.24
		0.5	12.30	58.57	210.90	40.64
		1	5.2	24.76	406.40	78.30
2	2	0	14.4	68.57	27.80	5.36
		0.5	9.9	47.14	209.30	40.33
		1	2.9	13.81	408.10	78.63
	5	0	19.8	94.29	23.60	4.55
		0.5	12.9	61.43	208.80	40.23
		1	4.7	22.38	405.10	78.05
	10	0	20.2	96.19	22.00	4.24
		0.5	12.4	59.05	209.10	40.29
		1	6.1	29.05	405.80	78.19
5	5	0	20.6	98.10	21.10	4.07
		0.5	13.4	63.81	212.20	40.89
		1	6.8	32.38	404.70	77.98
	10	0	19.9	94.76	23.00	4.43
		0.5	14	66.67	208.80	40.23
		1	5.1	24.29	403.90	77.82
10	10	0	20.6	98.10	19.10	3.68
		0.5	12.6	60.00	206.00	39.69
		1	5	23.81	408.50	78.71

Tabla 5: Indicadores detección salto de ciclo. Algoritmo de combinación de geometría libre con frecuencia múltiple. Umbral de detección duplicado.

La Tabla 5 muestra el desempeño del algoritmo con el umbral de detección incrementado por un factor de dos respecto al caso anterior. Se puede observar que el grado de detección es considerablemente mejor que anteriormente, especialmente en ausencia de ruido. Esto se debe principalmente a que el número de falsas detecciones se ha visto reducido, otra vez especialmente en ausencia de ruido.

Al aumentar el umbral, parte de las medidas ruidosas quedan por debajo del mismo y por tanto el algoritmo no declara la detección de forma errónea. Gracias a esto, el algoritmo es capaz de detectar un mayor número de saltos de ciclo, ya que no se encuentra casi constantemente reiniciando el algoritmo.

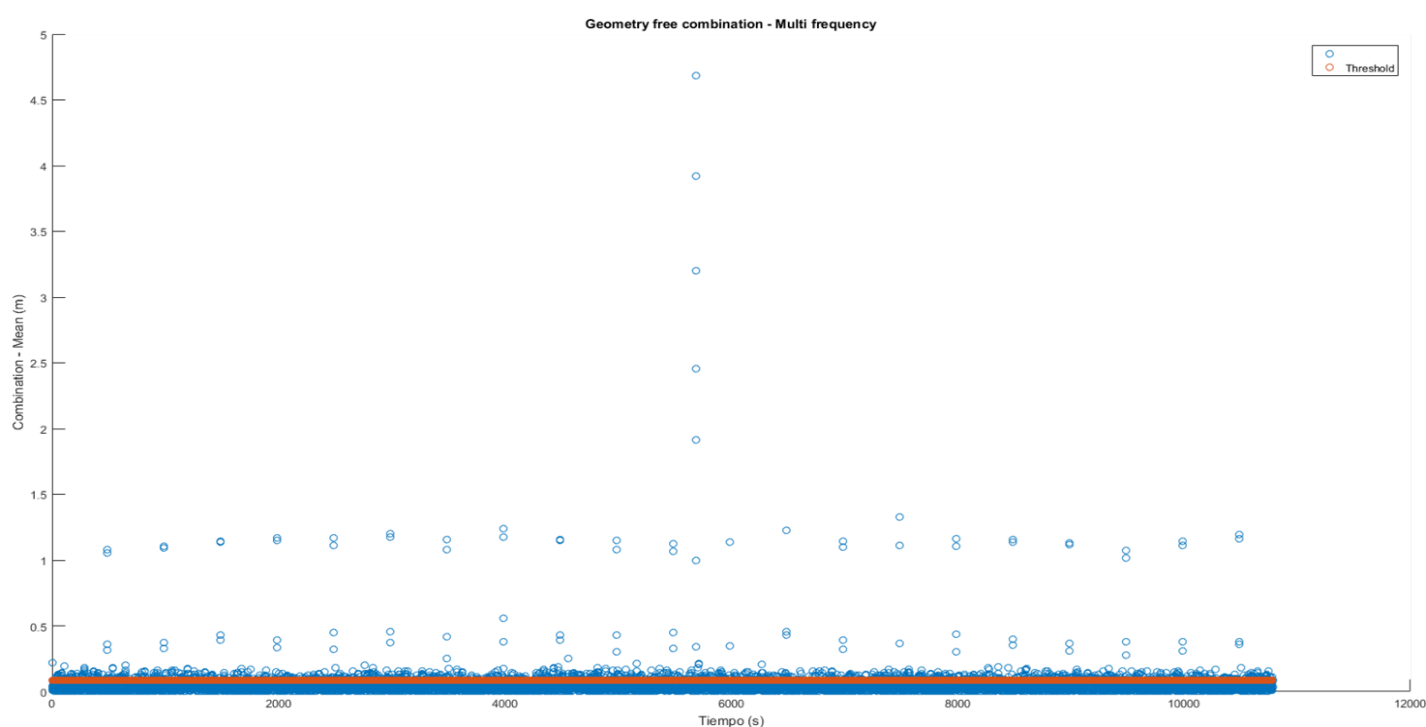


Figura 15: Detección saltos de ciclo para dos frecuencias utilizando la combinación libre de geometría. Saltos de ciclo de dos longitudes de onda. Umbral aumentado para reducir la influencia del ruido.

En la Figura 15 se puede observar la detección de saltos de ciclo utilizando la combinación libre de geometría. A pesar de que los saltos quedan por encima del umbral, también lo hacen medidas debido al ruido y al centelleo ionosférico.

Para el caso de la figura, el tamaño del salto es de 2 veces la longitud de onda y el umbral de detección se encuentra duplicado en relación con el caso original.

Tamaño saltos de ciclo L2	Tamaño saltos de ciclo L1	Grado ruido	Nº saltos detectados	Grado de detección	Nº falsas detecciones	Inmunidad frente al ruido
0	1	0	21.00	100.00	1.00	0.19
		0.5	21.00	100.00	1.00	0.19
		1	21.00	100.00	1.10	0.21
	2	0	21.00	100.00	1.00	0.19
		0.5	21.00	100.00	1.00	0.19
		1	21.00	100.00	1.00	0.19
	5	0	21.00	100.00	1.00	0.19
		0.5	21.00	100.00	1.00	0.19
		1	21.00	100.00	1.00	0.19
	10	0	21.00	100.00	1.00	0.19
		0.5	21.00	100.00	1.00	0.19
		1	21.00	100.00	1.00	0.19
1	1	0	20.90	99.52	1.00	0.19
		0.5	20.40	97.14	1.00	0.19
		1	18.30	87.14	1.00	0.19
	2	0	21.00	100.00	1.00	0.19
		0.5	21.00	100.00	1.00	0.19
		1	21.00	100.00	1.00	0.19
	5	0	21.00	100.00	1.00	0.19
		0.5	21.00	100.00	1.00	0.19
		1	21.00	100.00	1.00	0.19
	10	0	21.00	100.00	1.00	0.19
		0.5	21.00	100.00	1.00	0.19
		1	21.00	100.00	1.00	0.19
2	2	0	21.00	100.00	1.00	0.19
		0.5	21.00	100.00	1.00	0.19
		1	21.00	100.00	1.00	0.19
	5	0	21.00	100.00	1.00	0.19
		0.5	21.00	100.00	1.00	0.19
		1	21.00	100.00	1.00	0.19
5	5	0	21.00	100.00	1.00	0.19
		0.5	21.00	100.00	1.00	0.19
		1	21.00	100.00	1.00	0.19
	10	0	21.00	100.00	1.00	0.19
		0.5	21.00	100.00	1.00	0.19
		1	21.00	100.00	1.00	0.19
10	10	0	21.00	100.00	1.00	0.19
		0.5	21.00	100.00	1.00	0.19
		1	21.00	100.00	1.00	0.19

Tabla 6: Indicadores detección salto de ciclo. Algoritmo de combinación de geometría libre con frecuencia múltiple. Sin diferenciar la combinación de geometría libre.

Los resultados mostrados en la Tabla 6 pertenecen al caso en el que el algoritmo de geometría libre no lleva a cabo diferencias de las medidas para aumentar el valor del salto de ciclo presente en las medidas. El desempeño mostrado es muy positivo, ya que el grado de detección es extremadamente elevado y presenta una inmunidad frente al ruido casi total.

A pesar de ello, es importante mencionar que en todas las simulaciones de este caso específico se puede encontrar una detección falsa. Esta detección se corresponde con el efecto ionosférico implantado en las medidas. El algoritmo es incapaz de discriminar este fenómeno y si estuviese presente en más

ocasiones, el número de falsas detecciones aumentaría, empeorando al mismo tiempo el grado de detección del algoritmo.

3.4.4.4 Combinación Melbourne-Wübbena con frecuencia múltiple

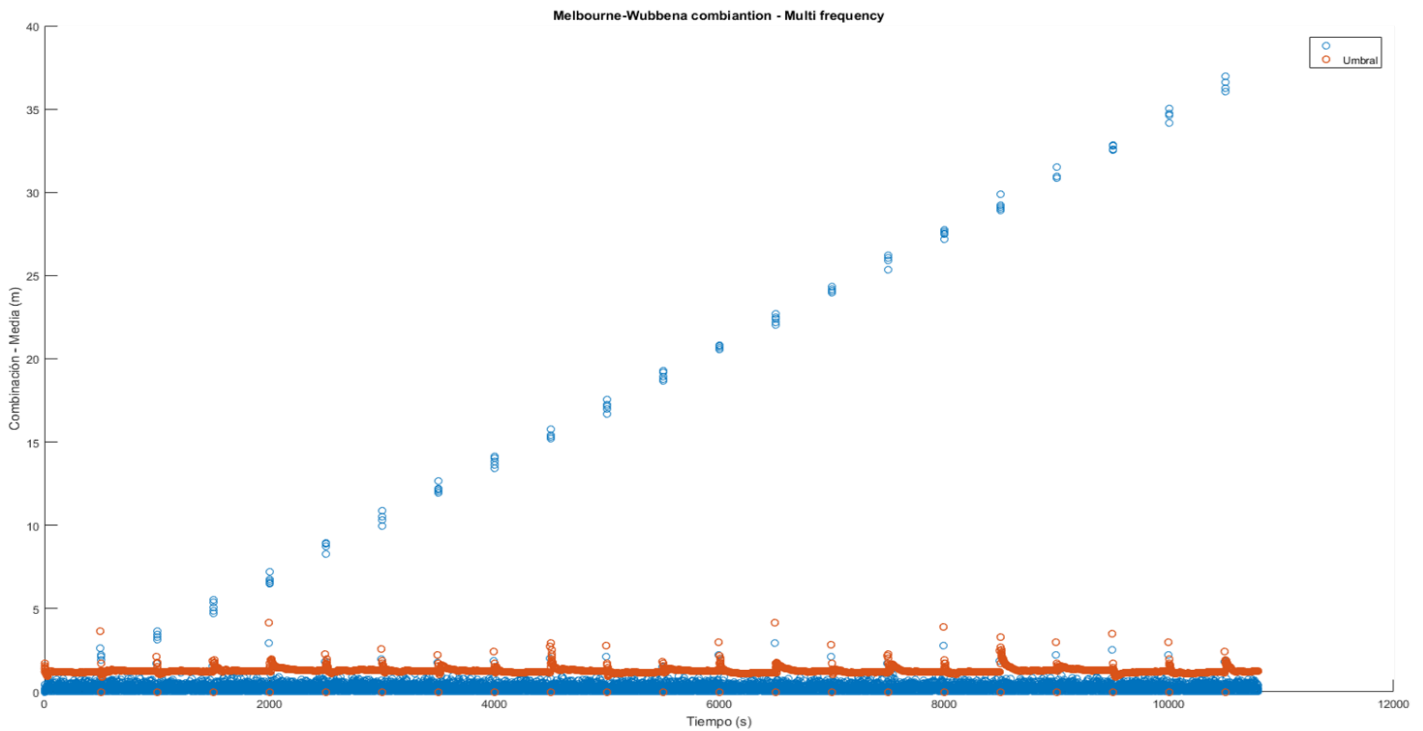


Figura 16: detección de saltos de ciclo utilizando la combinación Melbourne-Wübbena. Saltos de ciclo de dos longitudes de onda.

La Figura 16 muestra la detección de saltos de ciclo utilizando la combinación Melbourne-Wübbena para saltos de tamaño de 2 veces la longitud de onda. Se puede observar como el valor del umbral va variando a lo largo de las muestras. Además, el centelleo ionosférico no está presente en la combinación de la simulación.

Tamaño saltos de ciclo L2	Tamaño saltos de ciclo L1	Grado ruido	Nº saltos detectados	Grado de detección	Nº falsas detecciones	Inmunidad frente al ruido
0	1	0	21.00	100.00	0.00	0.00
		0.5	20.10	95.71	1.50	0.29
		1	0.70	3.33	4.00	0.77
	2	0	21.00	100.00	0.00	0.00
		0.5	20.90	99.52	0.20	0.04
		1	19.50	92.86	2.30	0.44
	5	0	21.00	100.00	0.00	0.00
		0.5	21.00	100.00	0.60	0.12
		1	21.00	100.00	0.80	0.15
	10	0	21.00	100.00	0.00	0.00
		0.5	21.00	100.00	0.40	0.08
		1	21.00	100.00	0.80	0.15
	1	0	0.00	0.00	0.00	0.00
		0.5	0.00	0.00	0.60	0.12
		1	0.00	0.00	0.70	0.13
1	1	0	21.00	100.00	0.00	0.00
		0.5	20.40	97.14	1.30	0.25
		1	1.60	7.62	9.70	1.87
	2	0	21.00	100.00	0.00	0.00
		0.5	21.00	100.00	0.40	0.08
		1	21.00	100.00	0.90	0.17
	5	0	21.00	100.00	0.00	0.00
		0.5	21.00	100.00	0.40	0.08
		1	21.00	100.00	0.80	0.15
	10	0	21.00	100.00	0.00	0.00
		0.5	21.00	100.00	0.40	0.08
		1	21.00	100.00	0.80	0.15
2	2	0	0	0.00	0.00	0.00
		0.5	0	0.00	0.50	0.10
		1	0	0.00	0.90	0.17
	5	0	21	100.00	0.00	0.00
		0.5	21	100.00	0.40	0.08
		1	21	100.00	0.80	0.15
	10	0	21	100.00	0.00	0.00
		0.5	21	100.00	0.40	0.08
		1	21	100.00	0.70	0.13
5	5	0	0	0.00	0.00	0.00
		0.5	0	0.00	0.80	0.15
		1	0	0.00	0.50	0.10
	10	0	21	100.00	0.00	0.00
		0.5	21	100.00	0.40	0.08
		1	21	100.00	0.50	0.10
10	10	0	0	0.00	0.00	0.00
		0.5	0	0.00	0.40	0.08
		1	0	0.00	0.50	0.10

Tabla 7: Indicadores detección salto de ciclo. Algoritmo de combinación Melbourne-Wübbena con frecuencia múltiple.

La Tabla 7 contiene los resultados obtenidos tras las simulaciones del algoritmo que hace uso de la combinación de Melbourne-Wübbena para la detección. Los resultados de ambos indicadores son considerablemente superiores a los obtenidos con los tres algoritmos anteriores, ya que presenta una mayor inmunidad frente al ruido que sus predecesores.

El algoritmo presenta una inmunidad frente al ruido prácticamente total. Para la mayor parte de las combinaciones de saltos de ciclo, el grado de detección es de alrededor del 100%.

Es necesario resaltar que el algoritmo en cuestión no es capaz de detectar ningún salto de ciclo cuando existe un salto de ciclo en cada frecuencia del mismo tamaño. Independientemente del tamaño de los saltos, si estos son del mismo tamaño, el algoritmo no consigue llevar a cabo ninguna detección.

Se afirma que este algoritmo ofrece mejores prestaciones que sus predecesores debido a que ofrece una inmunidad frente al ruido muy favorable. Como se comprobará en la sección que sigue, al aumentar el nivel del ruido el algoritmo sigue trabajando de manera similar, mientras que el algoritmo que utilizaba la combinación libre de geometría presenta resultados inferiores.

3.4.4.5 Uso de satélites con poca elevación

Por último, se van a simular los algoritmos que utilizan dos frecuencias en un caso en el que las medidas transmitidas pertenecen a una satélite que no tiene una elevación adecuada con respecto al receptor. Esta situación se va a simular con la existencia de un ruido superior al de los casos anteriores.

Las transmisiones de los satélites con poca elevación se encuentran más próximos a la superficie terrestre, y por tanto sujetas a más ruido térmico. Además, el recorrido de las transmisiones cuenta con mayores variaciones que si fuese totalmente vertical.

Para el algoritmo que utiliza la combinación libre de geometría se van a simular dos casos: llevando a cabo a las diferencias con el umbral incrementado y sin llevarlas a cabo con el umbral original.

Tamaño saltos de ciclo L2	Tamaño saltos de ciclo L1	Grado ruido	Nº saltos detectados	Grado de detección	Nº falsas detecciones	Inmunidad frente al ruido
0	1	0	19.90	94.76	19.30	3.72
		0.5	1.40	6.67	505.20	97.34
		1	1.20	5.71	505.70	97.44
	2	0	20.30	96.67	21.40	4.12
		0.5	1.50	7.14	505.00	97.30
		1	0.70	3.33	510.70	98.40
	5	0	20.70	98.57	19.00	3.66
		0.5	1.70	8.10	505.30	97.36
		1	1.60	7.62	509.30	98.13
	10	0	20.30	96.67	21.80	4.20
		0.5	1.80	8.57	505.10	97.32
		1	1.20	5.71	509.90	98.25
1	1	0	2.60	12.38	38.20	7.36
		0.5	1.00	4.76	506.00	97.50
		1	1.40	6.67	510.00	98.27
	2	0	18.70	89.05	21.10	4.07
		0.5	1.20	5.71	505.20	97.34
		1	1.00	4.76	510.30	98.32
	5	0	20.70	98.57	18.80	3.62
		0.5	0.80	3.81	505.70	97.44
		1	0.70	3.33	510.20	98.30
	10	0	20.30	96.67	25.10	4.84
		0.5	1.20	5.71	505.80	97.46
		1	1.10	5.24	510.10	98.29
2	2	0	14.80	70.48	27.80	5.36
		0.5	0.80	3.81	505.90	97.48
		1	1.00	4.76	510.30	98.32
	5	0	20.20	96.19	21.40	4.12
		0.5	1.20	5.71	505.20	97.34
		1	1.40	6.67	509.60	98.19
	10	0	20.00	95.24	21.10	4.07
		0.5	0.70	3.33	505.90	97.48
		1	1.40	6.67	509.60	98.19
5	5	0	20.20	96.19	21.70	4.18
		0.5	1.00	4.76	505.90	97.48
		1	1.30	6.19	510.10	98.29
	10	0	20.50	97.62	18.20	3.51
		0.5	1.20	5.71	505.50	97.40
		1	1.20	5.71	509.90	98.25
10	10	0	19.80	94.29	23.10	4.45
		0.5	1.80	8.57	504.70	97.24
		1	1.20	5.71	509.80	98.23

Tabla 8: Indicadores detección salto de ciclo. Algoritmo de combinación de geometría libre con frecuencia múltiple. Satélite con poca elevación.

La Tabla 8 muestra los indicadores del algoritmo que utiliza la combinación libre de geometría para un satélite con poca elevación. Es evidente que en presencia del ruido los resultados ofrecidos son pobres. La inmunidad frente al ruido de este algoritmo deja mucho que desear y esto se traslada a un grado de detección muy reducido.

Tamaño saltos de ciclo L2	Tamaño saltos de ciclo L1	Grado ruido	Nº saltos detectados	Grado de detección	Nº falsas detecciones	Inmunidad frente al ruido
0	1	0	21.00	100.00	1.00	0.19
		0.5	3.60	17.14	455.40	87.75
		1	1.50	7.14	496.80	95.72
	2	0	21.00	100.00	1.00	0.19
		0.5	3.20	15.24	457.90	88.23
		1	0.80	3.81	497.10	95.78
	5	0	21.00	100.00	1.00	0.19
		0.5	3.20	15.24	456.00	87.86
		1	2.30	10.95	496.20	95.61
	10	0	21.00	100.00	1.00	0.19
		0.5	2.70	12.86	456.80	88.02
		1	1.70	8.10	496.00	95.57
1	1	0	20.90	99.52	1.00	0.19
		0.5	2.30	10.95	455.50	87.76
		1	1.30	6.19	496.50	95.66
	2	0	21.00	100.00	1.00	0.19
		0.5	3.40	16.19	456.30	87.92
		1	1.00	4.76	496.50	95.66
	5	0	21.00	100.00	1.00	0.19
		0.5	3.20	15.24	458.20	88.29
		1	1.40	6.67	496.50	95.66
	10	0	21.00	100.00	1.00	0.19
		0.5	3.10	14.76	456.40	87.94
		1	1.50	7.14	496.50	95.66
2	2	0	21.00	100.00	1.00	0.19
		0.5	3.10	14.76	455.60	87.78
		1	1.30	6.19	496.40	95.65
	5	0	21.00	100.00	1.00	0.19
		0.5	3.30	15.71	457.30	88.11
		1	2.20	10.48	495.70	95.51
	10	0	21.00	100.00	1.00	0.19
		0.5	3.30	15.71	455.90	87.84
		1	1.70	8.10	496.50	95.66
5	5	0	21.00	100.00	1.00	0.19
		0.5	3.20	15.24	456.90	88.03
		1	1.50	7.14	496.90	95.74
	10	0	21.00	100.00	1.00	0.19
		0.5	3.20	15.24	457.30	88.11
		1	2.90	13.81	494.70	95.32
10	10	0	21.00	100.00	1.00	0.19
		0.5	4.40	20.95	455.10	87.69
		1	1.10	5.24	496.70	95.70

Tabla 9 Indicadores detección salto de ciclo. Algoritmo de combinación de geometría libre con frecuencia múltiple. Sin diferenciar la combinación de geometría libre. Satélite con poca elevación.

En la Tabla 9 se pueden observar resultados muy similares a los de la Tabla 8. En este caso, la combinación libre de geometría sin llevar a cabo la diferenciación también es extremadamente sensible al ruido.

Tamaño saltos de ciclo L2	Tamaño saltos de ciclo L1	Grado ruido	Nº saltos detectados	Grado de detección	Nº falsas detecciones	Inmunidad frente al ruido
0	1	0	21.00	100.00	0.00	0.00
		0.5	14.90	70.95	6.50	1.25
		1	0.20	0.95	1.40	0.27
	2	0	21.00	100.00	0.00	0.00
		0.5	21.00	100.00	0.50	0.10
		1	12.80	60.95	9.30	1.79
	5	0	21.00	100.00	0.00	0.00
		0.5	21.00	100.00	0.60	0.12
		1	21.00	100.00	1.30	0.25
	10	0	21.00	100.00	0.00	0.00
		0.5	21.00	100.00	0.80	0.15
		1	21.00	100.00	1.10	0.21
1	1	0	0.00	0.00	0.00	0.00
		0.5	0.00	0.00	0.70	0.13
		1	0.00	0.00	0.40	0.08
	2	0	21.00	100.00	0.00	0.00
		0.5	14.80	70.48	6.80	1.31
		1	0.30	1.43	2.60	0.50
	5	0	21.00	100.00	1.00	0.19
		0.5	21.00	100.00	0.50	0.10
		1	21.00	100.00	1.40	0.27
	10	0	21.00	100.00	0.00	0.00
		0.5	21.00	100.00	0.90	0.17
		1	21.00	100.00	1.10	0.21
2	2	0	0.00	0.00	0.00	0.00
		0.5	0.00	0.00	0.80	0.15
		1	0.00	0.00	1.00	0.19
	5	0	21.00	100.00	0.00	0.00
		0.5	21.00	100.00	0.70	0.13
		1	21.00	100.00	1.10	0.21
	10	0	21.00	100.00	0.00	0.00
		0.5	21.00	100.00	0.80	0.15
		1	21.00	100.00	1.80	0.35
5	5	0	0.00	0.00	0.00	0.00
		0.5	0.00	0.00	0.60	0.12
		1	0.00	0.00	0.70	0.13
	10	0	21.00	100.00	0.00	0.00
		0.5	21.00	100.00	0.30	0.06
10	10	1	21.00	100.00	1.20	0.23
		0	0.00	0.00	0.00	0.00
		0.5	0.00	0.00	0.60	0.12
		1	0.00	0.00	0.30	0.06

Tabla 10: Indicadores detección salto de ciclo. Algoritmo de combinación Melbourne-Wübbena con frecuencia múltiple. Satélite con poca elevación.

La Tabla 10 muestra los resultados del algoritmo que utiliza la combinación Melbourne-Wübbena para el caso de un satélite con poca elevación. Como se puede observar, el grado de detección y la inmunidad frente al ruido tienen valores similares al caso original.

Para los casos en los que se implantan saltos de ciclo del mismo tamaño en ambas frecuencias, el algoritmo sigue siendo incapaz de realizar detecciones. Estos casos hacen referencia a las situaciones en las que se introducen saltos de ciclo con el mismo tamaño (en longitudes de onda de cada frecuencia) en ambas frecuencia (L1 y L2). A pesar de ello, el desempeño general del algoritmo es muy positivo.

3.4.5 Comparativa y valoración

Los resultados mostrados en la sección anterior arrojan luz sobre ciertas características de notable importancia de los algoritmos analizados en este capítulo.

Una propiedad interesante de los algoritmos evaluados es la necesidad de algunos de poseer datos históricos para poder realizar la detección. Este es el caso de los tres primeros algoritmos (ambos algoritmos que utilizan una sola frecuencia y la combinación libre de geometría). En cambio, la combinación Melbourne-Wübbena no necesita de datos anteriores para llevar a cabo la detección. Esto supone una ventaja cualitativa sobre el resto de algoritmos.

Esta característica tiene un impacto directo sobre los que cuantifican la detección de saltos. Al requerir un “ventana” de datos para poder realizar la detección, ese número de muestras no se tiene en cuenta como para incluir un salto de ciclo. Si el algoritmo no tiene una elevada inmunidad frente al ruido, y por tanto detecta un gran número de saltos falsos, si un salto de ciclo se encontrase situado en una muestra incluida en la “ventana” de datos requerida para procesar el algoritmo, este no sería detectado. Esto causa que el grado de detección sea inferior al de la combinación Melbourne-Wübbena, especialmente en presencia de ruido.

La necesidad de modificar el umbral de detección para los algoritmos de frecuencia singular constituye un aliciente para utilizar los algoritmos de frecuencias múltiples. La variación del umbral según el tamaño del salto de ciclo es inviable para una aplicación real ya que no es posible anticipar el tamaño del salto de ciclo. Además, la presencia de ruido perjudica de forma considerable el desempeño de estos algoritmos.

Con respecto al algoritmo que utiliza la combinación libre de geometría, los resultados obtenidos en las simulaciones exhiben un grado de detección y una inmunidad frente al ruido muy reducidos. La razón de estos resultados reside en que al realizar las diferencias para amplificar el valor del salto en las muestras,

también se amplifica el ruido. Debido a una considerable presencia de ruido y a la necesidad de contar con datos históricos, el algoritmo detecta un número extremadamente elevado de saltos falsos y no es capaz de detectar ninguno real.

El valor del umbral se incrementa para reducir el efecto del ruido sobre el desempeño del algoritmo. Los saltos de ciclo siguen quedando por encima del umbral y por tanto se declaran las detecciones. En cambio, parte del ruido se queda por debajo del umbral, por lo que el número de falsas detecciones disminuye.

Por otro lado, el objetivo de la diferenciación es ampliar el valor de los saltos de ciclo presentes en las medidas. Basándose en los resultados obtenidos, se puede concluir que la diferenciación no es necesaria y empeora el desempeño del algoritmo. Al ampliar el ruido, el número de falsas detecciones incrementa considerablemente lo que impide la detección de los saltos. La aplicación de la diferenciación debería estar limitada a casos en los que los saltos tengan un tamaño muy reducido y en ausencia de ruido. Además, en todos los casos simulados con esta combinación, el algoritmo resulta incapaz de discriminar el efecto causado por el centelleo de la ionosfera.

Por último, el algoritmo que hace uso de la combinación Melbourne-Wübbena ofrece unos resultados muy favorables. Tanto el grado de detección como la inmunidad al ruido son prácticamente totales. El único inconveniente de este algoritmo es la incapacidad de detectar saltos de ciclo cuando estos se encuentran en ambas frecuencias y son del mismo número de longitudes de onda. Afortunadamente, este escenario es poco realista y se dará en la práctica con muy poca probabilidad.

Respecto a los casos simulados utilizando un satélite con poca elevación, el aumento del ruido presente en las medidas afecta de forma considerable al algoritmo que utiliza la combinación libre de geometría, tanto con y sin diferenciar. Al contrario, la combinación de Melbourne-Wübbena presenta una elevada inmunidad al ruido y sus resultados no varían significativamente con el incremento del ruido.

CAPÍTULO 4: Algoritmos de selección de satélites

En este capítulo se va a tratar el tema de la selección de satélites. Con la proliferación de los sistemas de navegación por satélite se cuenta con un amplio número de satélites a la vista para llevar a cabo el cálculo de la posición. Resulta necesario implementar una metodología de selección de satélites para utilizar en la navegación con el fin de mantener la carga computacional a niveles adecuados.

En primer lugar, se va a dedicar una sección a comentar las ventajas que ofrece el uso de sistemas multiconstelación en la navegación. Seguidamente, se revisarán distintos algoritmos de selección de satélites descritos en artículos científicos. Después, se implementarán dichos algoritmos utilizando datos simulados para poder valorar su desempeño. Por último, se llevará a cabo un análisis de los resultados obtenidos en las simulaciones y se compararán los algoritmos entre sí.

4.1 Sistemas multiconstelación

Esta sección servirá como apartado introductorio para los sistemas de navegación multiconstelación. En primer lugar se comentarán las distintas ventajas que aportan este tipo de sistemas en los sistemas de navegación por satélite. Después, se introducirá el concepto de DOP (*Dilution Of Precision*) que será de gran relevancia en lo que resta de capítulo. Por último, se argumentará la necesidad de una metodología de selección de satélites para sistemas multiconstelación.

4.1.1 Aportación de los sistemas multiconstelación a la solución de navegación

La navegación por satélite ha tenido una dependencia significativa en el sistema americano GPS, pero debido a la aparición de los sistemas Galileo (Europa) y BeiDou (China); y el resurgir del sistema ruso GLONASS, resulta necesaria la investigación de sistemas de navegación multiconstelación. Un sistema

multiconstelación utiliza más de una fuente GNSS (GPS, Galileo, GLONASS y BeiDou) para la navegación.

Con los receptores utilizando únicamente el sistema GPS se contaba con un total de 32 satélites para llevar a cabo el posicionamiento. Se estima que cuando los cuatro sistemas totalmente operativos el número de satélites disponibles para la navegación será de 120 satélites (Meng, Zhu, & Wang, Research of Fast Satellite Selection Algorithm for Multi-Constellation, 2016).

Los beneficios que supone la implementación de sistemas multiconstelación han sido estudiados en detalle por múltiples autores. La primera ventaja que se puede destacar es un aumento considerable en el número de satélites visibles para el receptor.

La tabla que se muestra a continuación contiene una media del número de satélites visibles medio al llevar a cabo una simulación utilizando las distintas constelaciones GNSS de forma conjunta (Pan, Cai, Santerre, & Zhang, 2016).

	GPS	GPS/GLO	GPS/GLO/BDS	GPS/GLO/BDS/GAL
Nº de satélites	6.9	13.3	24.8	27.8

Tabla 11: Número de satélites visibles según distintas combinaciones de sistemas GNSS.

Se puede observar que el número medio de satélites visibles aumenta al contar con más constelaciones. Entre el primer caso (únicamente con la constelación GPS) y el último (todas las constelaciones disponibles) se produce un incremento en el número de satélites visibles del 300%.

Este aumento conlleva una mayor redundancia de medidas y una mejor fiabilidad a la hora de realizar el cálculo de posición. Este incremento de satélites a la vista es especialmente interesante en entornos con presencia de obstáculos.

Además, con un mayor número de satélites se consigue una mejoría en la geometría espacial. Contar con una mejor distribución de los satélites en cielo se traduce en un aumento en la precisión de la solución de navegación.

Para valorar la precisión que ofrecen un conjunto de satélites se suele acudir a los factores DOP (*Dilution Of Precision*), que se explicarán en detalle en la sección siguiente. Básicamente, menores valores del DOP significan mayor precisión, mientras que un valor elevado de DOP conlleva una precisión inferior. Contar con más satélites para el cálculo de la posición significa reducir el valor de los factores del DOP y por tanto aumentar la precisión.

Es posible afirmar que la precisión del posicionamiento depende en cierta medida del número de satélites a la vista. Por ejemplo, en la tabla que se muestra a continuación se puede observar los distintos valores de GDOP (*Geometric Dilution Of Position*) obtenidos para distintas combinaciones de constelaciones GNSS (Meneghini & Parente, 2017).

	1<GDOP<3 (%)	3<GDOP<5 (%)	GDOP>5 (%)	Nº satélites insuficiente (%)
GPS	82	17	1	0
GLONASS	42	47	9	2
GALILEO	0	13	31	56
GPS+GLONASS	96	4	0	0
GPS+GALILEO	83	16	1	0
GLONASS+GALILEO	44	51	5	0
GPS+GLONASS+GALILEO	96	4	0	0

Tabla 12: Valor del GDOP según distintas combinaciones de sistemas GNSS.

Es imprescindible mencionar que estos resultados parten de datos de un sistema Galileo pre operacional, y por tanto no contaba con la flota de satélites al completo. Por ello, la mejora de GPS y GLONASS al combinar su utilización con Galileo no es significativa. Aun así, estos resultados evidencian que las combinaciones de constelaciones GNSS y por tanto el incremento del número de satélites visibles tienen un impacto positivo en el GDOP. Por tanto, aumentar

el número de satélites reduce el valor del GDOP e incrementa la precisión en la navegación.

Se puede afirmar que la implementación de los sistemas multiconstelación acarrea un incremento en el número de satélites visibles, una mejoría en la geometría espacial y un aumento de precisión, continuidad y fiabilidad de la solución.

4.1.2 Dilution Of Precision (DOP)

Para determinar las tres coordenadas de posición de un receptor y el error del reloj que tiene, se necesitan un mínimo de cuatro satélites. Como se ha mencionado en el apartado anterior, el uso de los factores del DOP es una práctica común para determinar la precisión en el posicionamiento que ofrece un conjunto de satélites. Por tanto, mediante el cálculo de los factores DOP de las distintas combinaciones de satélites de entre el conjunto de satélites visibles, es posible determinar el grupo óptimo de satélites que ofrece la mayor precisión.

Aunque es cierto que en la realidad se utilizan más satélites de los estrictamente necesarios por temas de redundancia y fiabilidad, la demostración de la obtención de los factores DOP se llevará a cabo con el número mínimo de satélites necesarios. Por cuestiones de claridad, en primer lugar se detallará el desarrollo para la obtención de los factores DOP en el caso de una única constelación.

El cálculo de los factores del DOP es interesante, en primer lugar, porque se trata de un método óptimo para escoger el conjunto de satélites que va a proporcionar la mejor precisión en la solución de navegación, y en segundo lugar, porque permite valorar la precisión que ofrecen distintos grupos de satélites.

En ausencia de todas las influencias físicas excepto el error del reloj del receptor, los pseudorangs de 4 satélites son:

Ecuación 4.1

$$\begin{aligned}\rho_1 &= \sqrt{(x_1 - x_u)^2 + (y_1 - y_u)^2 + (z_1 - z_u)^2} + c \cdot t_u \\ \rho_2 &= \sqrt{(x_2 - x_u)^2 + (y_2 - y_u)^2 + (z_2 - z_u)^2} + c \cdot t_u \\ \rho_3 &= \sqrt{(x_3 - x_u)^2 + (y_3 - y_u)^2 + (z_3 - z_u)^2} + c \cdot t_u \\ \rho_4 &= \sqrt{(x_4 - x_u)^2 + (y_4 - y_u)^2 + (z_4 - z_u)^2} + c \cdot t_u\end{aligned}$$

Por tanto, las ecuaciones de pseudorango puede expresarse como:

Ecuación 4.2

$$\rho_j = \sqrt{(x_j - x_u)^2 + (y_j - y_u)^2 + (z_j - z_u)^2} + c \cdot t_u = f(x_u, y_u, z_u, t_u)$$

Si contamos con una estimación de las coordenadas y el error del reloj del receptor (x' , y' , z' , t'), la ecuación anterior pasa a ser:

Ecuación 4.3

$$\rho'_j = \sqrt{(x_j - x'_u)^2 + (y_j - y'_u)^2 + (z_j - z'_u)^2} + c \cdot t'_u = f(x'_u, y'_u, z'_u, t'_u)$$

La posición real es la posición estimada (x'_u) más un incremento (Δx_u):

Ecuación 4.4

$$\begin{aligned}x_u &= x'_u + \Delta x_u \\ y_u &= y'_u + \Delta y_u \\ z_u &= z'_u + \Delta z_u \\ t_u &= t'_u + \Delta t_u\end{aligned}$$

Se puede concluir que la posición real del usuario es una función de la estimación y del incremento:

Ecuación 4.5

$$f(x_u, y_u, z_u, t_u) = f(x'_u + \Delta x_u, y'_u + \Delta y_u, z'_u + \Delta z_u, t'_u + \Delta t_u)$$

Desarrollando la serie de Taylor se pasa a obtener un sistema lineal de ecuaciones.

Ecuación 4.6

$$\rho_j = \rho'_j - \frac{x_j - x'_u}{r'_j} \cdot \Delta x_u - \frac{y_j - y'_u}{r'_j} \cdot \Delta y_u - \frac{z_j - z'_u}{r'_j} \cdot \Delta z_u + c \cdot t_u$$

Para mayor claridad se definen:

$$\Delta \rho = \rho'_j - \rho_j$$

$$a_{x_i} = \frac{x_j - x'_u}{r'_j}$$

$$a_{y_i} = \frac{y_j - y'_u}{r'_j}$$

$$a_{z_i} = \frac{z_j - z'_u}{r'_j}$$

Por tanto, para cada satélite j se tiene:

Ecuación 4.7

$$\Delta \rho_j = a_{x_j} \cdot \Delta x_u + a_{y_j} \cdot \Delta y_u + a_{z_j} \cdot \Delta z_u - c \cdot \Delta t_u$$

Para definir el sistema de ecuaciones en forma matricial se precisan las siguientes matrices:

$$\Delta \rho = \begin{bmatrix} \Delta \rho_1 \\ \Delta \rho_2 \\ \Delta \rho_3 \\ \Delta \rho_4 \end{bmatrix}$$

$$H = \begin{bmatrix} a_{x_1} & a_{y_1} & a_{z_1} & 1 \\ a_{x_2} & a_{y_2} & a_{z_2} & 1 \\ a_{x_3} & a_{y_3} & a_{z_3} & 1 \\ a_{x_4} & a_{y_4} & a_{z_4} & 1 \end{bmatrix}$$

$$\Delta x = \begin{bmatrix} \Delta x_u \\ \Delta y_u \\ \Delta z_u \\ -c \cdot \Delta t_u \end{bmatrix}$$

Por tanto, el sistema de ecuaciones en forma matricial es:

Ecuación 4.8

$$\Delta\rho = H \cdot \Delta x$$

Para el caso de cuatro satélites ($j=4$), el sistema se resuelve de la siguiente forma.

Ecuación 4.9

$$\Delta x = H^{-1} \cdot \Delta\rho$$

Si nos encontramos en un caso de sistema multiconstelación, en el que son necesario más de 4 satélites, las incógnitas se despejaría utilizando el método de los mínimos cuadrados.

Ecuación 4.10

$$\Delta x = (H^T \cdot H)^{-1} \cdot H^T \cdot \Delta\rho$$

Como las medidas de pseudorango no se encuentran en ausencia de errores, se define ρ_T como la medida de pseudorango sin errores, ρ_L como la medida de pseudorango en el punto de linearización y $d\rho$ el término de error.

Ecuación 4.11

$$\Delta\rho = \rho_T - \rho_L + d\rho$$

De forma similar:

Ecuación 4.12

$$\Delta x = x_T - x_L + dx$$

Por tanto, mediante la Ecuación 4.11 y la Ecuación 4.12 y la relación $x_T - x_L = (H^T H)^{-1} H^T (\rho_T - \rho_L)$ la Ecuación 4.10 pasa a ser:

Ecuación 4.13

$$\Delta x = [(H^T \cdot H)^{-1} \cdot H^T] \cdot d\rho = k \cdot d\rho$$

Esta ecuación hace referencia a la relación entre los errores de pseudorange y los errores en la posición computada.

La matriz H que se definió anteriormente depende únicamente en la geometría del receptor y los satélites.

La covarianza de los errores de la posición computada se define como un múltiplo escalar de la matriz $(H^T \cdot H)^{-1}$.

Ecuación 4.14

$$cov(dx) = (H^T \cdot H)^{-1} \cdot \sigma_{URE}^2$$

$$cov(dx) = \begin{bmatrix} \sigma_{x_u}^2 & \sigma_{y_u x_u}^2 & \sigma_{z_u x_u}^2 & \sigma_{t_u x_u}^2 \\ \sigma_{x_u y_u}^2 & \sigma_{y_u}^2 & \sigma_{z_u y_u}^2 & \sigma_{t_u y_u}^2 \\ \sigma_{x_u z_u}^2 & \sigma_{y_u z_u}^2 & \sigma_{z_u}^2 & \sigma_{t_u z_u}^2 \\ \sigma_{x_u t_u}^2 & \sigma_{y_u t_u}^2 & \sigma_{z_u t_u}^2 & \sigma_{t_u}^2 \end{bmatrix}$$

σ_{URE}^2 hace referencia a la suma estadística de las contribuciones de todas las fuentes de error asociadas a cada satélite (*User Equivalent Range Error*). Los componentes de la matriz $(H^T \cdot H)^{-1}$ cuantifican como los errores en el pseudorange se traducen en componentes de la covarianza de dx .

Los factores DOP se definen en términos del ratio de combinaciones entre los componentes de la covarianza de dx y σ_{URE} .

a) Geometry Dilution Of Precision

Ecuación 4.15

$$GDOP = \frac{\sqrt{\sigma_{x_u}^2 + \sigma_{y_u}^2 + \sigma_{z_u}^2 + \sigma_{t_u}^2}}{\sigma_{URE}}$$

b) Position Dilution Of Precision

Ecuación 4.16

$$PDOP = \frac{\sqrt{\sigma_{x_u}^2 + \sigma_{y_u}^2 + \sigma_{z_u}^2}}{\sigma_{URE}}$$

c) Horizontal Dilution Of Precision

Ecuación 4.17

$$HDOP = \frac{\sqrt{\sigma_{x_u}^2 + \sigma_{y_u}^2}}{\sigma_{UERE}}$$

d) Vertical Dilution Of Precision

Ecuación 4.18

$$VDOP = \frac{\sqrt{\sigma_{z_u}^2}}{\sigma_{UERE}}$$

e) Time Dilution Of Precision

Ecuación 4.19

$$TDOP = \frac{\sqrt{\sigma_{t_u}^2}}{\sigma_{UERE}}$$

Las geometrías que cuenten con valores elevados en los factores DOP ofrecerán menor precisión en la solución que las que tengan valores reducidos en los factores DOP.

Retomando lo comentado en el apartado anterior, se han llevado a cabo varios estudios que analizan la variación de los factores DOP al modificar el número de satélites.

Si se añade un satélite que pertenece a una constelación que ya estaba presente, el valor del GDOP disminuye siempre. En cambio, si el satélite pertenece a una constelación distinta, el valor del GDOP aumenta (Teng & Wang, New Characteristics of Geometric Dilution of Precision (GDOP) for Multi-

GNSS Constellations, 2014). Se obtienen conclusiones similares cuando se analizan otros factores como el PDOP y el TDOP (Teng & Wang, Some Remarks on PDOP and TDOP for Multi-GNSS Constellations, 2016). Esto se debe a que al añadir un satélite de una constelación nueva, se introduce una nueva incógnita (la diferencia instantánea entre las referencias de tiempo de ambas constelaciones), por lo que la ecuación del satélite adicional tiene como única función resolver dicha incógnita.

En el caso de los sistemas multiconstelación se tendrá un satélite extra por cada constelación adicional que se utilice. Además, la resolución del sistema de ecuaciones se debe llevar a cabo mediante el método de los mínimos cuadrados. Obviamente, el sistema ahora cuenta con un mayor número de ecuaciones.

Ecuación 4.20

$$\Delta x_A = \hat{H}_A \cdot \Delta \rho_A$$

$$\Delta x_B = \hat{H}_B \cdot \Delta \rho_B$$

$$\Delta x_C = \hat{H}_C \cdot \Delta \rho_C$$

$$\Delta x_D = \hat{H}_D \cdot \Delta \rho_D$$

Siendo A, B, C y D las distintas constelaciones de satélites del sistema.

$$\Delta \rho_n = \begin{bmatrix} \Delta \rho_1 \\ \Delta \rho_2 \\ \vdots \\ \Delta \rho_{\alpha_n} \end{bmatrix}$$

$$H_n = \begin{bmatrix} a_{nx_1} & a_{ny_1} & a_{nz_1} & 1 \\ a_{nx_2} & a_{ny_2} & a_{nz_2} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ a_{nx_{\alpha_n}} & a_{ny_{\alpha_n}} & a_{nz_{\alpha_n}} & 1 \end{bmatrix}$$

$$\Delta x_n = \begin{bmatrix} \Delta x_u \\ \Delta y_u \\ \Delta z_u \\ -c \cdot \Delta t_{n_u} \end{bmatrix}$$

Siendo α_n el número de satélites para constelación n.

Resulta necesario añadir una variación a la matriz H. La última columna de la matriz está relacionada con el offset del reloj del receptor. Cada constelación utilizada contará con un offset particular, lo que provoca la aparición de nuevas variables en el sistema de ecuaciones.

Existe la posibilidad de computar estos errores de reloj en función de uno de ellos. Es decir, los errores de reloj de las constelaciones B, C y D pueden ser con respecto al error de la constelación A. Con esto, la matriz H de una constelación B sería:

$$H_B = \begin{bmatrix} a_{Bx_1} & a_{By_1} & a_{Bz_1} & 1 & 1 \\ a_{Bx_2} & a_{By_2} & a_{Bz_2} & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{Bx_{\alpha_B}} & a_{By_{\alpha_B}} & a_{Bz_{\alpha_B}} & 1 & 1 \end{bmatrix}$$

Con esto, se consigue evitar la aparición de una matriz dispersa, en la que la mayor parte de sus elementos son nulos.

Dicho de otra forma, en lugar de incluir el error de reloj en absoluto para cada constelación se toma una constelación como referencia y los errores de las otras se expresan mediante un desfase con respecto a la de referencia. De este modo se asegura que al menos una columna tendrá todo ‘unos’ y se reduce la aparición de ‘ceros’ en la matriz H. Esto proporciona una mejor estabilidad numérica a la hora de realizar los cálculos mediante una computadora.

Para el conjunto de constelaciones se tiene:

Ecuación 4.21

$$\begin{aligned} \Delta\rho &= H \cdot \Delta x \\ \Delta\rho &= [\Delta\rho_A \quad \Delta\rho_B \quad \Delta\rho_C \quad \Delta\rho_D]^T \\ \Delta x &= [\Delta x_u \quad \Delta y_u \quad \Delta z_u \quad c \cdot t_{A_u} \quad c \cdot t_{B_u} \quad c \cdot t_{C_u} \quad c \cdot t_{D_u}]^T \end{aligned}$$

$$H = \begin{bmatrix} a_A & 1_{\alpha_A} & 0_{\alpha_A} & 0_{\alpha_A} & 0_{\alpha_A} \\ a_B & 1_{\alpha_B} & 1_{\alpha_B} & 0_{\alpha_B} & 0_{\alpha_B} \\ a_C & 1_{\alpha_C} & 0_{\alpha_C} & 1_{\alpha_C} & 0_{\alpha_C} \\ a_D & 1_{\alpha_D} & 0_{\alpha_D} & 0_{\alpha_D} & 1_{\alpha_D} \end{bmatrix}$$

$$a_n = \begin{bmatrix} a_{nx_1} & a_{ny_1} & a_{nz_1} \\ a_{nx_2} & a_{ny_2} & a_{nz_2} \\ \vdots & \vdots & \vdots \\ a_{nx_{\alpha_n}} & a_{ny_{\alpha_n}} & a_{nz_{\alpha_n}} \end{bmatrix}$$

Para resolver el sistema se debe acudir al método de los mínimos cuadrados.

$$\Delta x = (H^T \cdot H)^{-1} \cdot H^T \cdot \Delta \rho$$

Los factores DOP se obtienen de la misma forma que para el sistema de una única constelación.

4.1.3 Necesidad de la selección de satélites

En teoría, un receptor puede utilizar todos los satélites visibles para mitigar los errores en las medidas. Como se ha apuntado en la sección anterior, a mayor número de satélites, mayor será la precisión de la solución. Va a existir por tanto un compromiso entre el desempeño del sistema de navegación y la carga computacional.

Al aumentar el número de satélites se incrementan los recursos computacionales necesarios y el consumo energético del receptor. En el caso de aplicaciones en tiempo real, se requiere de un nivel de desempeño específico en un periodo de tiempo reducido.

Además, llegados a un determinado número de satélites, el aumento de la precisión en la solución pasa a ser insignificante al seguir aumentando la cantidad de satélites.

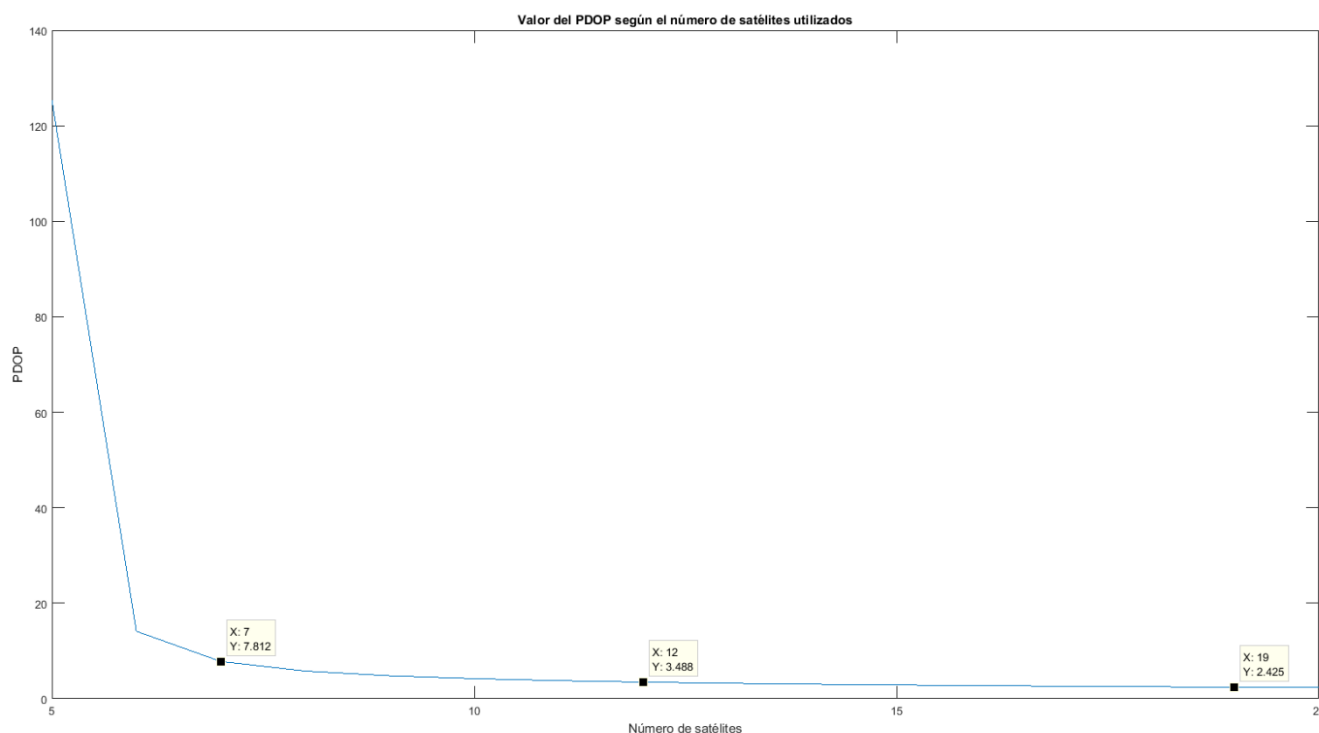


Figura 17: Valor del PDOP según el número de satélites utilizados.

La figura anterior muestra como varía el valor del PDOP según el número de satélites que se utilicen para su cómputo. Al aumentar el número de satélites, el valor del PDOP decrece. Esta reducción se produce de forma más pronunciada cuando el número de satélite es menor. Al llegar a cierto número de satélites (~12) la reducción del PDOP es mínima. A partir de este valor, no compensa seguir aumentando el número de satélites ya que la mejoría en la precisión no es suficiente y se requiere de una carga computacional excesivamente elevada.

En la tabla que se expone a continuación se muestran los cambios en la carga computacional en un sistema de navegación al incrementar el número de satélites partiendo de una solución con 10 satélites (Nie, Gao, Wang, & Ji, 2016).

	20 satélites	30 satélites
Número de multiplicaciones	2.63 veces	5.93 veces
Número de sumas	2.99 veces	6.05 veces

Tabla 13: Número de multiplicaciones y sumas en comparación con el caso de 10 satélites.

Se puede observar claramente el incremento significativo en el número de operaciones necesarias cuando se aumenta el número de satélites.

La Figura 18 muestra el tiempo requerido para el cálculo de la solución de navegación mediante el método de los mínimos cuadrados al ir modificando el número de satélites utilizados para la navegación.

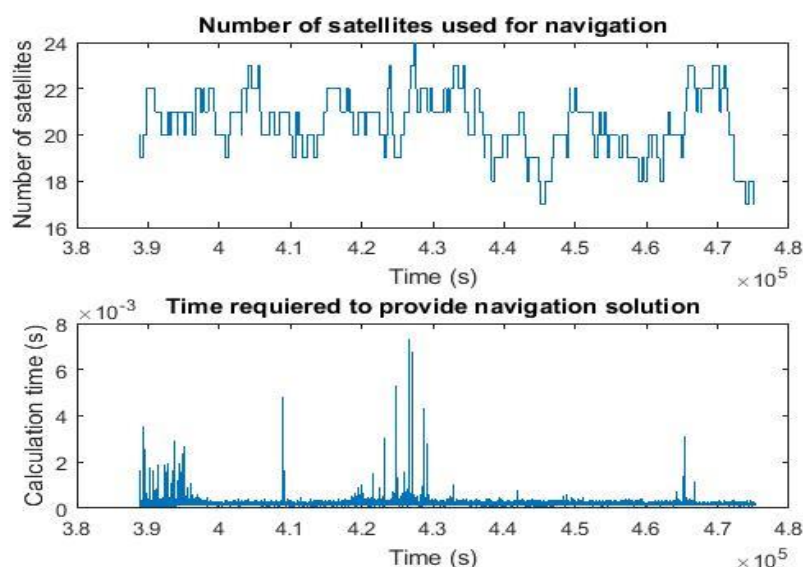


Figura 18: Tiempo necesario para el cómputo de la solución de navegación según el número de satélites utilizados.

Es posible apreciar que en los instantes de tiempo en los que se cuenta con un mayor número de satélites, el tiempo requerido para el cálculo de la solución es significativamente mayor que en el resto de los casos. Utilizar un número sumamente elevado de satélites para la navegación provocaría que la resolución necesitase de periodos de tiempo exagerados para llevarse a cabo.

Por tanto, se puede concluir que aunque contar con un elevado número de satélites va a mejorar el desempeño del sistema de navegación, no resulta ni necesario ni conveniente hacer uso de todos ellos. Lo cual lleva a la pregunta de qué satélites se deben utilizar para la obtención de la posición.

4.2 Descripción de los algoritmos de selección de satélites

En esta sección se van a describir una serie de algoritmos para la selección de satélites que existen en la actualidad. Los algoritmos en cuestión se implantarán en Matlab para llevar a cabo varias simulaciones para valorar su desempeño.

En su mayoría, los algoritmos de selección de satélites estudiados hasta el momento se pueden colocar en dos grandes grupos. El primero se ha centrado en encontrar un método de calcular el GDOP (u otro factor DOP) de una forma que requiera menos carga computacional. El objetivo es reducir la cantidad de sumas y multiplicaciones del proceso mediante una fórmula en forma cerrada (Teng & Wang, A closed-form formula to calculate geomtric dilution of precision (GDOP) for multi-GNSS constellations, 2015). El otro grupo ha preferido escoger el subconjunto de satélites en función de la distribución geométrica de los mismos en el espacio. (Meng, Zhu, & Wang, A New Fast Satellite Selection Algorithm for BDS-GPS Receivers, 2013).

También se ha propuesto utilizar únicamente el determinante de $(H^T \cdot H)$ en vez de su inversa para cuantificar las variaciones del GDOP. Como se requiere menores cálculos para obtener el determinante en comparación con la inversa, se consigue reducir la carga computacional (Tian, Fu, Zhang, & Zhou, 2013).

4.2.1 Optimal Satellite Selection Algorithm (OSSA)

El OSSA es un algoritmo que asegura alcanzar la solución óptima. La solución óptima se logra mediante la computación del PDOP de todas las combinaciones posibles de satélites. La combinación de satélites que presente el menor PDOP será la solución con mayor precisión, y por tanto la óptima. (Roongpiboonsopit & Karimi, A Multi-Constellations Satellite Selection Algorithm for Integrated Global Navigation Satellite Systems, 2013).

La necesidad de llevar a cabo el cálculo del PDOP para todas las combinaciones posibles de satélites visibles conlleva una elevada carga computacional y un tiempo de computación inaceptable para aplicaciones en tiempo real. Lógicamente, al aumentar el número de satélites a la vista, la cantidad de combinaciones posibles también se incrementa y con ello el tiempo requerido para la obtención de la solución.

El objetivo de la selección de satélites es reducir el número de cuentas que llevar a cabo. Si el método de selección es más costoso computacionalmente que la reducción de satélites, no tiene sentido su aplicación.

4.2.2 Highest Elevation satellite selection Algorithm (HEA)

El algoritmo HEA utiliza la elevación con respecto al receptor de los satélites a la vista para escoger el grupo de satélites para obtener la posición. Este algoritmo selecciona los satélites con mayor ángulo de elevación con respecto al receptor. (Roongpiboonsopit & Karimi, A Multi-Constellations Satellite Selection Algorithm for Integrated Global Navigation Satellite Systems, 2013).

Al no ser necesario el cálculo del DOP, el tiempo de computación necesario para este algoritmo es significativamente menor que en el caso del OSSA. Por otro lado, si los satélites con mayor elevación se encuentran aglomerados en un punto, la geometría resultante para el cálculo de la posición ofrece resultados deficientes.

4.2.3 Maximum Volume Algorithm (MVA)

El algoritmo MVA se basa en el hecho de que el GDOP es inversamente proporcional al volumen del tetraedro formado por vectores unitarios apuntando desde la posición del receptor a los cuatro satélites seleccionados. Se propone por tanto, un algoritmo que pretende maximizar el volumen de un tetraedro formado por cuatro satélites y el receptor.

Inicialmente, este algoritmo calculaba el volumen del tetraedro formado por todas las combinaciones posibles de satélites a la vista. Debido al elevado número de combinaciones posible se necesitaba un tiempo de computación considerable. El algoritmo que se describe a continuación es una evolución del descrito que consigue reducir el número de veces que se lleva a cabo el cálculo del volumen de un tetraedro compuesto por cuatro satélites. (Kong, Mao, & Li, 2014).

El algoritmo consta de cuatro pasos (Roongpiboonsopit & Karimi, A Multi-Constellations Satellite Selection Algorithm for Integrated Global Navigation Satellite Systems, 2013):

S₁: De entre los satélites visibles se selecciona el que cuenta con un mayor ángulo de elevación con respecto a la posición del receptor.

S₂: De entre los satélites visible se selecciona el que tenga un ángulo ($\theta_{S_1S_2}$) cercano a 109.5° con respecto a S_1 .

S₃: De entre los satélites visibles se selecciona el que maximice el volumen del tetraedro (según la Ecuación 4.22) formado por S_1 , S_2 , S_3 y S_3' . Donde S_3' tiene el mismo ángulo con respecto a S_1 que S_3 ($\theta_{S_1S_3}$).

Ecuación 4.22:

$$V = \frac{1}{6} \left[\sqrt{2 \cdot (1 - l_2) \cdot (1 + l_3) \cdot (1 - l_2 \cdot l_3 - m_2 \cdot m_3)} + |m_2 \cdot m_3| \right] \cdot (1 - l_3)$$

Siendo:

$$l_2 = \cos(\theta_{S_1S_2}); m_2 = \sin(\theta_{S_1S_2}); l_3 = \cos(\theta_{S_1S_3})$$

$$m_3 = \frac{\cos(\theta_{S_2 S_3}) - l_2 * l_3}{m_2}; n_3 = \pm \sqrt{1 - l_3^2 - m_3^2}$$

S4: De entre los satélites a la vista se selecciona el que maximice el volumen del tetraedro (según la Ecuación 4.23) formado por S₁, S₂, S₃ y S₄.

Ecuación 4.23:

$$V = \frac{1}{6} \det(S_1; S_2; S_3; S_4)$$

La limitación evidente de este algoritmo es la imposibilidad de seleccionar más de cuatro satélites. Estudios anteriores han probado que utilizar el volumen de un poliedro (para casos en los que se utilicen más de 4 satélites para el cálculo) no es una aproximación razonable para el cálculo de GDOP. (Miaoyan, Jun, & Yong, 2008).

4.2.4 Dartboard DOP Algorithm

Este algoritmo utiliza un diagrama polar con la posición de todos los satélites visibles (D'Angelo, Diez, Fernández, Gabaglio, & Marradi, 2005). Este diagrama se divide en 12 zonas como si se tratase de un tablero.

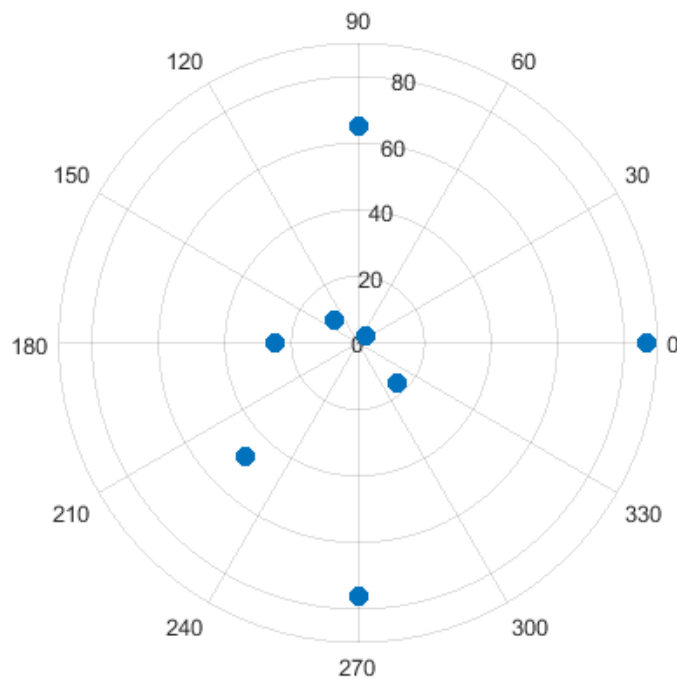


Figura 19: Diagrama polar.

En un diagrama polar se colocan los puntos en función de su elevación y el azimut.

El espacio se divide en 12 zonas según el diagrama polar. Se escogerá el satélite de cada zona que se encuentra más próximo al centro de dicha zona. Para construir el diagrama polar, se debe conocer la elevación y el azimut de cada satélite con respecto al receptor.

La figura que se muestra a continuación representa una forma de dividir el espacio en función de la elevación y el azimut.

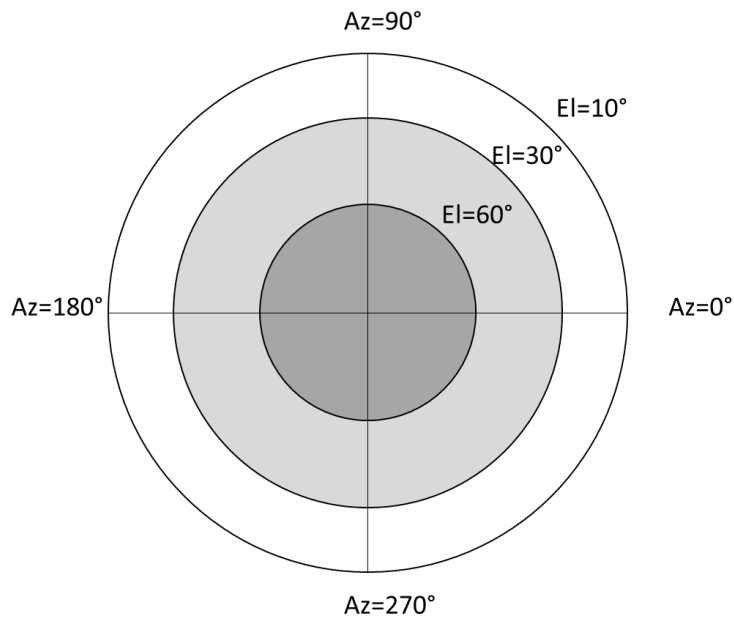


Figura 20: Distribución 1 del espacio algoritmo Dartboard DOP.

Además, se comprobará el desempeño de este algoritmo utilizando otra distribución del espacio, que se muestra a continuación.

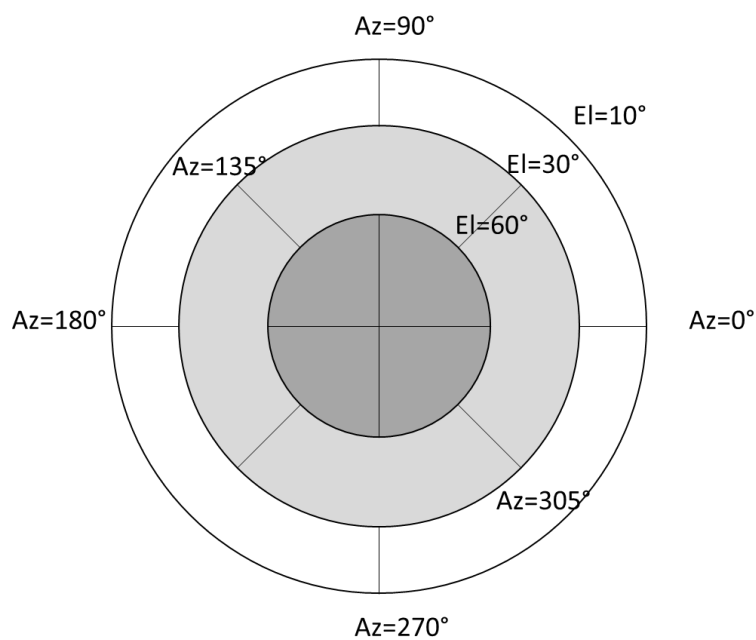


Figura 21: Distribución 2 del espacio algoritmo Dartboard DOP.

El algoritmo en cuestión selecciona el satélite que se encuentra más próximo al centro de cada zona y en el caso de que no hubiese ningún satélite en una zona se escogería el satélite más cercano a dicha zona. De forma que se seleccionan 12 satélites repartidos de la forma más equitativa posible por el espacio aéreo.

(Roongpiboonsopit & Karimi, A Multi-Constellations Satellite Selection Algorithm for Integrated Global Navigation Satellite Systems, 2013).

Este algoritmo presenta una gran inflexibilidad ya que únicamente se puede llevar a cabo para seleccionar 12 satélites exactamente. A pesar de ello, se consigue una geometría más que aceptable con este método.

4.2.5 Controllable weighted PDOP threshold (WPDOP)

Este algoritmo propone una modificación al PDOP de forma que se tenga en cuenta que las varianzas de los errores de rango equivalentes del usuario (*UERE*) no son iguales para los distintos satélites. Por ello se realiza una ponderación del PDOP obteniendo el WPDOP (*Weighted Position Dilution Of Precision*). (Nie, Gao, Wang, & Ji, 2016). (Blanco-Delgado & Nunes, 2010).

A cada medida se le asigna un peso según:

Ecuación 4.24

$$w_i = \frac{1}{\sigma_i^2}$$

Ecuación 4.25

$$\sigma_i^2 = \sigma_{eph}^2 + \sigma_{iono}^2 + \sigma_{tropo}^2 + \sigma_{noise}^2$$

Donde σ_{eph} es la desviación estándar del error de la efemérides, σ_{iono} y σ_{tropo} hacen referencia a la desviación estándar de los errores de la ionosfera y la troposfera, respectivamente, y σ_{noise} representa la desviación estándar del error introducido por el ruido en las medidas. Estos valores se pueden obtener de la información transmitida por los satélites, pero en muchos casos se calcula σ_i^2 en función de la elevación.

Se define:

Ecuación 4.26

$$W = \text{diag}(w_1 \ w_2 \ \dots \ w_n)$$

Ecuación 4.27

$$M = H^T \cdot W \cdot H$$

Mediante el método de mínimos cuadrados:

Ecuación 4.28

$$\Delta x = M^{-1} \cdot H^T \cdot W \cdot \Delta \rho$$

La covarianza de los errores ahora es:

Ecuación 4.29

$$\text{cov}(dx) = M^{-1} \cdot \sigma_{URE}^2$$

El procedimiento llevado a cabo por este algoritmo consta de cuatro pasos:

- 1) Si de entre los satélites visibles uno de ellos es el único perteneciente a su constelación, se debe eliminar de la lista de satélites. Esto se lleva a cabo porque al ser el único, introduce más perjuicio que beneficio al usarlo.
- 2) Cálculo de la matriz M y el valor del PDOP.
- 3) Comprobar si el valor del WPDOP es superior al umbral establecido. En caso afirmativo, el proceso de selección concluye.
De otra forma, se debe calcular el valor de la función de contribución para cada satélite, eliminando el inferior. Después, recalcular la matriz M y el valor del PDOP.
- 4) Comprobar si existe un satélite que es el único de su constelación, para eliminarlo y obtener la matriz M eliminando todos los elementos de la fila y columna correspondientes al satélite eliminado.
- 5) Repetir los pasos 2 y 3 hasta lograr $\text{WPDOP} > \text{Umbral}$.

Cuando se elimina un satélite, existen dos opciones en cuanto a la dimensión de la nueva matriz M: variante e invariante. Si el satélite eliminado no era el único de su constelación la dimensión será invariante, es decir, la nueva dimensión será igual a la antigua. En el caso de que el satélite eliminado fuese el único de constelación, la dimensión será variante. Se puede probar que la eliminación de este satélite no tiene efecto ninguno sobre el valor del WPDOP.

En el caso de contar con unas matrices H_n y W_n en un sistema con n satélites, se decide eliminar el satélite i. Se obtienen las nuevas matrices H_{n-1} , W_{n-1} y M_{n-1} . La matriz M_n pasa a ser:

Ecuación 4.30

$$M_n = M_{n-1,i} + w_i * a_i^T * a_i$$

El valor de la función de contribución para el satélite i será:

Ecuación 4.31

$$J_i = \lambda_i * a_i * M_n^{-2} * a_i^T$$

Siendo:

Ecuación 4.32

$$\lambda_i = w_i + w_i^2 * a_i * M_{n-1,i} * a_i^T$$

La posibilidad de modificar el valor del umbral le confiere a este algoritmo flexibilidad a la hora de ofrecer el nivel de precisión. El algoritmo cambiará el número de satélites seleccionados para aproximarse al valor del umbral establecido y por tanto conseguirá mantener un valor considerablemente constante de precisión.

4.2.6 EPR (Estimación Ponderada por Reducción)

El último algoritmo considerado se presenta como una combinación de 3 de los detallados anteriormente con el fin de explotar sus ventajas de forma conjunta. Este algoritmo ha sido desarrollado internamente en GMV y no se encuentra publicado en ninguna revista científica. Consta de 7 pasos:

- 1) Se descartan los satélites con relaciones señal a ruido (SNR) por debajo de un umbral establecido.
- 2) Se seleccionan cuatro satélites GPS mediante el MVA (Sección 1.1.1).
- 3) Se seleccionan cuatro satélites Galileo mediante el MVA (Sección 1.1.1).
- 4) Se añade a los 8 satélites seleccionados el que cuente con mayor elevación (GPS o Galileo).
- 5) Se completa el conjunto de 12 satélites aprovechando la idea del algoritmo Dartboard DOP (Sección 0). Los 3 satélites restantes se seleccionan de la banda intermedia (fijada entre 25° y 65°) del diagrama polar.
- 6) Dentro de la banda intermedia se seleccionan un máximo de 10 satélites con mejor relación señal a ruido (SNR).
- 7) Los 3 satélites restantes se seleccionan de entre los 10 candidatos aplicando el OSSA (Sección 4.3.1.1).

Este algoritmo se aprovecha del algoritmo MVA para seleccionar 8 satélites (4 de cada constelación) sabiendo que ofrecen factores DOP reducidos y próximos al óptimo. Mediante el uso del algoritmo Dartboard DOP Algorithm, se asegura una geometría espacial favorable. Finalmente, se lleva a cabo una búsqueda exhaustiva para completar el conjunto con los satélites que ofrezcan la mejor solución posible.

El algoritmo EPR reduce considerablemente la carga computacional. El algoritmo OSSA requiere de un total de 2704156 cálculos del DOP, mientras que el EPR únicamente lleva a cabo 120.

4.3 Simulación y resultados

En esta sección se van a detallar las simulaciones llevadas a cabo para poner a prueba los algoritmos descritos en la sección anterior. Para validar el funcionamiento de los algoritmos se va a utilizar una simulación de una serie de satélites de las constelaciones GPS y Galileo a lo largo de 24 horas. En dicha simulación, se actualiza la información de los satélites con una frecuencia de 2 Hz. Es decir, se cuentan con 172800 muestras para llevar a cabo las pruebas.

Las simulaciones se realizaron utilizando un simulador desarrollado internamente en GMV en el que ha colaborado el proyectando desarrollando algunos de los módulos de introducción de error. La arquitectura de dicho simulador se presenta en Apéndices en la sección 8.6. El simulador en cuestión emula las trayectorias de una serie de satélites de las constelaciones GPS y Galileo, así como toda la información pertinente a la navegación que transmiten dichos satélites. Estas simulaciones se utilizarán para valorar tanto los algoritmos de selección de satélites como las técnicas de detección de medidas erróneas del capítulo siguiente.

En este trabajo se llevaron a cabo pruebas para tres situaciones distintas. En primer lugar se consideró un receptor estático. En segundo lugar se utilizó un receptor que sigue una trayectoria en dos dimensiones. Por último, se consideró un receptor que sigue una trayectoria en tres dimensiones. Con estos tres casos se pretende comprobar el desempeño de los algoritmos de selección de satélites en las distintas situaciones que se pueden dar en la realidad.

Para cada caso, será interesante examinar los valores del DOP que proporcionan los distintos algoritmos y los errores de posición. Además, se compararán los recursos computacionales que requiere cada algoritmo para llevar a cabo la selección. Para realizar esta comparación, se ha tomado cuenta del tiempo que necesita cada algoritmo para llevar a cabo la selección y se han calculado los valores medios. Con ello, se va a valorar si los algoritmos son capaces de reducir el tiempo de computación.

También se inspeccionará el número de veces que cada algoritmo cambia el conjunto de satélites seleccionados a lo largo del tiempo de simulación. Con esto, se pretende observar si los algoritmos se encuentran constantemente variando el conjunto de satélites o si por el contrario, después de una primera selección no requiere de cambios frecuentes. Este aspecto es de interés ya que se debe comprobar que los filtros de suavizado con fase no se están constantemente reiniciando, ya que requieren de cierto tiempo para mitigar el ruido.

Por último, se examinará el número de satélites utilizados por cada algoritmo de selección. Aunque en la mayoría de casos esta cantidad es fija (4 ó 12), en el caso del algoritmo WPDOP este número varía ya que el algoritmo está condicionado por un umbral.

En la siguiente figura se puede observar la cantidad de satélites visibles para cada instante de la simulación.

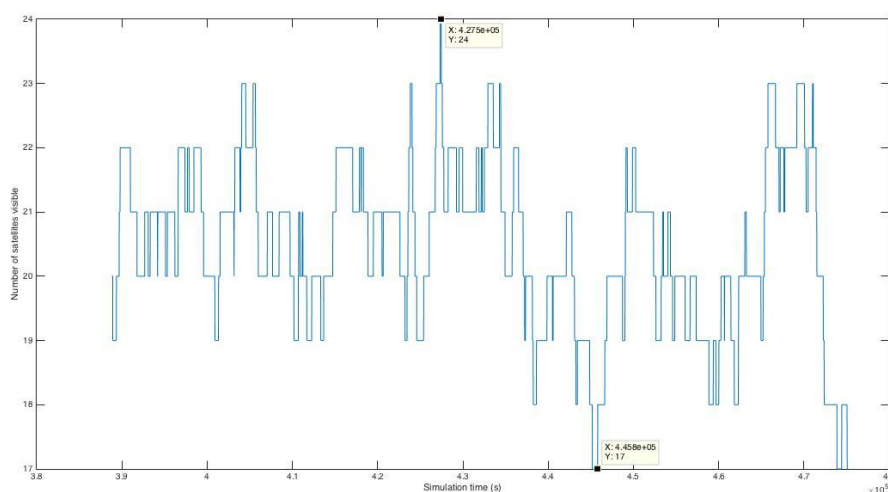


Figura 22: Cantidad de satélites visibles para los algoritmos de selección de satélites.

Como se puede observar en la figura anterior, la simulación cuenta en todo momento con un número de satélites disponibles que oscila entre los 17 y los 24. A continuación, se muestra el número de satélites de GPS y de Galileo en cada instante de tiempo.

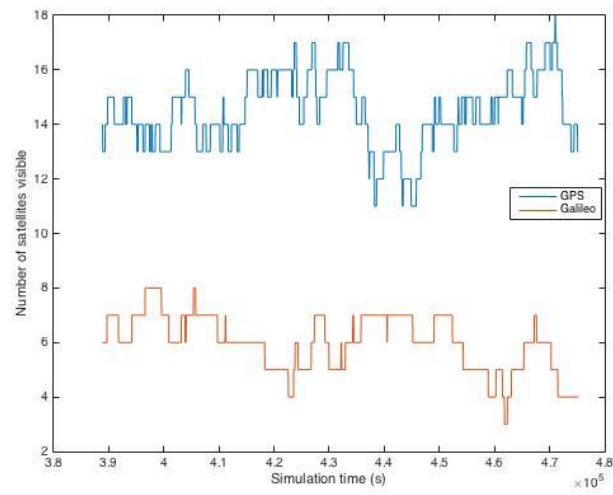


Figura 23: Cantidad de satélites visibles según la constelación.

4.3.1 Caso estático

El receptor de las señales de estos satélites en la simulación se encuentra situado en las oficinas de GMV en Tres Cantos, con coordenadas: (40.591363889°, -3.706916667°, 765m).

La Figura 24 se muestra los valores del PDOP obtenidos en el caso de utilizar todos los satélites disponibles en cada momento. Por tanto, los valores mostrados representan los valores mínimos del PDOP que se pueden obtener en los distintos instantes de la simulación. Los valores del PDOP que ofrecerán los algoritmos de selección de satélites no podrán ser inferiores a los de la Figura 24.

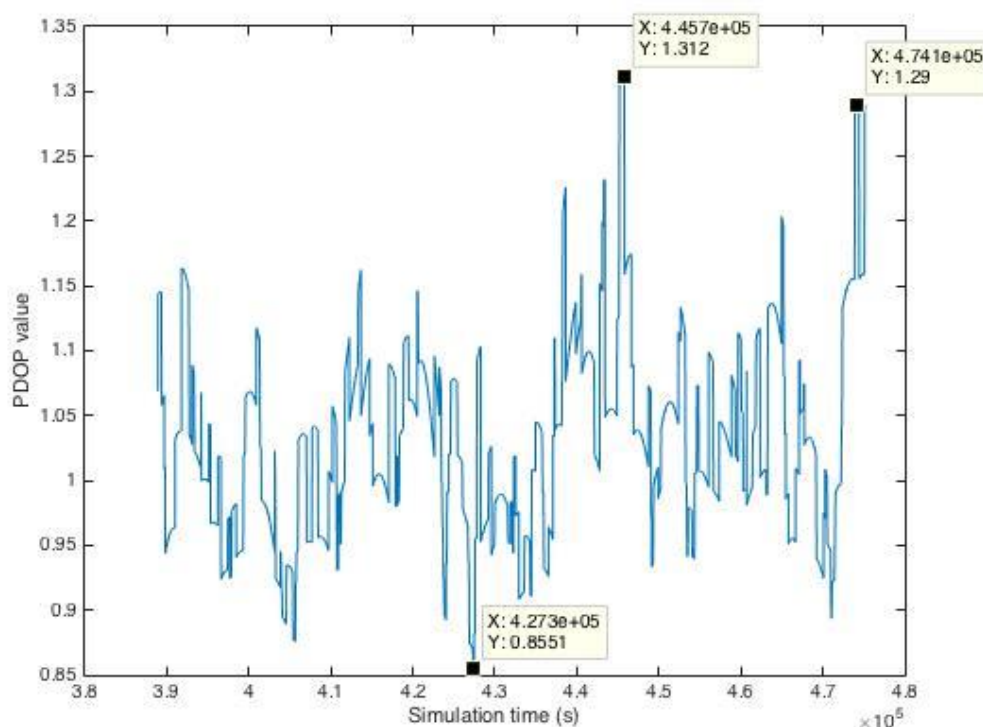


Figura 24: Valores del PDOP utilizando todos los satélites disponibles en cada momento.

En la figura se encuentran marcados tres puntos, siendo estos los valores máximos y mínimos del PDOP. Estos puntos corresponden a los momentos de la simulación en los que se tienen un menor y mayor número de satélites disponibles, respectivamente (Figura 22). Esto demuestra lo mencionado en secciones anteriores: un mayor número de satélites disponibles provoca una mejoría en la precisión, que se demuestra con valores DOP inferiores.

En la Figura 25 se puede observar el error de posición a lo largo del tiempo de simulación cuando se hace uso de todos los satélites visibles para calcular la posición del receptor. Al igual que para los valores del PDOP de la Figura 24, estos valores del error de posición son los mínimos que se pueden obtener.

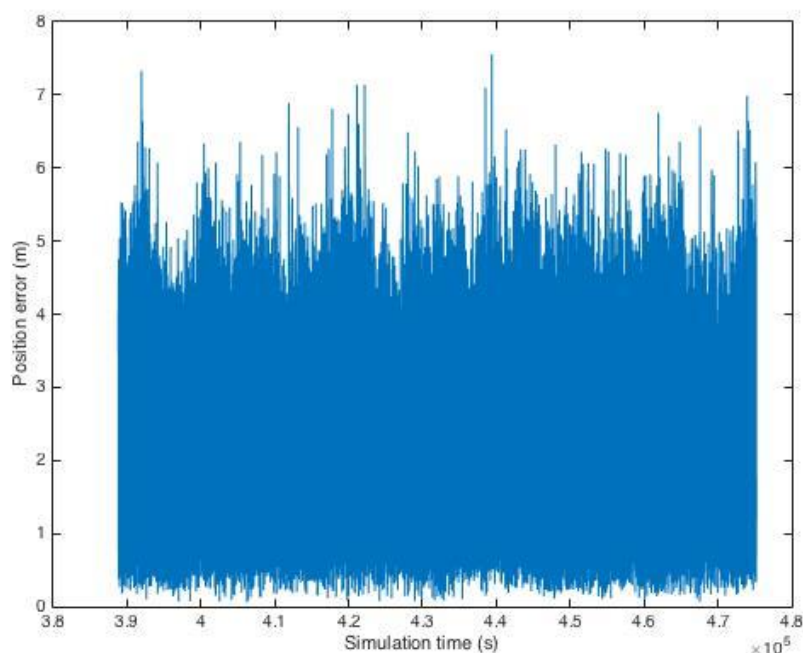


Figura 25: Error de posición en el caso de utilizar todos los satélites disponibles en cada momento.

Para mayor claridad, los errores de posición a lo largo de este trabajo se compararán utilizando la media y la desviación típica de los valores del error. Los valores obtenidos del error de posición tienen una media de 2.11 m y una desviación estándar de 0.87 m. En la Figura 26 se puede observar la función de densidad de probabilidad para este caso. Para el caso de cada algoritmo particular, se hará una comparación con esta función para valorar la distribución del error de posición.

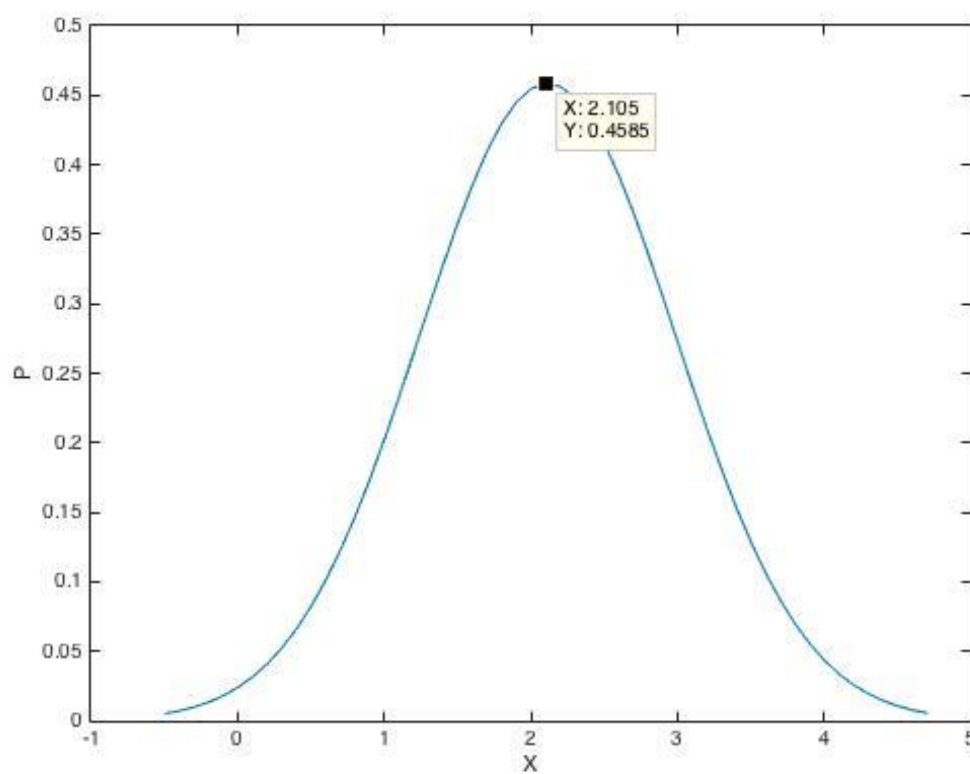
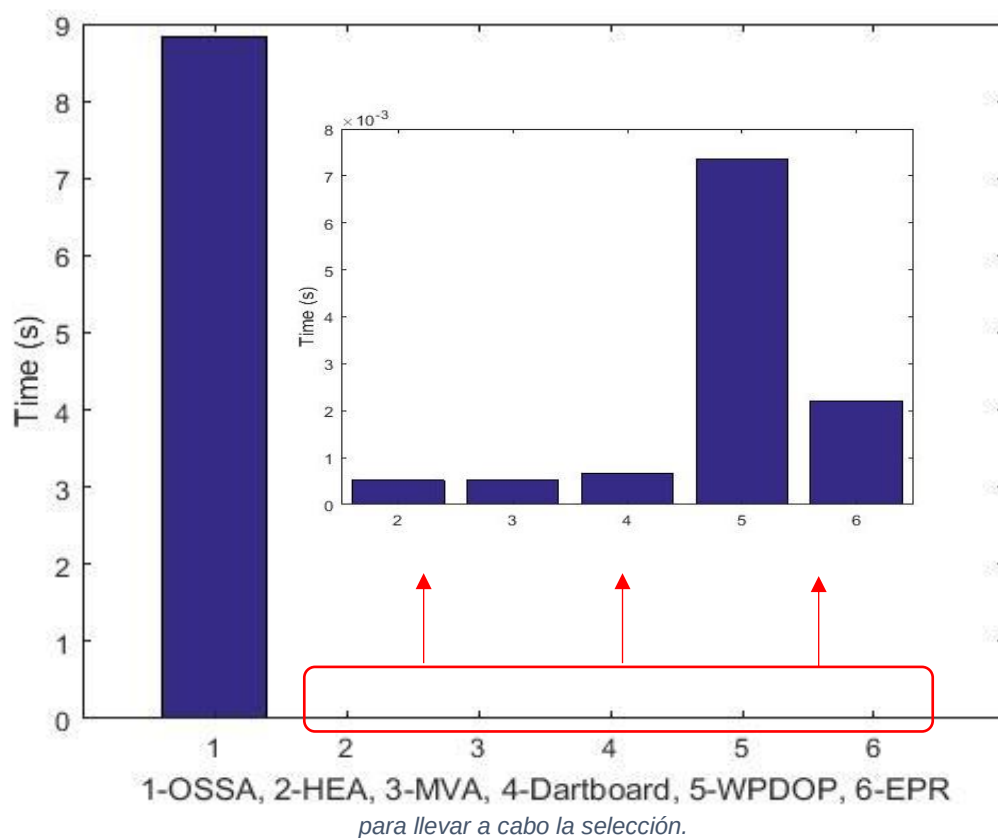


Figura 26: Función de densidad de probabilidad del error de posición para el caso de todos los satélites disponibles.

La

Figura 27 contiene el tiempo medio que requiere cada algoritmo para llevar a cabo la selección del subconjunto de satélites. Como se puede observar, el eje de ordenadas de la parte izquierda de la gráfica hace referencia al algoritmo OSSA, mientras que el eje de ordenadas de la parte derecha hace referencia al resto de algoritmos. Esto se debe a que el algoritmo OSSA necesita de un tiempo mucho mayor que el resto de los algoritmos. El tiempo que necesita cada algoritmo para seleccionar es una medida de la carga computacional.

Figura 27: Tiempo medio requerido por los algoritmos OSSA, HEA, MVA, Dartboard DOP, WPDOP y EPR



Como se puede observar, los algoritmos HEA, MVA y Dartboard son los que necesitan un periodo menor para realizar la selección (aproximadamente medio milisegundo). El algoritmo EPR requiere de aproximadamente 2 milisegundos para llevar a cabo la tarea. Finalmente, el WPDOP necesita aproximadamente 8 milisegundos para seleccionar el subconjunto de satélites. Estos valores no son comparables con el tiempo requerido por el OSSA.

El algoritmo OSSA necesita de aproximadamente 9 segundos para realizar la selección. Obviamente, dependiendo del número de satélites disponibles en cada instante, este algoritmo requerirá de mayor o menor tiempo. Esto se debe a que mayor número de satélites disponibles, existe un mayor número de combinaciones posibles de conjuntos de satélites.

La Tabla 14 contiene el número de cambios de conjunto de satélites que realiza cada algoritmo a lo largo de las 24 horas de simulación. Es decir, la tabla muestra el número de veces que cada algoritmo encuentra un subconjunto de satélites que mejora al subconjunto anterior.

Algoritmo	Número de cambios de conjunto de satélites
<i>HEA</i>	232
<i>MVA</i>	227
<i>Dartboard</i>	296
<i>WPDOP</i>	198
<i>EPR</i>	444

Tabla 14: Número de cambios del conjunto de satélites seleccionados.

Se puede observar que el EPR es el algoritmo que más cambios de conjunto realiza, seguido por el Dartboard DOP. Esta comparación no incluye el algoritmo OSSA, ya que debido a la excesiva carga computacional de esta técnica, se llevó a cabo la simulación para un periodo más corto.

Además se han analizado los tiempos máximos, mínimos (sin contar con los valores nulos) y medios que cada algoritmo permanece sin modificar la selección

de satélites. La tabla que se muestra a continuación contiene los resultados obtenidos.

Algoritmo	Valor máximo (s)	Valor medio (s)	Valor mínimo (s)
<i>OSSA</i>	5810	1134.3864	22
<i>HEA</i>	7824	1481.2704	4
<i>MVA</i>	7824	1513.7982	4
<i>Dartboard</i>	6854	1197.9444	2
<i>WPDOP</i>	8392	1752.3046	4
<i>EPR</i>	5404	783.4364	2

Tabla 15: Valores máximos, mínimos y medios para el tiempo que permanecen los algoritmos sin modificar la selección.

En definitiva, se puede afirmar que el EPR es el algoritmo que más cambios de selección realiza durante el periodo de simulación. Además, se puede comprobar que el valor medio de tiempo que este algoritmo permanece sin modificar el subconjunto es el menor de entre todos los satélites. Existe la posibilidad de mejorar el EPR introduciendo una histéresis en la selección de satélites. Esta modificación queda fuera del alcance de este trabajo.

Por otro lado, el algoritmo WPDOP muestra unas características opuestas a este respecto. Este algoritmo es el que realiza menos cambios de subconjunto de satélites y mantiene dicho subconjunto durante un tiempo medio superior al resto de algoritmos evaluados.

En las secciones que siguen, se van a analizar los algoritmos de manera individual, evaluando el valor del PDOP y la distribución del error de posición.

4.3.1.1 *Optimal Satellite Selection Algorithm (OSSA)*

Como se mencionó en el apartado anterior, este algoritmo logra la solución óptima para un número de satélites pre definido. Para el caso de este trabajo, se fijó el número de satélites para utilizar en el cálculo de la solución en 12. Es decir, este algoritmo llevará a cabo numerosas iteraciones con el objetivo de encontrar el conjunto de 12 satélites que ofrece el menor valor del PDOP.

Debido a la elevada carga computacional que requiere este algoritmo (ver Figura 27) no se ha llevado a cabo la simulación para 24 horas. En cambio, se ha realizado aproximadamente 6 horas. La cantidad de satélites visibles durante el periodo elegido se muestra en la siguiente figura.

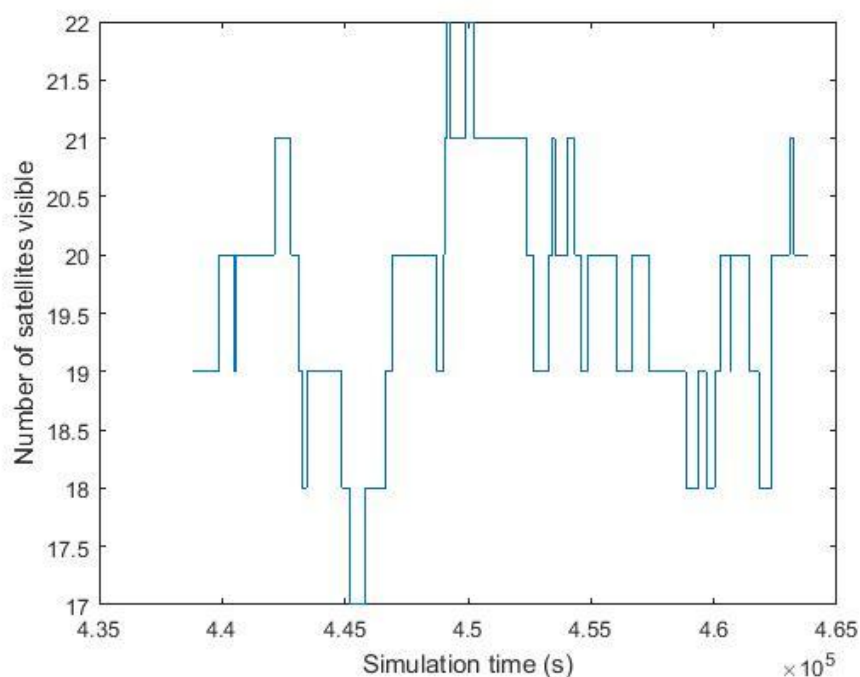


Figura 28: Cantidad de satélites visibles para un periodo de tiempo reducido para el caso del algoritmo OSSA.

El número de satélites visibles varía entre 17 y 22 con el objetivo de que sea posible observar los cambios en el PDOP con la variación del número de satélites disponibles.

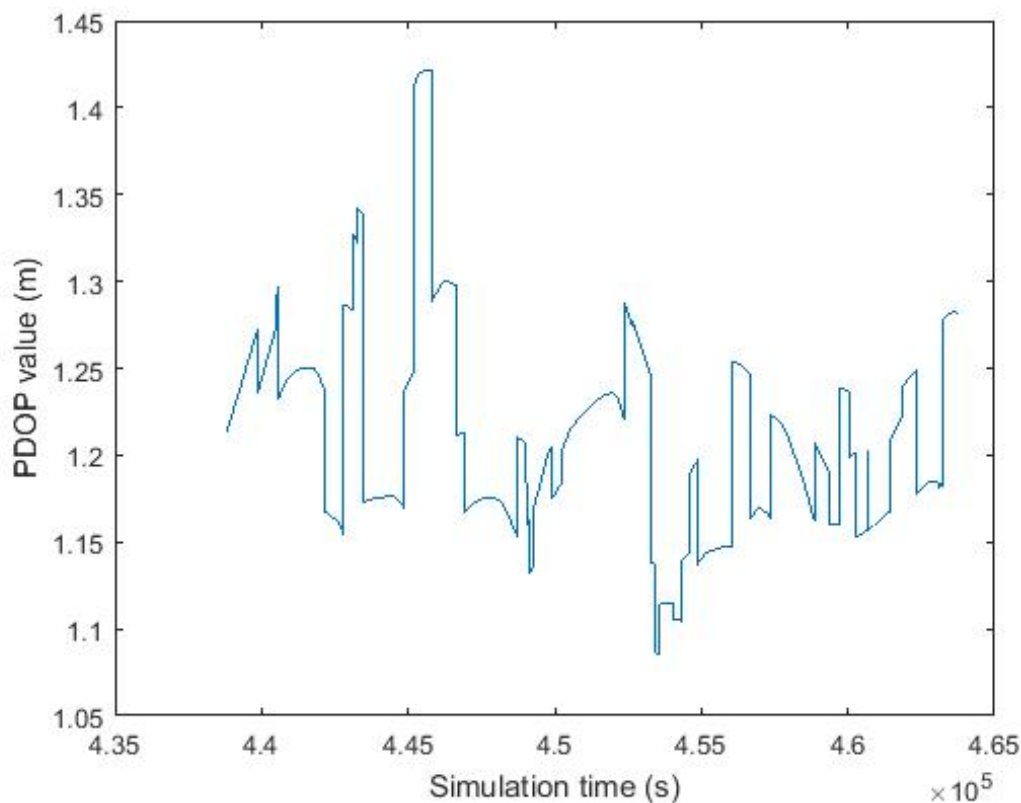


Figura 29: Valores del PDOP para el algoritmo OSSA.

Los valores del PDOP mostrados en la figura anterior representan los valores mínimos de este factor que se pueden lograr utilizando un conjunto de 12 satélites. Se puede observar con claridad como el valor del PDOP responde de forma inversamente proporcional al número de satélites disponibles. En los instantes con menor número de satélites el valor del PDOP alcanza su máximo, y viceversa.

Es apreciable que el valor del PDOP ha sufrido un aumento con respecto al caso con todos los satélites disponibles. Sin embargo, es un incremento considerablemente escaso, teniendo en cuenta que se utilizan entre 5 y 10 satélites menos.

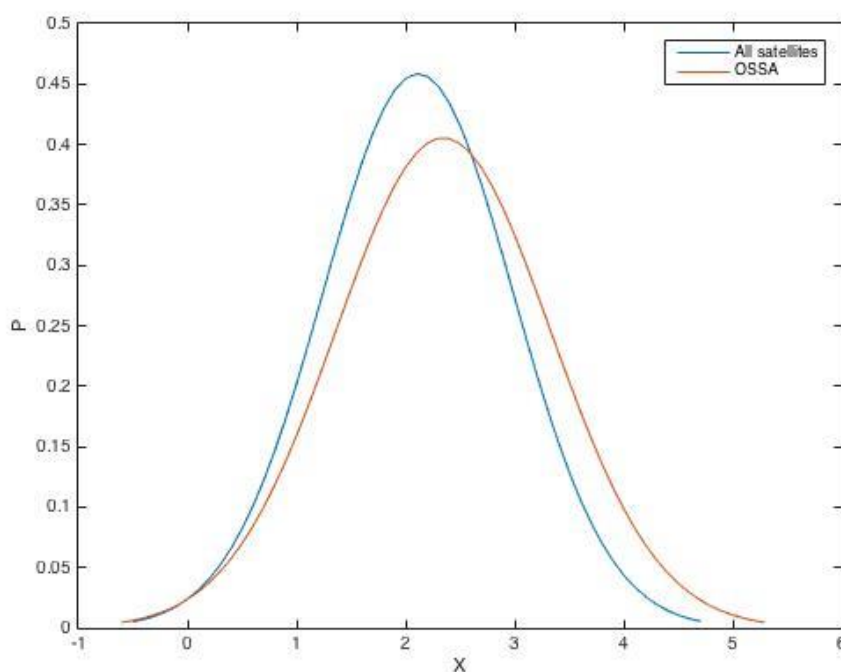


Figura 30: FDP del error de posición para el algoritmo OSSA.

En la Figura 30 se puede observar como la distribución del error de posición es extremadamente similar a la del caso con todos los satélites disponibles. Esto pone de manifiesto que llegado a cierto punto, el aumento del número de satélites no provoca un incremento en la precisión de la navegación que compense el aumento de la carga computacional.

4.3.1.2 *Highest Elevation Algorithm (HEA)*

El algoritmo HEA selecciona los 12 satélites de entre los disponibles que tienen los valores máximos de elevación con respecto al receptor. La Figura 31 muestra los valores del PDOP para este algoritmo a lo largo del tiempo de la simulación.

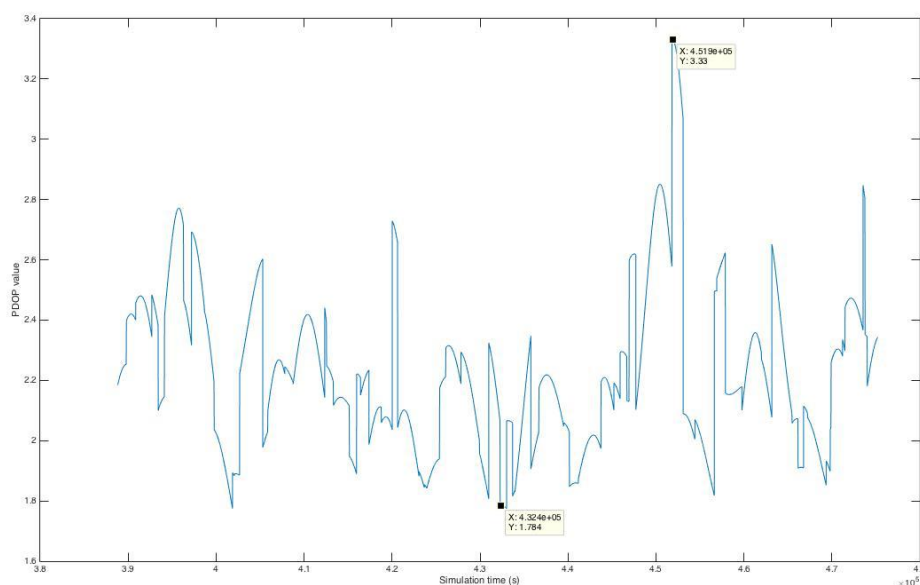


Figura 31: Valores del PDOP para el algoritmo HEA.

Como era de esperar, estos valores son en todo momento superiores al caso utilizando todos los satélites disponibles. Se puede observar que el PDOP alcanza su valor máximo en uno de los instantes de tiempo de número mínimo de satélites visibles.

Como se puede observar en la Figura 32, el valor medio del error de posición con la selección del HEA es de 3.2 m, siendo considerablemente superior al obtenido en el caso de todos los satélites. La desviación típica también es mayor con un valor de 1.6 m.

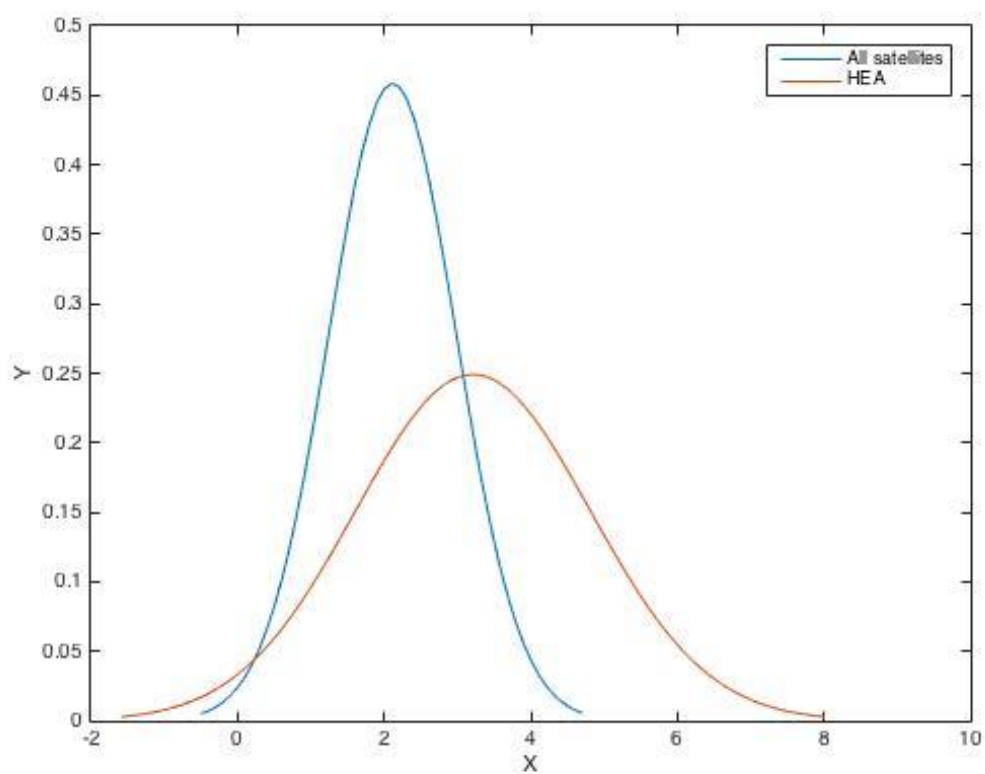


Figura 32: FDP del error de posición para el algoritmo HEA.

4.3.1.3 *Maximum Volume Algorithm (MVA)*

Este algoritmo selecciona el conjunto de 4 satélites que maximiza el volumen del tetraedro que forman los mismos. A pesar de que el algoritmo logra elegir el conjunto óptimo de 4 satélites, se espera que los resultados (valores de DOP y error de posición) sean considerablemente inferiores a los del resto de algoritmos estudiados en este trabajo. Esto se debe a que va a existir una notable diferencia entre el número de satélites que utiliza este algoritmo y los que utilizan el resto de métodos.

La Figura 33 muestra que los valores del PDOP obtenidos son considerablemente superiores a los logrados con el algoritmo HEA y al caso utilizando todos los satélites disponibles. Aun así, alcanza su valor mínimo en el instante en que se tiene un mayor número de satélites visibles y viceversa.

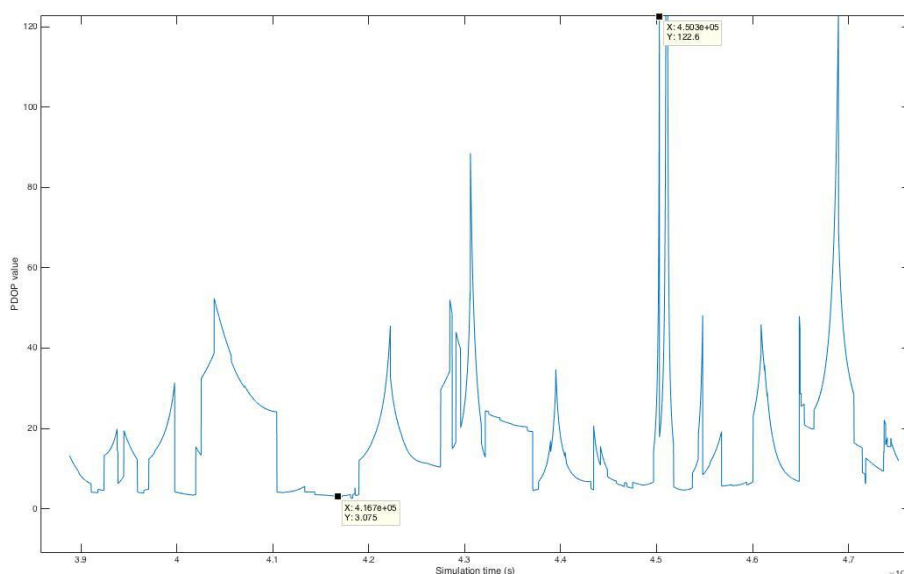


Figura 33: Valores del PDOP para el algoritmo MVA.

La Figura 34 muestra como con el algoritmo MVA se obtiene un error de posición medio de aproximadamente 18 m. Además, el valor de la desviación típica es de 19.3 m.

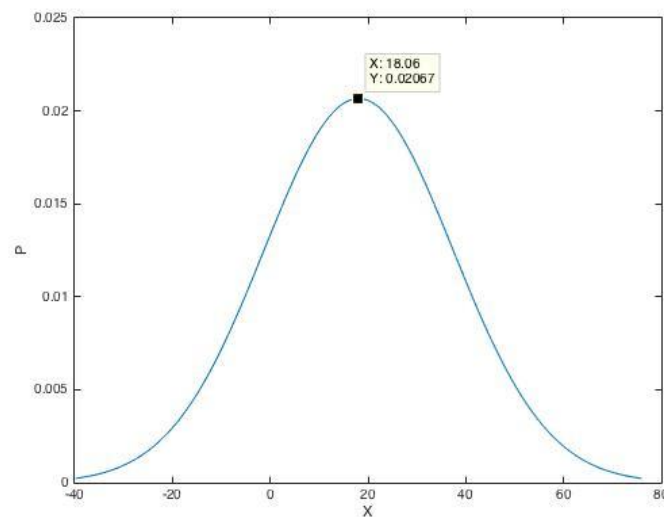


Figura 34: FDP del error de posición para el algoritmo MVA.

Es evidente que existe una diferencia desmedida entre la precisión que es capaz de proporcionar el algoritmo MVA en comparación con el caso de todos los satélites o cualquier otro algoritmo tratado en este trabajo. La principal causa de esta diferencia es que el MVA selecciona únicamente 4 satélites.

4.3.1.4 *Dartboard DOP Algorithm*

La Figura 35 muestra los valores del PDOP para el algoritmo Dartboard DOP. Se puede observar como estos valores son similares a los obtenidos con el algoritmo HEA e inferiores a los del caso con todos los satélites.

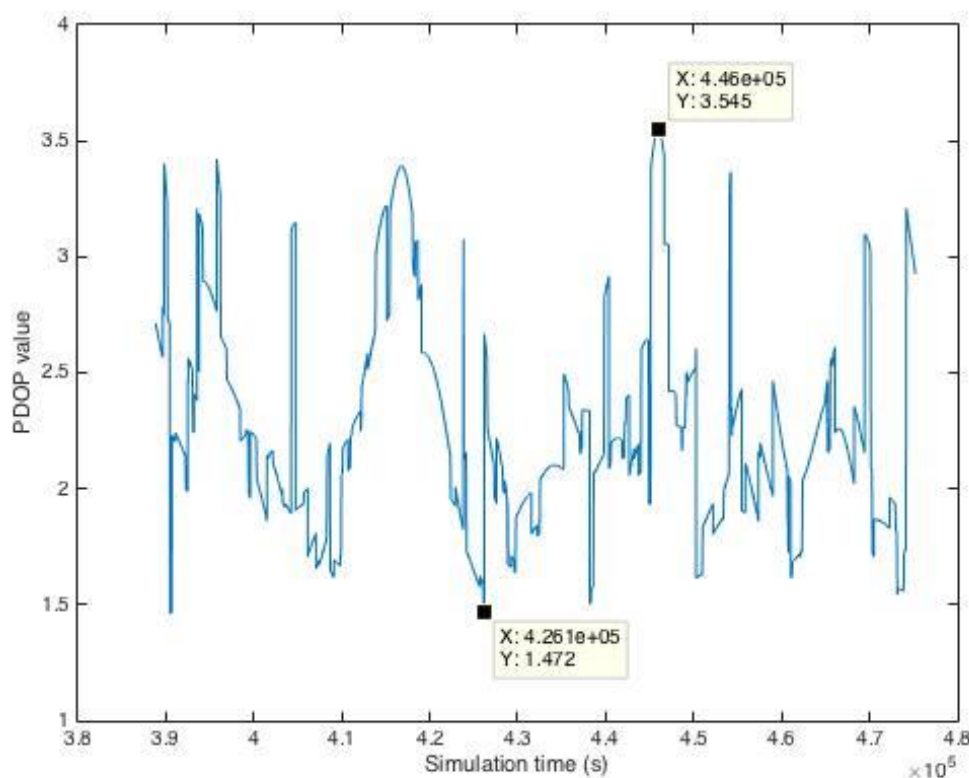


Figura 35: Valores del PDOP para el algoritmo Dartboard DOP.

Cuando se tiene un mayor número de satélites disponibles, este algoritmo obtiene un valor mínimo del PDOP. Esto se debe a que al contar con un mayor número de satélites se tiene una mejor geometría espacial y con ella aumenta la precisión en la solución. En el instante con menor número de satélites visibles ocurre lo contrario.

La Figura 36 muestra como el error de posición tiene un valor medio de 3.4 m, siendo este un valor muy similar al de el algoritmo HEA. La desviación típica de 1.85 m es ligeramente superior al del HEA.

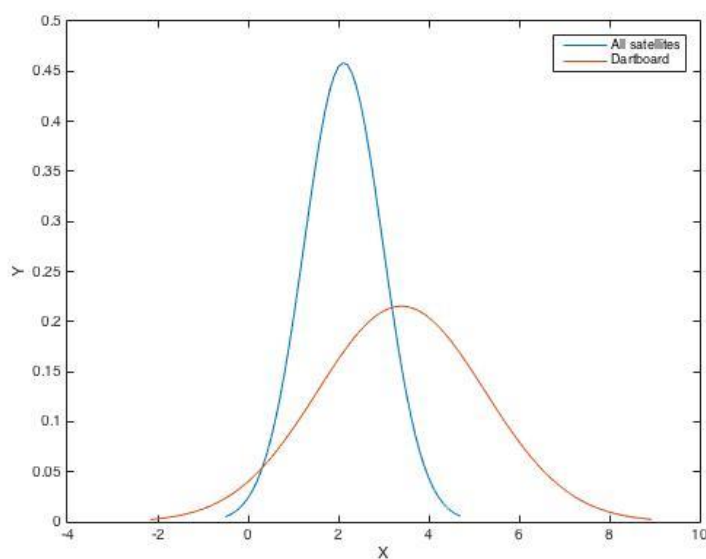


Figura 36: FDP del error de posición para el algoritmo Dartboard DOP.

En la siguiente figura se puede observar una comparación entre el valor del PDOP y el error de posición para el algoritmo Dartboard DOP. Debido a que los valores del PDOP obtenidos con este algoritmo varían significativamente, es posible observar claramente como al aumentar el valor del DOP, aumenta también el error de posición.

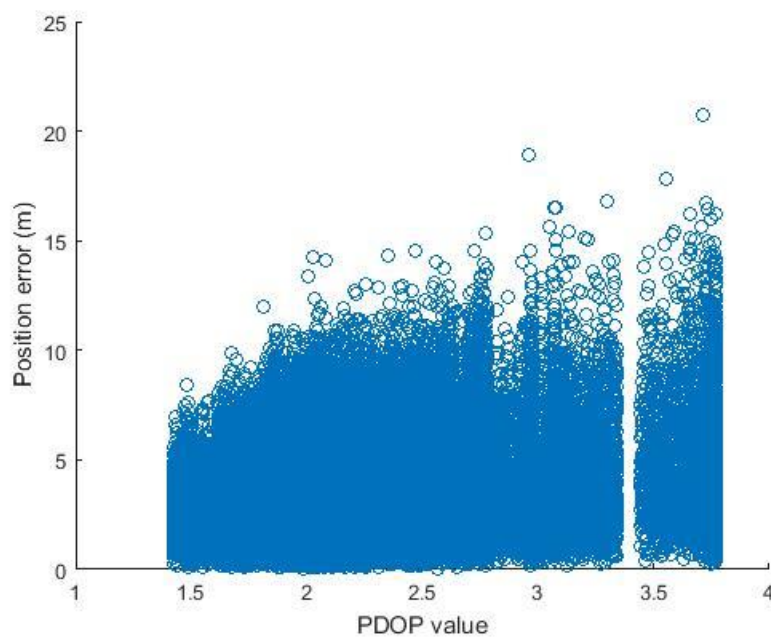


Figura 37: Comparación entre los valores del PDOP y del error de posición para el algoritmo Dartboard DOP.

Como se mencionó previamente, se va a comprobar también el desempeño de un algoritmo Dartboard DOP modificado. La modificación en cuestión está presente en la distribución del espacio que lleva a cabo este algoritmo. La Figura 38 y la Figura 39 muestran los valores del PDOP y la función de densidad de probabilidad obtenida. Es evidente que los resultados son prácticamente idénticos a los del caso Dartboard DOP original.

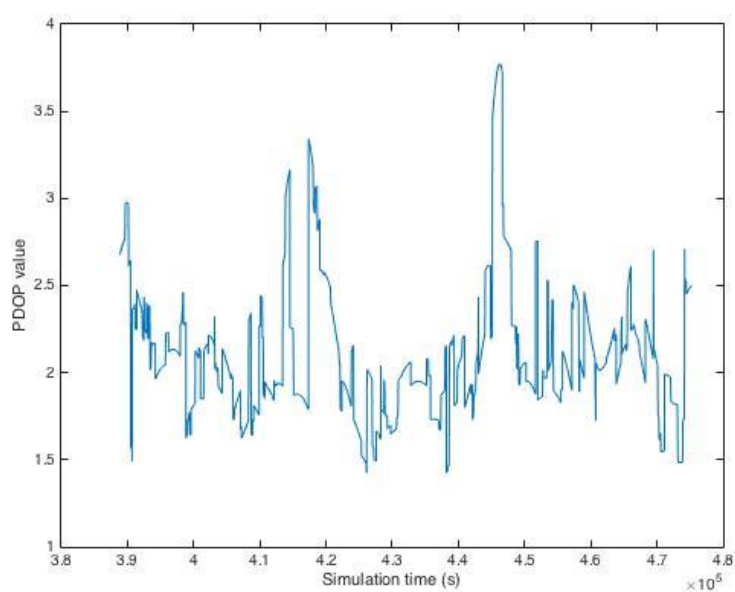


Figura 38: Valores del PDOP para el algoritmo Dartboard DOP modificado.

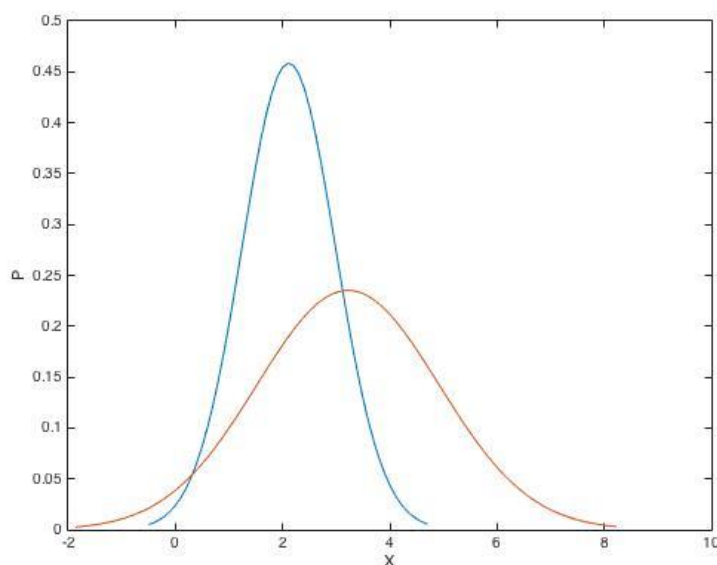


Figura 39: FDP del error de posición para el algoritmo Dartboard DOP modificado.

Debido a la similitud entre las dos versiones del algoritmo Dartboard DOP, en los casos de prueba con un receptor que sigue una trayectoria bidimensional (sección 4.3.2) y tridimensional (sección 4.3.3), se realizará la simulación únicamente con la versión alternativa de este algoritmo (Figura 21).

4.3.1.5 *WPDOP Algorithm*

Este algoritmo requería de un valor de umbral que se establecía para limitar la eliminación de satélites del conjunto inicial. Se estableció este valor en 10, ya que con ello se lograba que el número de satélites seleccionados para el cálculo fuese de 12, como la mayoría de los algoritmos de selección contemplados en este trabajo.

En la Figura 40 se pueden observar los valores del PDOP obtenidos con el algoritmo WPDOP. Estos valores se encuentran contenidos entre 1,3 m y 2,1 m durante el tiempo de la simulación, lo cual supone una menor variabilidad que con los algoritmos comentados en las secciones anteriores.

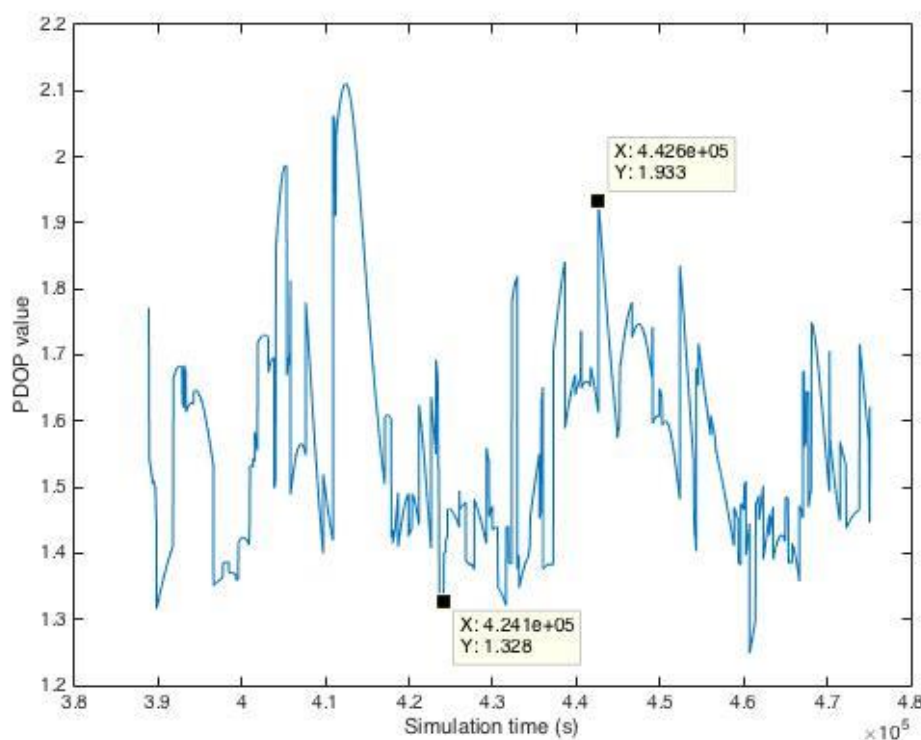


Figura 40: Valores del PDOP para el algoritmo WPDOP.

En la Figura 41 se vuelven a mostrar los valores del PDOP junto al número de satélites que selecciona el algoritmo en cada caso. Se puede observar como el algoritmo WPDOP modifica el número de satélites y consigue mantener el PDOP en un valor prácticamente constante.

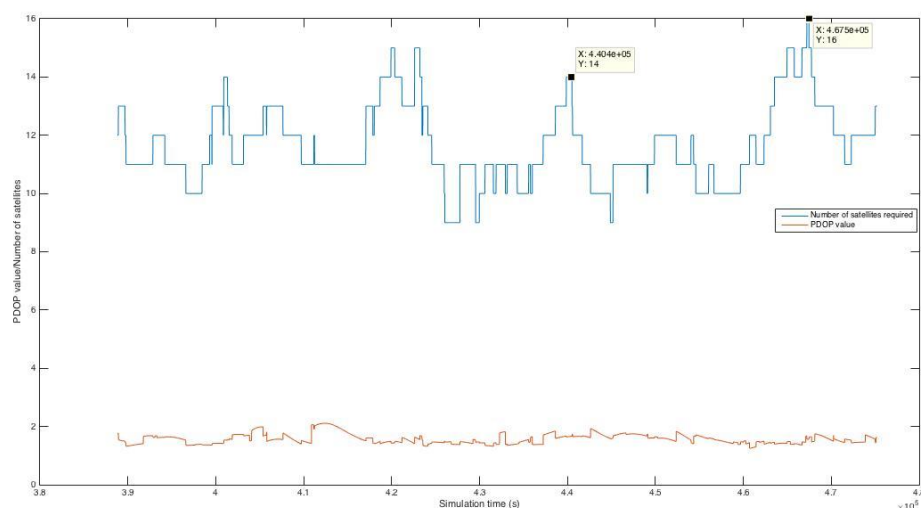


Figura 41: Valores del PDOP y número de satélites utilizados para el algoritmo WPDOP.

El valor medio y la desviación típica del error de posición son de 2.8 m y 1.31 m, respectivamente. En la Figura 42 se puede observar como la distribución del error es más favorable que en los casos del HEA, MVA o Dartboard DOP.

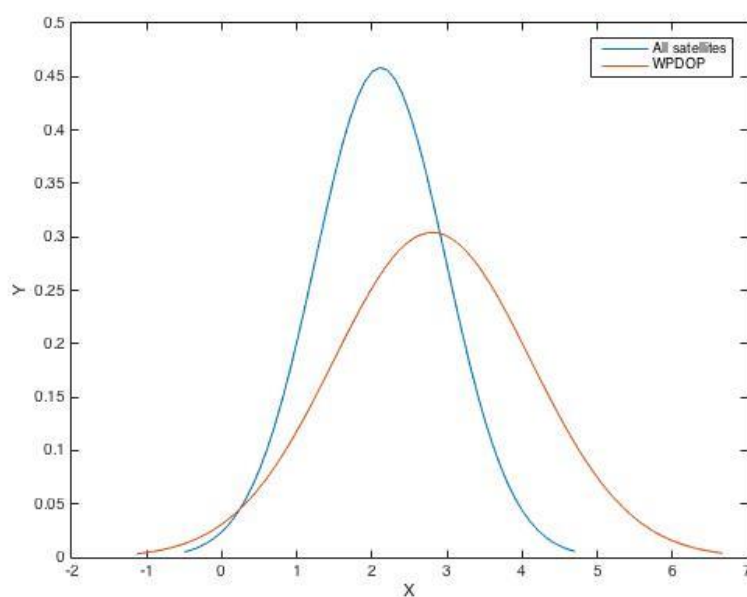


Figura 42: FDP del error de posición para el algoritmo WPDOP.

4.3.1.6 *Estimación Ponderada por Reducción (EPR)*

Los valores del PDOP que se obtienen al utilizar el algoritmo EPR se muestran en la Figura 43. Al igual que en el caso anterior, estos valores están contenidos en un margen considerablemente estrecho (entre 1,2 y 2). Además, los valores obtenidos son inferiores a los logrados mediante los algoritmos anteriores. Obviamente, estos valores siguen siendo inferiores al caso con todos los satélites.

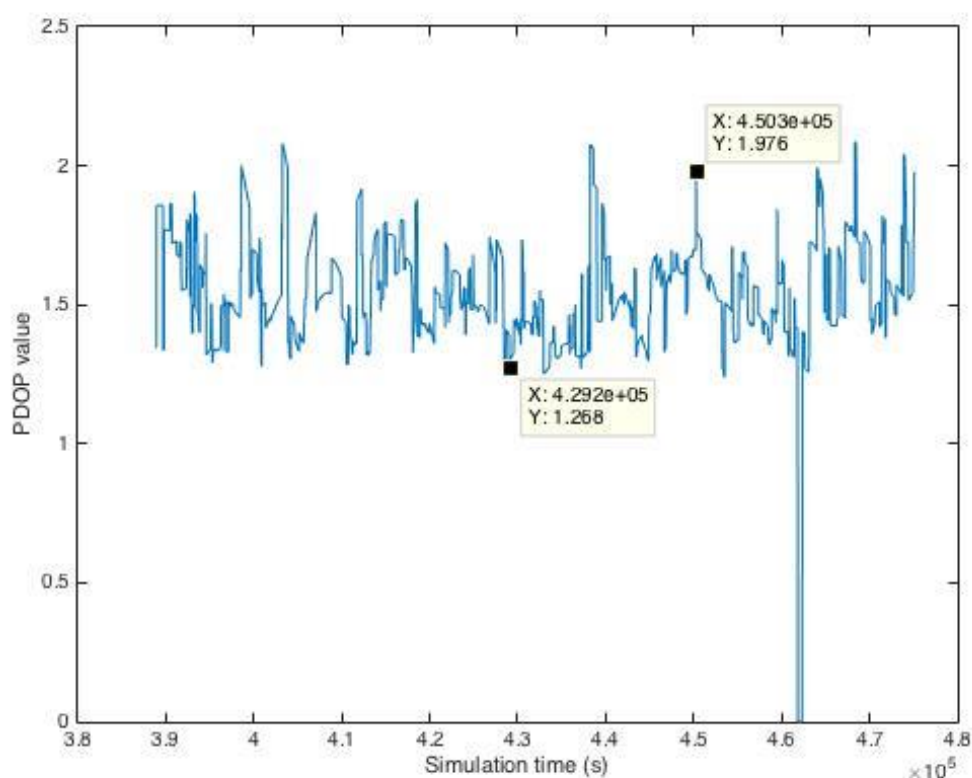


Figura 43: Valores del PDOP para el algoritmo EPR.

Se puede observar cómo alrededor del segundo 4.6 ($s \cdot 10^5$) se obtiene un valor nulo del PDOP. Esto se debe a que durante ese periodo de tiempo el algoritmo no fue capaz de llevar a cabo la selección y por tanto no calculó el PDOP.

Durante esos instantes de simulación, se contaba únicamente con 3 satélites de la constelación Galileo (ver Figura 23). Como el primer paso del algoritmo EPR es llevar a cabo la misma metodología que el MVA para GPS y Galileo por separado, el algoritmo no fue capaz de realizar esta acción para Galileo ya que el número de satélites disponibles para esta constelación era menor que 4.

En la Figura 44 se puede observar la función de densidad de probabilidad del error para el caso del algoritmo EPR. El valor medio (2.72 m) y la desviación típica (1.24 m) de la distribución del error son ligeramente inferiores a los del algoritmo WPDOP de la sección anterior.

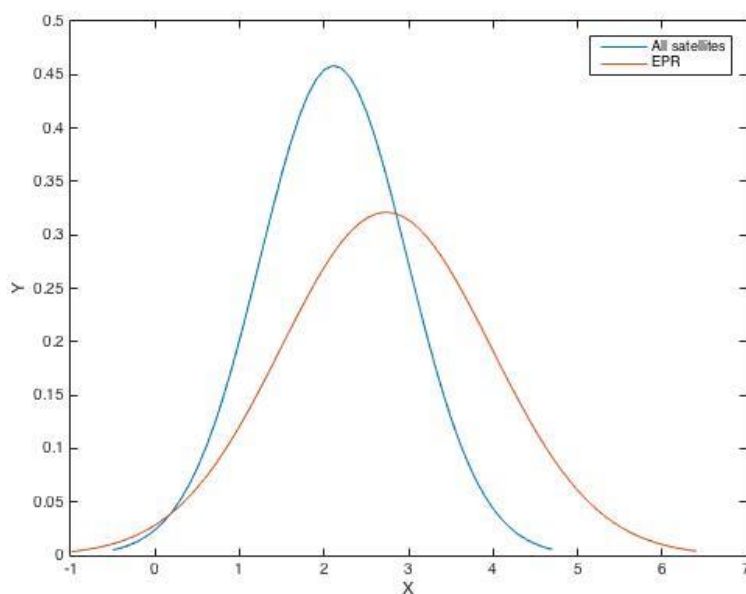


Figura 44: FDP del error de posición para el algoritmo EPR.

4.3.2 Caso 2-D

Además del caso estático comentado en la sección anterior, los algoritmos de selección de satélites se pusieron a prueba con un receptor dinámico. En este primer caso dinámico, el receptor siguió una trayectoria elíptica como se puede observar en la figura siguiente. Los satélites disponibles para este caso son exactamente los mismos que para el caso estático (Figura 22).

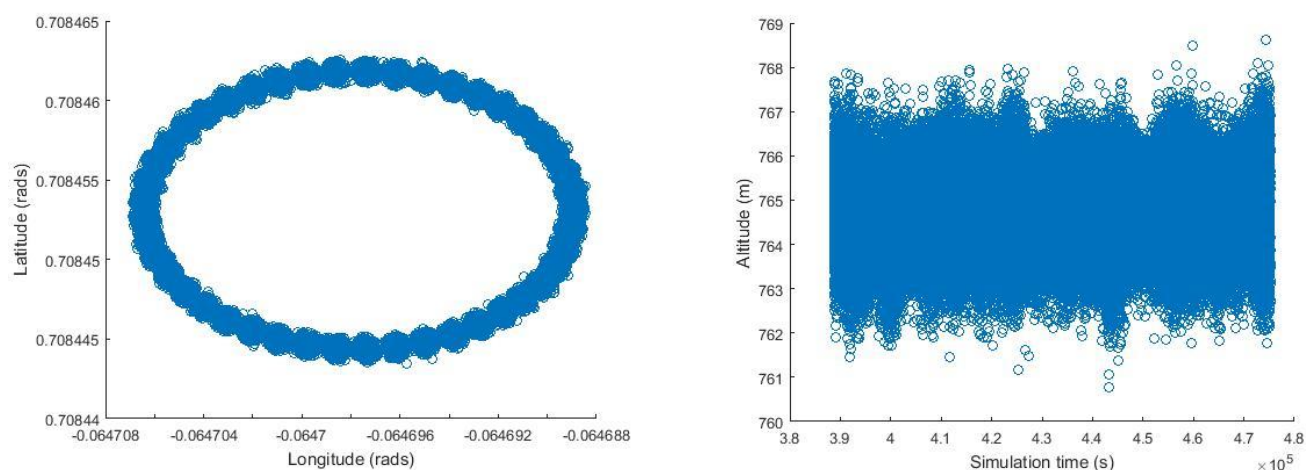


Figura 45: Solución utilizando todos los satélites para el caso de trayectoria bidimensional.

Para ambos casos dinámicos, no se volverá a simular el algoritmo OSSA ya que carece de interés científico y requiere de un tiempo exageradamente elevado para llevar a cabo la selección.

La figura que se muestra a continuación contiene los valores del PDOP obtenidos en el caso de utilizar todos los satélites disponibles para su cálculo. Al igual que en el caso dinámico, estos valores son los mínimos que se pueden alcanzar en la simulación.

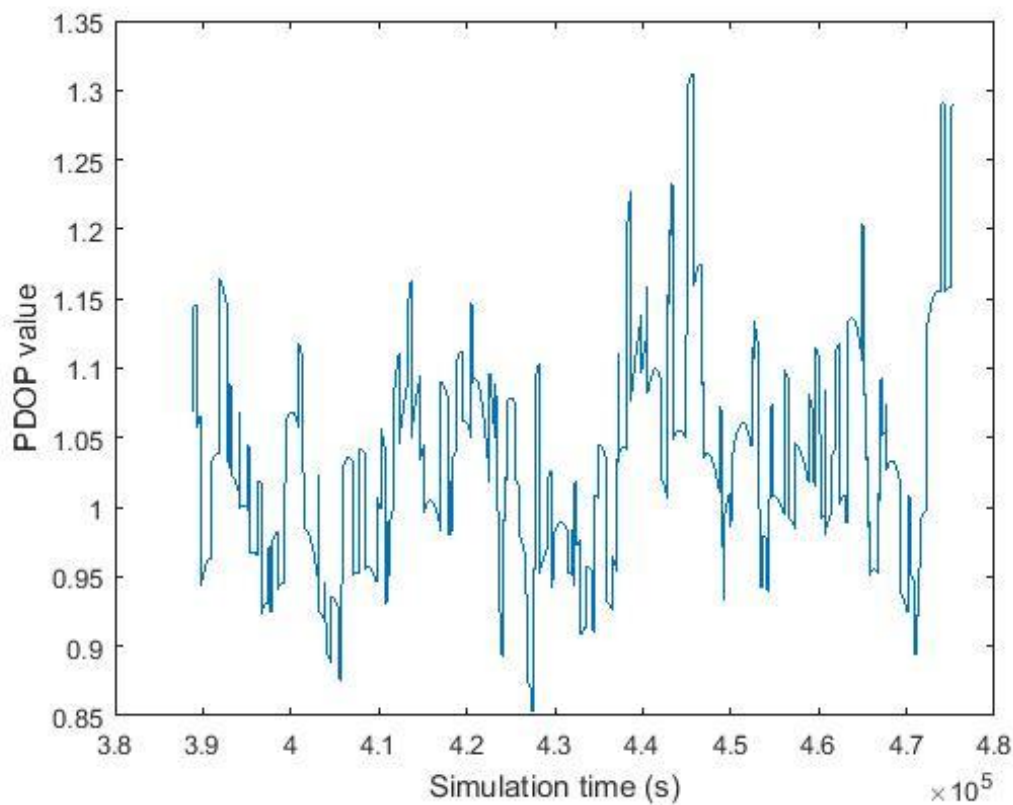


Figura 46: Valores del PDOP utilizando todos los satélites para el caso de trayectoria bidimensional.

Se puede apreciar la similitud entre estos valores y los correspondientes al caso estático. Este hecho se repetirá a lo largo de ambos casos dinámicos, ya que se cuenta con los mismos satélites que en el caso anterior. Se puede observar como la distribución del error del error de posición cuando se hace uso de todos los satélites disponibles es prácticamente idéntica a la del caso estático.

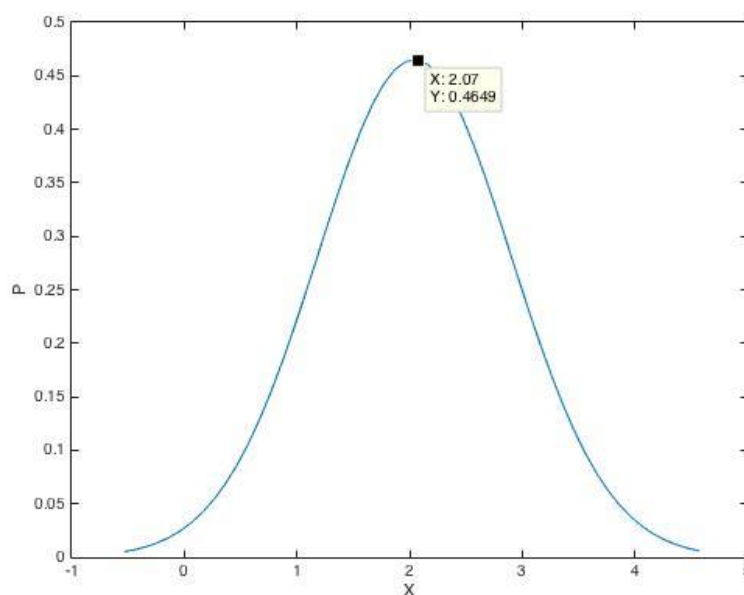


Figura 47: Función de densidad de probabilidad del error de posición utilizando todos los satélites para el caso de trayectoria bidimensional.

Se han vuelto a computar los tiempos medios requeridos por los distintos algoritmos para llevar a cabo la selección del subconjunto de satélites. Los resultados se muestran en la Figura 49. De nuevo, los valores obtenidos son muy parecidos a los del caso estático. El algoritmo WPDOP continúa necesitando de más tiempo para realizar la selección, seguido por el EPR. Los algoritmos restantes (HEA, MVA y Dartboard DOP) requieren de tiempos menores para llevar a cabo la selección.

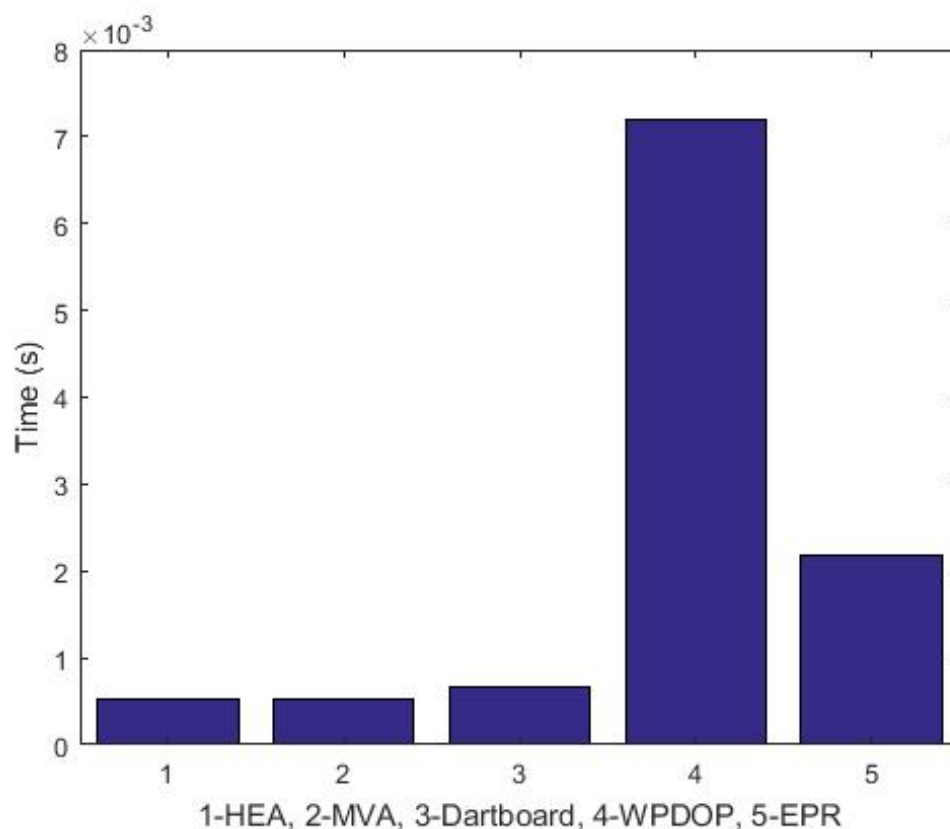


Figura 48: Tiempo medio requerido por los algoritmos para llevar a cabo la selección para el caso de trayectoria bidimensional.

Para completar el análisis, se ha vuelto a examinar el número de cambios de selección y los tiempos máximos, medios y mínimos en los que los algoritmos mantienen una selección fija de satélites.

Algoritmo	Número de cambios de conjunto de satélites
<i>HEA</i>	232
<i>MVA</i>	227
<i>Dartboard</i>	325
<i>WPDOP</i>	196
<i>EPR</i>	438

Tabla 16: Número de cambios del conjunto de satélites seleccionados para el caso de trayectoria bidimensional.

Algoritmo	Valor máximo (s)	Valor medio (s)	Valor mínimo (s)
<i>HEA</i>	7824	1481.2704	4
<i>MVA</i>	7824	1513.7982	4
<i>Dartboard</i>	7848	1067.9566	2
<i>WPDOP</i>	8392	1752.3248	4
<i>EPR</i>	5576	792.469	4

Tabla 17: Valores máximos, mínimos y medios para el tiempo que permanecen los algoritmos sin modificar la selección para el caso de trayectoria bidimensional.

La Tabla 16 y Tabla 17 muestran como, de la misma forma que para el caso estático, el algoritmo WPDOP es la técnica que menos veces cambia de subconjunto y mantiene durante un mayor tiempo la misma selección. El EPR vuelve a ofrecer los resultados más adversos, mientras que el resto de técnicas se sitúan entre este y el WPDOP a este respecto.

Los resultados obtenidos en la evaluación individual para los distintos algoritmo se encuentran contenidos en el apéndice de este trabajo, con el objetivo de no sobrecargar en demasía la memoria y teniendo en cuenta que las conclusiones extraídas de dichas evaluaciones no difieren de los explicado para el caso estático en la sección 4.3.1. Los resultados en cuestión aparecen en Apéndices en la sección 8.1.1.

4.3.3 Caso 3-D

El tercer y último caso que se tratará en este trabajo será el de un receptor que sigue una trayectoria tridimensional. La trayectoria en cuestión se muestra en la siguiente figura.

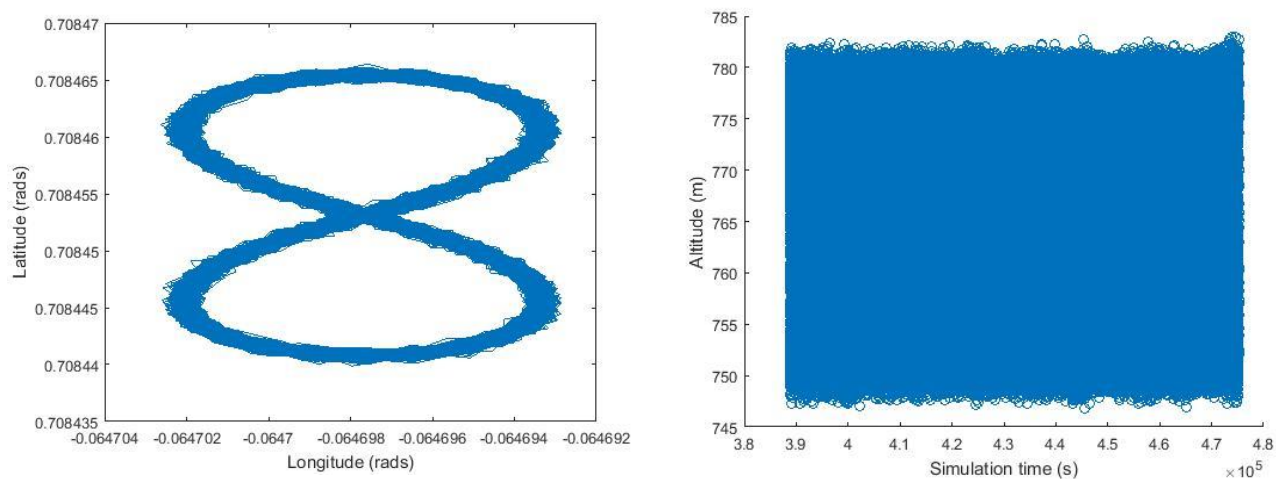


Figura 49: Solución utilizando todos los satélites para el caso de trayectoria tridimensional.

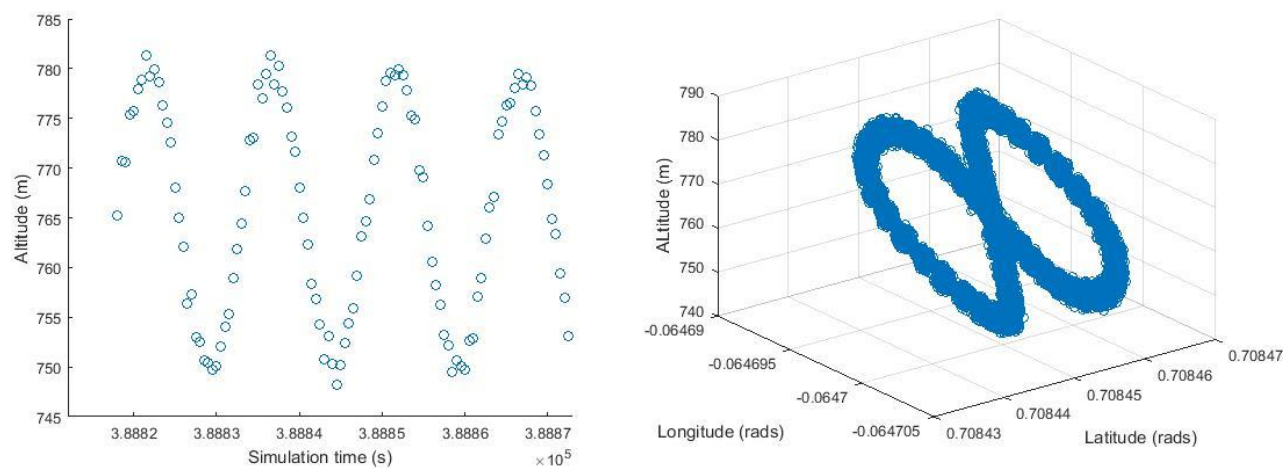


Figura 50: Detalle de la altitud (izquierda). Representación tridimensional de la solución (derecha).

Si se realiza el cómputo del PDOP a lo largo del tiempo de simulación haciendo uso de todos los satélites disponibles, se obtienen los valores representados en la figura que se muestra a continuación.

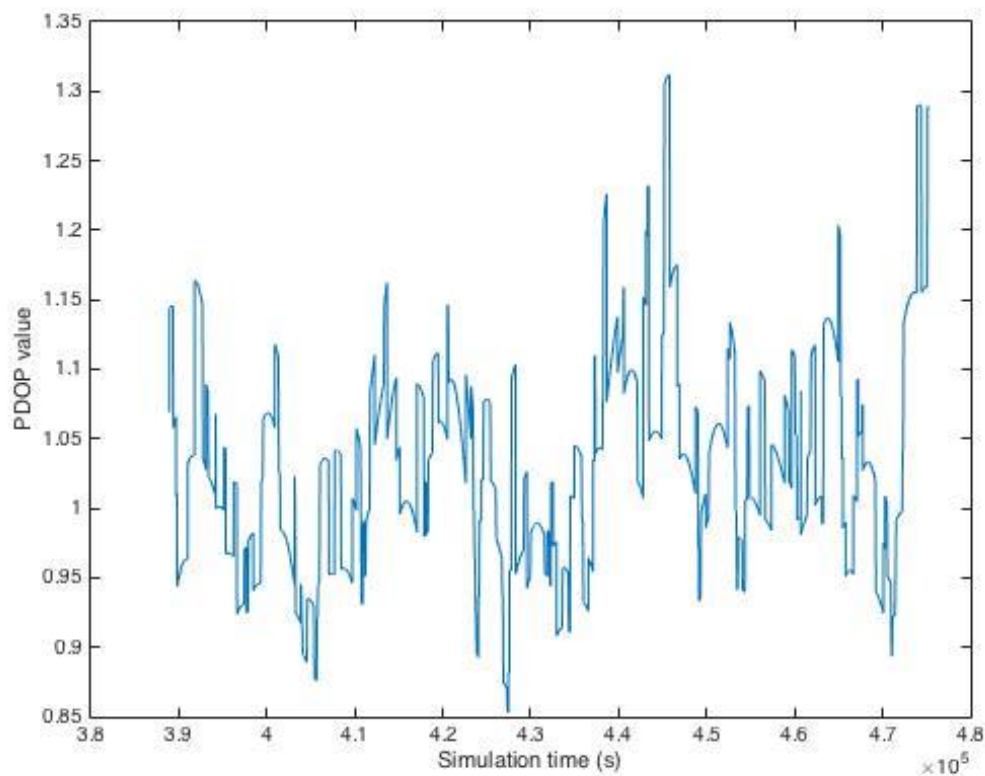


Figura 51: Valores del PDOP utilizando todos los satélites disponibles para el caso de trayectoria tridimensional.

Es posible apreciar la gran similitud entre estos valores y los del caso estático y de trayectoria bidimensional.

Por último, se puede observar en la siguiente figura los tiempos requeridos por los distintos algoritmos se mantienen alrededor de los mismos valores que para el caso estático y bidimensional.

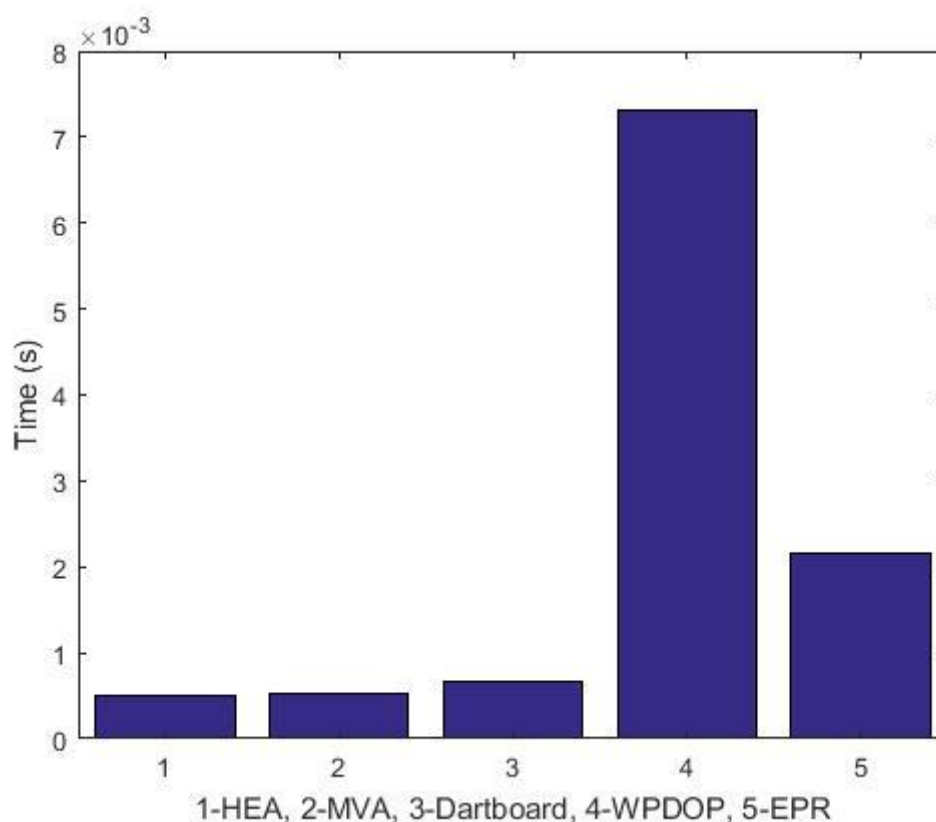


Figura 52: Tiempo medio requerido por los algoritmos para llevar a cabo la selección para el caso de trayectoria tridimensional.

Los resultados obtenidos en la evaluación individual para los distintos algoritmos se encuentran contenidos en el apéndice de este trabajo, con el objetivo de no sobrecargar en demasía la memoria y teniendo en cuenta que las conclusiones extraídas de dichas evaluaciones no difieren de los explicados para el caso estático en la sección 4.3.1. Los resultados en cuestión aparecen en Apéndices en la sección 8.1.2.

4.4 Comparativa y valoración

Las pruebas llevadas a cabo en las distintas situaciones hacen posible realizar una comparación de las prestaciones de los algoritmos presentados.

Como era de esperar, es imposible lograr el mismo nivel de precisión que en los casos con utilizando todos los satélites disponibles. Como ya se aventuró en una de las secciones previas, a mayor número de satélites, menor valor de DOP y menor error de posición.

Aun así, el algoritmo OSSA presenta los resultados más favorables de entre los algoritmos utilizados. Esta era una conclusión previsible que se ha visto demostrada durante las simulaciones. El inconveniente que presenta este algoritmo es la necesidad de contar con unos recursos computacionales considerables. Este se ha visto reflejado en los tiempos medios que requiere el OSSA para llevar a cabo la selección. El tiempo que necesita es exageradamente elevado e incompatible con aplicaciones en tiempo en real.

Al igual que el OSSA, el algoritmo MVA también escoge un conjunto óptimo de satélites. Pero a diferencia del OSSA, esta selección cuenta con un total de sólo 4 satélites. Al utilizar únicamente 4 satélites, el algoritmo hace uso de solamente una de las dos constelaciones disponibles, lo que el priva de las ventajas que acarrea un sistema multiconstelación. Por tanto, la principal desventaja del MVA es que limita la selección a 4 satélites, lo que provoca valores del DOP significativamente más elevados que para el resto de algoritmos.

En las simulaciones llevadas a cabo en este trabajo, se contaba con una cantidad reducida de satélites de la constelación de Galileo. Cuando se da el caso de que el satélite con la mayor elevación es de la constelación Galileo, y por tanto es el primer satélite seleccionado, el subconjunto de satélites debe estar compuesto de satélites únicamente de la constelación Galileo. Por ello, la geometría presentada por los satélites en este caso no es la más favorable, lo que se ve reflejado en elevados valores del PDOP.

En cuanto al HEA, este algoritmo necesita de recursos computacionales prácticamente insignificantes debido a su simpleza. Sin embargo, debido a que los satélites con mayor elevación no tienen por qué componer una geometría espacial especialmente favorable, la precisión de la solución puede variar significativamente.

El algoritmo Dartboard DOP puede considerarse una versión mejorada del HEA, ya que además de tener en cuenta la elevación de los satélites, también utiliza su azimut para llevar a cabo la selección. Este algoritmo asegura la selección de un conjunto de satélites con una geometría favorable.

Por otro lado, el algoritmo WPDOP presenta unos valores del DOP y del error de posición muy favorables. Además, la flexibilidad del algoritmo para modificar el número de satélites que utilizar asegura una precisión de valor estable. La posibilidad de modificar el umbral le confiere a este algoritmo la capacidad de adaptarse a diversas aplicaciones que requieran de distintos grados de precisión. Como principal inconveniente se debe destacar el tiempo que requiere para llevar a cabo la selección. Computacionalmente, es menos exigente que el OSSA, sin embargo el tiempo medio de selección es considerablemente mayor que para los algoritmos restantes.

De los algoritmos evaluados en este trabajo, el EPR (Estimación Ponderada por Reducción) es el que más se asimila al OSSA, y por tanto el que ofrece mejores resultados. El conjunto de 12 satélites seleccionado por este algoritmo ofrece valores del PDOP ligeramente superiores al OSSA. Además, lleva a cabo la selección en menor tiempo que el OSSA y el WPDOP, lo que indica que computacionalmente es menor exigente que estos algoritmos.

El algoritmo EPR aprovecha los puntos fuertes de varios de los otros algoritmos tratados en este trabajo. En primer lugar selecciona dos conjuntos óptimos de cuatro satélites de cada constelación, asegurando niveles de precisión elevados. Seguidamente, se incluye en el subconjunto el satélite que cuente con la mayor elevación, que contará con niveles de ruido reducidos, al mismo tiempo que

proporcionará una geometría conveniente. Finalmente, haciendo uso de la idea del algoritmo Dartboard DOP, el EPR selecciona un grupo de satélites que aseguren una geometría adecuada. Esta selección se lleva a cabo comparando los valores del PDOP de todas las combinaciones posibles y por tanto, cerciorándose de escoger el subconjunto óptimo.

En la Figura 53 se pueden observar los diagramas de cajas de los errores de posición para los algoritmos de selección de satélites evaluados en este trabajo. Se puede apreciar como el algoritmo MVA tiene una dispersión considerablemente mayor que el resto de algoritmos, debido a que selecciona únicamente cuatro satélites. Por otro lado, el algoritmo OSSA presenta una dispersión de valores mucho más limitada. Los algoritmos WPDOP y EPR son los que presentan una distribución más similar al algoritmo óptimo.

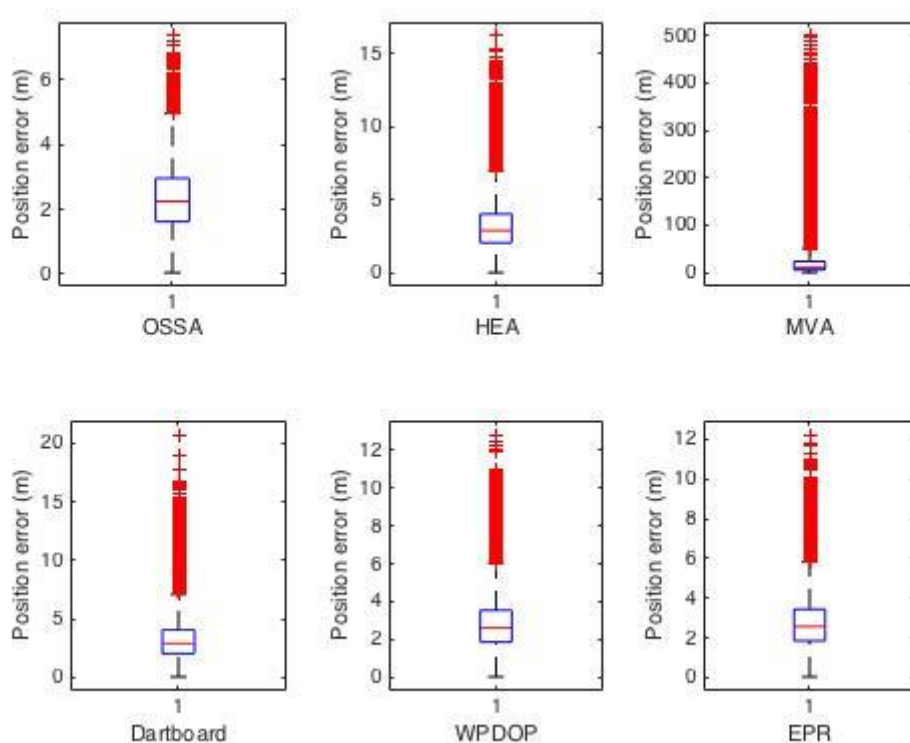


Figura 53: Diagramas de cajas del error de posición para los distintos algoritmos de selección de satélites.

Los valores medios del PDOP presentados en la Tabla 18 vienen a corroborar lo establecido anteriormente. Los valores reducidos del PDOP que ofrecen los algoritmos OSSA, WPDOP y EPR indican una elevada precisión en la solución

de navegación. Al contrario, el algoritmo HEA proporciona una precisión en la solución escasa y esto lo demuestra su valor elevado del PDOP.

<i>Algoritmo</i>	Valor medio PDOP
<i>OSSA</i>	1.2112
<i>HEA</i>	2.2327
<i>MVA</i>	19.2116
<i>Dartboard DOP</i>	2.2720
<i>WPDOP</i>	1.5715
<i>EPR</i>	1.5436

Tabla 18: Valores medios del PDOP para los distintos algoritmos de selección de satélites.

La realización de simulaciones en los casos de receptores que siguen una trayectoria bidimensional y tridimensional han servido como confirmación de que los métodos evaluados en este trabajo para la selección de satélites también son capaces de actuar en estas situaciones, que concuerdan con la aplicación real de los sistemas GNSS. Los resultados ofrecidos por estas simulaciones son idénticos a los del caso estático, con la excepción de un ligero aumento del error de posición en algunos de los casos tratados.

CAPÍTULO 5: Detección de medidas anómalas (RAIM)

En este capítulo se van a tratar los algoritmos de detección de medidas anómalas o erróneas. En primer lugar se introducirán una serie de conceptos básicos sobre la aparición de medidas erróneas y el RAIM (*Receiver Autonomous Integrity Monitoring*). A continuación se entrará en detalle acerca de cómo el RAIM y sus variantes son capaces de detectar medidas erróneas y presentará la noción de nivel de protección. Seguidamente se detallará una metodología distinta al RAIM que busca lograr la detección de medidas erróneas mediante la estimación robusta. Por último se describirán las simulaciones llevadas a cabo utilizando los algoritmos descritos previamente para llevar a cabo una comparación y valoración de sus respectivos desempeños.

5.1 Introducción

Las señales de los sistemas GNSS se pueden ver degradadas por obstáculos naturales y artificiales en ciertos escenarios. En estas situaciones los sistemas GNSS son incapaces de asegurar una solución precisa de forma continua debido a la presencia de errores en el conjunto de medidas.

Mediante la utilización de los sistemas multiconstelación explicados en la sección anterior, se puede aprovechar la redundancia de medidas de los satélites para comprobar la integridad de la solución de posición.

La integridad del sistema se puede definir como la habilidad de identificar disfunciones en el sistema que puedan comprometer el desempeño del mismo. Aunque los propios sistemas de navegación GNSS proporcionan información acerca de la integridad de la solución en sus mensajes, esta información no está disponible en todo momento, por lo que no es suficiente para aplicaciones en tiempo real.

Resulta necesaria la implementación de un algoritmo para detectar y excluir medidas erróneas de un conjunto de medidas que serán posteriormente utilizadas para computar la solución de navegación.

En 1986 se propone el RAIM (*Receiver Autonomous Integrity Monitoring*) para llevar a cabo una comprobación de integridad en los sistemas de navegación por satélite. El funcionamiento del RAIM se basa en comprobar la integridad utilizando las medidas redundantes para llevar a cabo un test de hipótesis con los residuos de los mínimos cuadrados. El RAIM es una técnica que utiliza una solución sobre determinada para llevar a cabo una comprobación de consistencia sobre las medidas de los satélites.

El RAIM se desarrolló para proporcionar una medida de la integridad para las aplicaciones de la aviación civil. Estos primeros algoritmos suponían la presencia de una única medida errónea dentro del conjunto. Esta suposición no es aplicable en áreas urbanas, donde la presencia de obstáculos conlleva la aparición de más de una medida errónea.

Para una sola constelación, el RAIM es capaz de detectar la presencia de una medida errónea utilizando un mínimo de 5 satélites. La técnica FDE (*Fault Detection and Exclusion*) es una extensión del RAIM. Esta técnica no solo detecta medidas anómalas, sino que es capaz de excluirlas del conjunto de medidas, para lo que requiere de un mínimo de 6 satélites.

Existe otra estrategia para la detección de medidas erróneas denominada estimación robusta. Con este método se asegura que los residuos utilizados sean aptos para llevar a cabo la comprobación de integridad. Si los residuos no fuesen válidos, no se podrían utilizar correctamente como estadísticos para la detección de medidas erróneas.

El método de los mínimos cuadrados no responde adecuadamente si la distribución de los errores no es normal. Por tanto, utilizar los residuos de los mínimos cuadrados puede no proporcionar una estimación precisa. Los estimadores robustos se proponen como una alternativa para la detección de medidas anómalas. Estas técnicas son de especial interés ya que se pretende detectar errores que no tiene por qué responder a un modelo de error aleatorio

de media nula como pueden ser desvíos del satélite de su órbita o reflejos por multitrayecto.

En las secciones que siguen, se describe en detalle el algoritmo convencional de RAIM y dos variaciones del mismo: RAIM de separación de solución y nivel de protección basado en isotropía (IBPL). Además, se dedica un apartado a la detección de medidas anómalas mediante estimación robusta. Después, se evalúan las simulaciones llevadas a cabo para comparar el desempeño de los algoritmos descritos.

5.2 RAIM (Receiver Autonomous Integrity Monitoring)

5.2.1 RAIM convencional

Como se ha comentado previamente, las técnicas RAIM utilizan un conjunto sobredeterminado de medidas GNSS para comprobar la fiabilidad de las mismas. Para ello, se lleva a cabo un contraste de hipótesis con el objetivo de revisar la congruencia de dichas medidas (Innac, Bhuiyan, Söderholm, Kuusniemi, & Gaglione, 2016).

Este método utiliza los residuos de las medidas para identificar medidas erróneas. Al igual que en la selección de satélites del capítulo anterior, se parte de las medidas de pseudorango de cada satélite que se rigen por la siguiente ecuación:

Ecuación 5.1

$$\rho_j = \sqrt{(x_j - x_u)^2 + (y_j - y_u)^2 + (z_j - z_u)^2} + c \cdot t_u$$

Siendo (x_j, y_j, z_j) las coordenadas del satélite j , (x_u, y_u, z_u) las coordenadas de la posición aproximada del receptor, c la velocidad de la luz y t_u el offset del reloj del receptor. Siguiendo el mismo procedimiento que en el capítulo anterior en el caso de una única constelación, se obtiene el siguiente sistema de ecuaciones para j satélites:

Ecuación 5.2

$$\Delta\rho = H \cdot \Delta x + r$$

Donde

$\Delta\rho, H$ y Δx ya han sido definidos en la sección 4.1.2.

r es el vector de residuos.

Aunque el sistema cuenta únicamente con 4 incógnitas en el vector Δx , se requiere de un número de medidas $j > 4$ para poder aplicar la metodología RAIM.

Para la resolución del sistema de ecuaciones se recurre a los mínimos cuadrados:

Ecuación 5.3

$$\Delta x = (H^T W H)^{-1} H^T W \Delta \rho$$

La matriz W corresponde a la matriz de pesos utilizada en el capítulo anterior en el algoritmo WPDOP (sección 1.1.1). Suponer que todas las medidas tienen la misma influencia sobre la solución no es una aproximación realista ya que se ven afectadas por diversos errores, por ello resulta conveniente ponderar las medidas con la matriz W .

Los residuos se calculan como la diferencia entre las medidas y sus predicciones. Estos residuos se utilizarán para verificar la consistencia entre las medidas para detectar posibles *outliers*.

Ecuación 5.4

$$r = \Delta \rho - H \Delta x$$

Resulta necesario definir un estadístico (D) para llevar a cabo el denominado “Test Global” en el que se compara el estadístico con un umbral para verificar la consistencia de las medidas y la existencia de medidas erróneas.

Ecuación 5.5

$$D = r^T W r$$

Considerando que los errores de las medidas están distribuidos de forma normal, el estadístico D debe seguir una distribución chi cuadrado con un número de grados de libertad igual al número de medidas redundantes.

El valor del umbral corresponde a la distribución mencionada con un nivel de confianza de $1-\alpha$.

Ecuación 5.6

$$T_G = \chi^2_{1-\alpha, gdl}$$

Si el valor del estadístico D es mayor que el umbral, se rechaza la hipótesis nula H_0 . La hipótesis nula establece que la distribución de los errores elegida es la correcta. Al rechazar la hipótesis nula, se asume la existencia de una inconsistencia en el conjunto de medidas.

Cuando se rechaza la hipótesis nula se debe identificar la medida errónea mediante el denominado “Test Local”. Para ello, el estadístico a utilizar será el vector de residuos normalizados:

Ecuación 5.7

$$w_j = \left| \frac{r_i}{\sqrt{(C_r)_{ii}}} \right|$$

Donde $(C_r)_{ii}$ representa el elemento i de la diagonal de la matriz de covarianza de los residuos que se computa como:

Ecuación 5.8

$$C_r = W^{-1} - H(H^TWH)^{-1}H^T$$

Suponiendo que los residuos estandarizados siguen una distribución normal, se define el umbral según un nivel de confianza $(1-\alpha/2)$.

Ecuación 5.9

$$T_L = N_{1-\frac{\alpha}{2}}$$

Si el estadístico w_j es mayor que el umbral, se acepta la hipótesis alternativa H_a y la medida j se identifica como errónea. En el caso de que no se rechace H_0 , no se puede admitir la solución como válida debido a la inconsistencia entre el test global y el local.

Otra opción para la exclusión de la medida errónea es computar la solución de posicionamiento para todos los subconjuntos eliminando uno de los satélites visibles para cada caso. Se eliminará el satélite que presente el menor valor del estadístico D . Este procedimiento se puede ejecutar de forma repetitiva si se

desea excluir más de una medida errónea, pero resultaría extremadamente costoso.

Utilizando esta técnica es posible detectar la existencia de una medida que contiene erróneas y excluirla del conjunto que se utilizará para computar la solución de navegación. Se puede hacer uso de esta metodología siempre y cuando se cuenten con una redundancia de medidas. Después de llevar a cabo la detección y exclusión de medidas anómalas, resulta interesante valorar la integridad del sistema sin medidas erróneas. Para ello se introduce el concepto de nivel de protección.

5.2.2 Niveles de protección

5.2.2.1 *Definición*

El nivel de protección (PL) es un parámetro de suma importancia para definir la integridad del sistema. El rendimiento de los algoritmos RAIM se mide en función del nivel de protección proporcionado. El objetivo de este parámetro es proporcionar un límite dentro del cual el error de posición está contenido. En la siguiente figura se puede observar una representación del concepto de nivel de protección, que se divide en nivel horizontal y vertical.

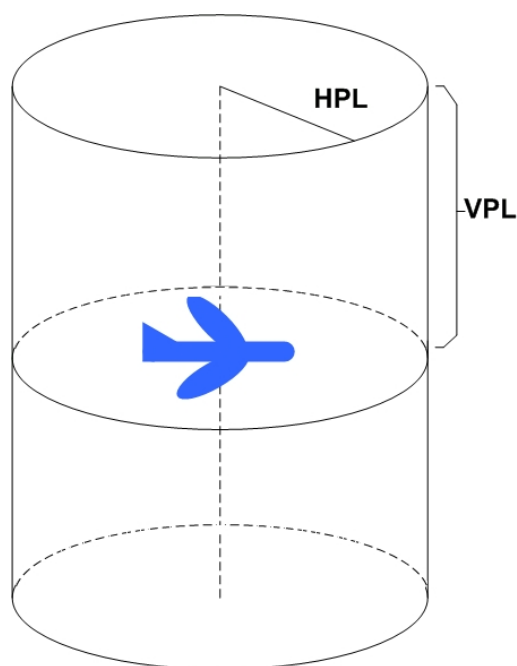


Figura 54: Niveles de protección horizontal y vertical.

El nivel de protección horizontal (HPL) es el radio de un círculo en el plano horizontal que describe la región del espacio que asegura contener la posición horizontal indicada para unas probabilidades de falsa alarma y de no detección. Una falsa alarma es una indicación de fallo de posicionamiento cuando este no ha ocurrido. El nivel de protección vertical se define de la misma forma para el caso de la posición vertical.

El cálculo del nivel de protección se lleva a cabo después de la detección y exclusión de medidas erróneas. Es decir, cuando se computa el nivel de protección para un determinado número de medidas, se supone que en el conjunto de medidas no existe ninguna errónea.

5.2.2.2 Cálculo

Para explicar el proceso de cálculo de los niveles de protección nos debemos fijar en la siguiente figura. El eje horizontal hace referencia al valor del estadístico utilizado, mientras que el eje vertical representa el error de posición obtenido.

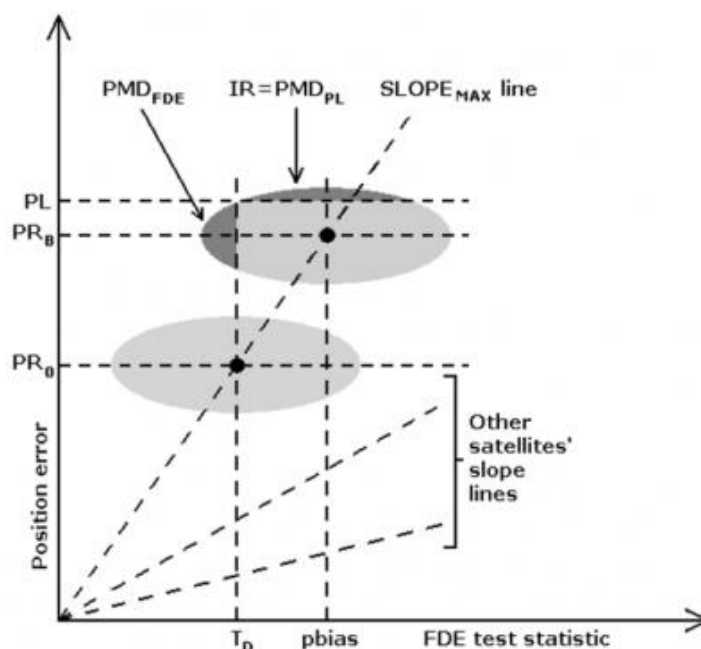


Figura 55: Cálculo del nivel de protección.

Cada satélite se puede representar mediante una recta que hace referencia al error de posición introducido para un tamaño del estadístico por un error en la medida de dicho satélite.

Como se ha explicado anteriormente, el estadístico se compara con un umbral T_D . Si el valor del estadístico es menor que el umbral, se declara que no hay ningún satélite defectuoso. Aunque esto ocurra, puede ser que no todas las medidas estén libres de errores (la contribución del error es lo suficientemente pequeña para pasar el test). Obviamente, la medida que supone un mayor peligro si falla es la que produce el mayor error de posición para un tamaño del estadístico. Es decir, el satélite que tenga mayor pendiente. Con esto, se podría establecer el nivel de protección PR_0 , para el satélite con mayor pendiente y un valor infinitesimalmente inferior al umbral T_D .

A pesar de que únicamente exista un satélite con medidas erróneas, se debe tener en cuenta que las medidas del resto de satélites no son perfectas, y están sujetas ciertos errores que se suelen modelar mediante una distribución Gaussiana con media nula. En la figura, estos errores se visualizan como una elipse de color grisáceo. Como se puede observar, aunque no se sobrepase el umbral T_D , la parte superior izquierda de la elipse supera el nivel PR_0 .

De hecho, la parte izquierda de la elipse inferior representa la probabilidad de no detección. Ya que el valor del estadístico se encuentra por debajo del umbral, pero existen errores en la medida del satélite. Esto supondría un 50% de probabilidad de no detección. Como esta probabilidad es un input para el diseño y se suele fijar a un valor significativamente inferior a 50%, se debe trasladar la elipse sobre la recta hasta el punto en el que la función de probabilidad a la izquierda de T_D sea inferior o igual a la tasa de no detección fijada en el diseño (PMD_{FDE}). Con ello, nos encontramos con la elipse superior del diagrama y el nivel de protección PR_B . El desplazamiento horizontal requerido para lograr esta nueva posición de denominada p_{bias} .

Por último, como la parte superior de la elipse se encuentra por encima de nivel PR_B , se hace uso del riesgo de integridad objetivo (IR) para llevar a cabo un último desplazamiento en la dirección vertical. Este desplazamiento es tal que la función de probabilidad por encima del nivel de protección (PL) es igual a IR.

A continuación se va a explicar el cálculo del nivel de protección para el RAIM convencional de forma numérica. Se requiere como parámetro de diseño el valor del riesgo de integridad (IR). El riesgo de integridad se puede definir como:

Ecuación 5.10

$$IR = P_{nd} \cdot P_{er}$$

Siendo P_{nd} la probabilidad de no detección y P_{er} la probabilidad de error. A su vez, la probabilidad de no detección se puede definir como una distribución gaussiana acumulativa.

Ecuación 5.11

$$P_{nd} = \text{erfc}\left(\frac{\mu - T}{\sigma}\right)$$

Siendo μ igual al valor del p_{bias} mencionado anteriormente, T el valor del umbral y σ el valor de la desviación típica de la solución de posicionamiento.

Ecuación 5.12

$$\sigma = \max\{\sqrt{\text{eig}(A)}\}$$

$$A = (H^T W H)^{-1} H^T W$$

La Ecuación 5.11 se puede formular como:

Ecuación 5.13

$$p_{bias} = T + \sigma \cdot \text{erfc}^{-1}(P_{nd})$$

La probabilidad de error se puede definir como:

Ecuación 5.14

$$P_{er} = 1 - \text{erfc}\left(\frac{PL - SLOPE_{\max} \cdot p_{bias}}{\sigma_x}\right)$$

Discriminando en las dimensiones horizontal y vertical se obtiene las siguientes definiciones de los niveles de protección:

Ecuación 5.15

$$HPL = HSLOPE_{max} \cdot pbias + \sigma_H \cdot erfc^{-1}(1 - P_{er})$$

Ecuación 5.16

$$VPL = VSLOPE_{max} \cdot pbias + \sigma_V \cdot erfc^{-1}(1 - P_{er})$$

Ecuación 5.17

$$\sigma_H = \max\{\sqrt{eig\left(\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}\right)}\}$$

Ecuación 5.18

$$\sigma_V = \sqrt{eig(A_{33})}$$

Las pendientes máximas se obtienen escogiendo la mayor pendiente de entre todos los satélites disponibles.

Ecuación 5.19

$$HSLOPE_i = \frac{\sqrt{A_{i,1}^2 + A_{i,2}^2}}{\sqrt{S_{i,i}}}$$

Ecuación 5.20

$$VSLOPE_i = \frac{|A_{i,3}|}{\sqrt{S_{i,i}}}$$

Ecuación 5.21

$$S = I_n - H(H^TWH)^{-1}H^TW$$

Mediante el algoritmo descrito en la sección anterior es posible detectar y excluir las medidas erróneas del subconjunto si nos encontramos con un sistema de ecuaciones sobredimensionado. El procedimiento explicado se puede repetir reiterativamente si se diese el caso de que existiese más de una medida errónea. Posterior a la detección y exclusión, se propone el cómputo del nivel de protección para valorar el grado de integridad del sistema.

En las secciones que siguen se van a presentar dos variaciones del algoritmo RAIM convencional. La primera variación consigue detectar y excluir múltiples medidas erróneas evitando la iteración del algoritmo. La segunda variación propone una manera alternativa de calcular el nivel de protección.

5.2.2.3 *Diagrama de Stanford*

El diagrama de Stanford es una herramienta que permite revisar la integridad del sistema de una forma rápida y sencilla. El eje de abscisas del diagrama contiene el error de posición, mientras que el eje de ordenadas muestra el nivel de protección (GMV, 2011).

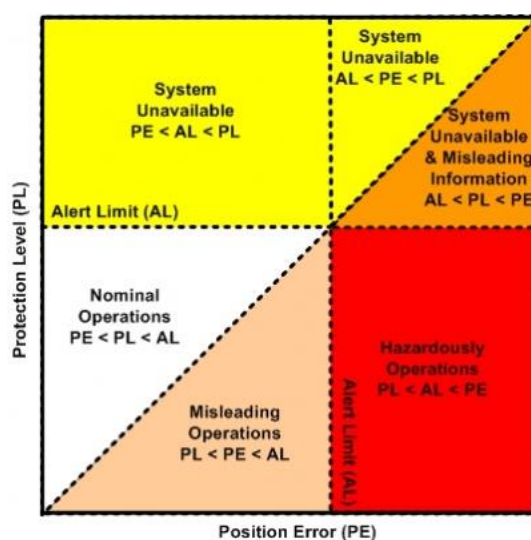


Figura 56: Diagrama de Stanford.

Lógicamente, interesa que los puntos se encuentren en la parte superior a la línea diagonal que atraviesa el diagrama, donde el nivel de protección es superior al error de posición.

Como en el caso de este trabajo no se establece ningún nivel de alerta, se limitará a discernir entre la zona de operación nominal y engañosa. Se hará uso del diagrama de Stanford para valorar la integridad de las técnicas evaluadas en este trabajo.

5.2.3 RAIM con separación de soluciones

El RAIM con separación de soluciones supone una variación del RAIM convencional introducido anteriormente. Este algoritmo se basa en la observación de forma separada de la solución empleando el total de los satélites disponibles y las soluciones referidas a subconjuntos de satélites. Estos subconjuntos están compuestos por todos los satélites excepto uno (Joerger, Chan, & Pervan, 2014).

Para obtener las soluciones mencionadas se hace uso del método de los mínimos cuadrados con pesos (al igual que en el algoritmo WPDOP de la sección 4.2.5).

Ecuación 5.22

$$\Delta x = (H^T \cdot W \cdot H)^{-1} \cdot H^T \cdot W \cdot \Delta \rho$$

Con la matriz H al completo, la solución obtenida se denomina solución completa, y se denotará como $\Delta \bar{x}_0$.

Para calcular las soluciones referidas a cada subconjunto de satélites, se introduce un valor nulo en la diagonal de la matriz W en la posición referida al satélite que se desea eliminar. Las soluciones de los subconjuntos se denotan como $\Delta \bar{x}_l$. Siendo l el indicador de cada subconjunto.

Para cada subconjunto, se computa un discriminador (que se denotará como d_l) que servirá como estadística de test para detectar las medidas erróneas. El discriminador se calcula como la diferencia entre la solución completa y la de cada subconjunto.

Ecuación 5.23

$$d_l = \Delta x_0 - \Delta x_l$$

Si se calcula la diferencia entre la posición real del receptor y la posición obtenida en la solución completa, esta dependerá del error que de la solución completa $\Delta \bar{\rho}$.

Ecuación 5.24

$$\Delta x - \Delta x_0 = -(H^T \cdot W \cdot H)^{-1} \cdot H^T \cdot W \cdot \Delta \rho = -A_0 \cdot \Delta \rho$$

Siguiendo un procedimiento similar, se pueden definir los discriminadores de cada subconjunto como:

Ecuación 5.25

$$d_l = \Delta x_0 - \Delta x_l = (A_l - A_0) \cdot \Delta \rho$$

La matriz de covarianza de los discriminadores se define como:

Ecuación 5.26

$$dP_i = (A_i - A_0)W^{-1}(A_i - A_0)^T$$

Cada discriminador se comparará con un umbral y si este es superado se declarará la presencia de un error en las medidas. Se dividirán los discriminadores en $d_{i,H}$ y $d_{i,V}$ para los componentes horizontal y vertical, respectivamente. Los umbrales se dividirán de la misma forma: D_i y V_i para los componentes horizontal y vertical, respectivamente.

Ecuación 5.27

$$\begin{cases} H_0 \text{ si } \forall i \in [1, N], |d_{i,H}| \leq D_i \text{ y } |d_{i,V}| \leq V_i \\ H_k \text{ si } \forall k \in [1, N], |d_{k,H}| > D_k \text{ o } |d_{k,V}| > V_k \end{cases}$$

Para el caso horizontal, se puede demostrar que:

Ecuación 5.28

$$D_i = \sqrt{\lambda_i} Q^{-1} \left(\frac{P_{fa}}{2N} \right)$$

Donde,

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{inf} e^{-\frac{t^2}{2}} dt$$

λ_i es el mayor autovalor de la matriz $dP_{i,H} = dP_i(1:2,1:2)$

Para el caso vertical, se puede demostrar que:

Ecuación 5.29

$$V_i = \sigma_{V,i} Q^{-1} \left(\frac{P_{fa}}{2N} \right)$$

Donde,

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\frac{t^2}{2}} dt$$

$$\sigma_{V,i} = dP_i(3,3)$$

Por tanto, los valores de los umbrales se calculan a partir de la probabilidad de falsa alarma (P_{fa}) y de la matriz de covarianza de los discriminadores. Se declarará la existencia de una medida errónea en el caso de que el valor del discriminador sea mayor que el umbral.

Con los estadísticos y los umbrales calculados, se procede ahora a computar los niveles de protección correspondientes. La probabilidad de que el error de posicionamiento sea mayor que dicho nivel de protección viene dada por el parámetro de diseño definido como riesgo de integridad (IR).

Ecuación 5.30

$$P(\|\Delta\bar{x} - \Delta\bar{x}_0\| > PL | \exists \text{ un error que no ha sido detectado}) = IR$$

Como se supone existe un error que no ha sido detectado, el valor de los discriminadores debe ser menor al de los umbrales.

Ecuación 5.31

$$|d_{i,H}| \leq D_i \text{ y } |d_{i,V}| \leq V_i$$

Sabiendo que:

$$\Delta x - \Delta x_0 = \Delta x - \Delta x_i + \Delta x_i - \Delta x_0$$

Se puede definir lo siguiente:

Ecuación 5.32

$$\|\Delta x - \Delta x_0\| \leq \|\Delta x - \Delta x_i\|_{H,V} + (D_i | V_i)$$

Por tanto, para calcular el nivel de protección únicamente resulta necesario limitar el valor de $\|\Delta x - \Delta x_i\|_{H,V}$

Para el caso horizontal, se puede demostrar que $\|\Delta\bar{x} - \Delta\bar{x}_i\|_H$ está limitada para un IR por:

Ecuación 5.33

$$\delta_i = \sqrt{\sigma_H \cdot \delta}$$

$$\delta = F_{\chi^2}^{-1}(1 - IR)$$

Donde σ_H corresponde con la definición de la sección 5.2.2 con la diferencia de referirse a la matriz A_i .

Por tanto, se define:

Ecuación 5.34

$$HPL = \max_{i \in N} (\delta_i + D_i)$$

Para el caso vertical, se puede demostrar que $\|\Delta\bar{x} - \Delta\bar{x}_i\|_V$ está limitada para un IR por:

Ecuación 5.35

$$\gamma_i = \sigma_V \cdot Q^{-1} \left(1 - \frac{IR}{2} \right)$$

Por tanto, se define:

Ecuación 5.36

$$VPL = \max_{i \in N} (\gamma_i + V_i)$$

El método RAIM con separación de soluciones es capaz de detectar múltiples errores en las medidas mediante el cómputo de una serie de soluciones según distintos subconjuntos de satélites. El algoritmo se basa en la comparación de las variaciones de las soluciones de posición que provocan la eliminación de los satélites del conjunto de medidas. Como principal inconveniente se debe destacar que requiere de un periodo de tiempo significativo para llevar a cabo la detección y exclusión.

5.2.4 RAIM IBPL (*Integrity-Based Protection Level*)

El RAIM IBPL difiere del RAIM convencional y el RAIM con separación de soluciones en que esta técnica se limita a computar únicamente los niveles de protección, sin detectar ni excluir previamente medidas. Este algoritmo pretende adaptarse a las situaciones de navegación en entornos urbanos, donde muchas de las suposiciones utilizadas en el RAIM desarrollado para la aviación civil (RAIM convencional) no son válidas. Este método ha sido desarrollado internamente por GMV (Azaola, Calle, Mozo, & R.Piriz, 2016).

A diferencia de los algoritmos presentados en las secciones anteriores, el IBPL no implementa mecanismos de detección y exclusión de medidas. El algoritmo se limita a calcular un nivel de protección como los comentados en la sección 5.2.2. El cómputo del nivel de protección se basa en la isotropía del error y permite considerar más de una medida errónea.

El nivel de protección asegura que, para un nivel de confianza de $1 - \alpha$, el error de posición este contenido en la circunferencia de radio de valor el nivel de protección (en metros). Dicho de otra forma, la probabilidad de que el error de posición exceda al nivel de protección es de α (también denominado riesgo de integridad).

Este algoritmo se basa en que la relación entre el vector de residuos estimado mediante el método de mínimos de cuadrados y el error de posición es lineal. El nivel de protección horizontal del IBTL se puede formular como (Azaola, Calle, Mozo, & R.Piriz, 2016):

Ecuación 5.37

$$HPL = k \cdot \|r\| \cdot HDOP$$

Siendo $\|r\|$ el vector de residuos, $HDOP$ el factor de dilución de precisión horizontal y k el ratio de confianza isotrópico. Este ratio asegura que $H \cdot \Delta\bar{x}$ se encuentre limitado por el tamaño del vector de residuos un número k veces con una probabilidad $1 - \alpha$.

Ecuación 5.38

$$P(\|H \cdot \Delta x\| \leq k \cdot \|r\|) \geq 1 - \alpha$$

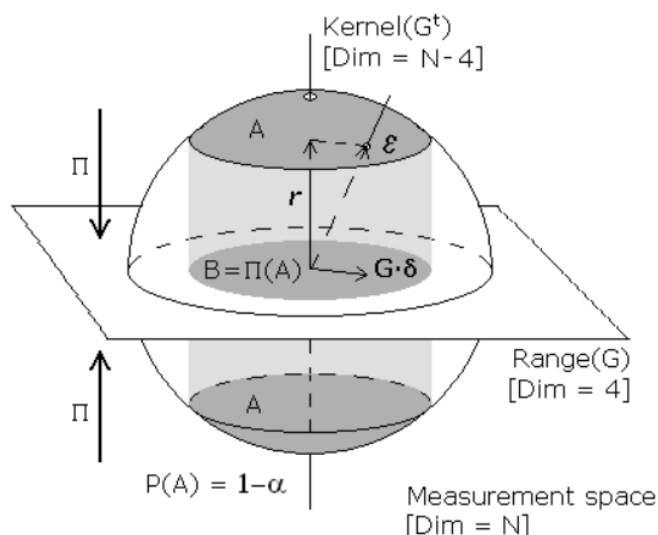


Figura 57: Ilustración del concepto IBPL.

La isotropía implica que el vector de error ϵ apunta según una distribución uniforme en un espacio de N dimensiones. El valor de k se puede definir como:

Ecuación 5.39

$$k \geq \frac{\|H \cdot \Delta x\|}{\|r\|}$$

Para cada valor de k se puede definir una región de la esfera (denominada A en la Figura 57) en la que se cumple la Ecuación 5.39. Si se escoge k para que el área de A sea una fracción $1 - \alpha$ del área total de la esfera, también se cumpliría la ecuación Ecuación 5.38.

Los algoritmos RAIM convencional y con separación de soluciones aseguran que el cálculo de la solución de navegación y del nivel de protección se lleva a cabo únicamente con medidas válidas. Este tipo de métodos se denominan MRA (*Measurement Rejection Approach*). En cambio, el RAIM IBPL realiza el cálculo del nivel de protección basándose en la caracterización de los errores en las

medidas. A este tipo de técnicas se las denomina ECA (*Error Characterization Approach*).

Las técnicas ECA suelen proporcionar niveles de protección más elevados que las MRA, por lo que no suelen cumplir los requisitos establecidos en las aplicaciones de aviación. Por otro lado, ofrecen también una mayor disponibilidad de solución, lo cual resulta sumamente útil en entornos urbanos.

5.3 Estimadores robustos

Los residuos obtenidos del método de los mínimos cuadrados no se comportan de forma correcta si la distribución de los errores no es normal. Los estimadores robustos presentan una buena aproximación aún si los errores no estuviesen distribuidos de forma normal. Además, la extrema sensibilidad de los métodos tradicionales a las medidas anómalas u *outliers* como los mínimos cuadrados causa que las soluciones de posición no cuenten con la precisión esperada.

Por otro lado, los métodos RAIM se desarrollaron basándose en la suposición de la existencia de una única medida errónea. Resulta necesario revisar esta suposición ya que en entornos urbanos aumenta la posibilidad de que exista más de una medida errónea. Además, con la mayor disponibilidad de satélites visibles que garantizan los sistemas multiconstelación, también se incrementa la probabilidad de la existencia de múltiples errores.

Un estimador robusto se caracteriza mediante el punto de quiebre y la función de influencia. El punto de quiebre mide la fiabilidad global de la estimación determinando la cantidad mínima de observaciones erróneas para que el estimador proporcione una estimación incoherente. La función de influencia representa cómo se comporta el estimador al variar la distribución de las observaciones.

Existen numerosos tipos de estimadores robustos. Uno de los más populares es el estimador-M. Este estimador es esencialmente una media ponderada, en la que la ponderación tiene objetivo limitar la influencia de las medidas anómalas en el estimador (Ruckstuhl, 2016).

Ecuación 5.40

$$\widehat{\beta}_M = \frac{\sum_{i=1}^n w_i y_i}{\sum_{i=1}^n w_i}$$

El método seguido para los estimadores-M se basa en minimizar una función de los residuos (Gaglione, Innac, Pastore, & Angrisano, RObust estimation methods applied to GPS in harsh environments, 2017).

Ecuación 5.41

$$\min \sum_{i=1}^n \theta(r_i) = \min \sum_{i=1}^n \theta(\Delta\rho_i - H_i\Delta x) = 0$$

Si definimos $\psi=\theta'$ y derivamos la expresión anterior se obtiene:

Ecuación 5.42

$$\sum_{i=1}^n \psi(\Delta\rho_i - H_i\Delta x)H_i = 0$$

Finalmente, definiendo $w(r) = \psi(r)/r$ se obtiene:

Ecuación 5.43

$$\sum_{i=1}^n w_i(\Delta\rho_i - H_i\Delta x)H_i = 0$$

Los pesos w_i se definen mediante una función $w_i(r_i)$ que depende de los residuos. Los distintos tipos de estimadores-M dependen de la función que se escoja.

El procedimiento a seguir para la obtención del estimador robusto es el siguiente:

- 1) Se obtiene una estimación inicial de Δx mediante el método de mínimos cuadrados.
- 2) Se calculan los residuos y sus correspondientes pesos.
- 3) Se comprueba la convergencia de la Ecuación 5.43. En caso negativo, vuelve a calcular Δx utilizando como matriz de pesos la compuesta por los pesos del paso 2).
- 4) Se repiten los pasos 2) y 3) hasta alcanzar la convergencia de la ecuación Ecuación 5.43.

También existen otros estimadores robustos como los estimadores-L, que se basan en combinaciones lineales de estadísticos de orden, o los estimadores-R, que establecen rangos a los residuos para computar ciertas ponderaciones. Para este trabajo únicamente se van a analizar los estimadores-M, ya que son los más populares y cuentan con numerosas variaciones para mejorar su desempeño.

5.3.1 Estimador-M de Huber

El método de estimadores-M propuesto por Huber en 1964 utiliza una función para calcular los pesos en función de los residuos que tiene la siguiente forma:

Ecuación 5.44

$$(w_i)_{j+1} = \begin{cases} 1, & \left| \frac{(r_i)_j}{\sigma_0} \right| \leq 1,345 \\ \frac{1,345}{\left| \frac{(r_i)_j}{\sigma_0} \right|}, & \left| \frac{(r_i)_j}{\sigma_0} \right| > 1,345 \end{cases}$$

Donde σ_0 es la desviación estándar de los residuos que se define como (Knight & Wang, 2009):

Ecuación 5.45

$$\sigma_0 = \frac{1}{0,6745} \cdot \text{Mediana}(|r_i - \text{Mediana}(r_i)|)$$

La función ψ de Huber se puede representar gráficamente de la siguiente forma:

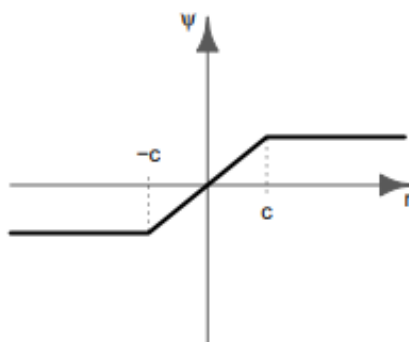


Figura 58: Función de Huber.

Definiendo $w(r) = \psi(r)/r$, la obtención de los pesos se puede representar gráficamente como:

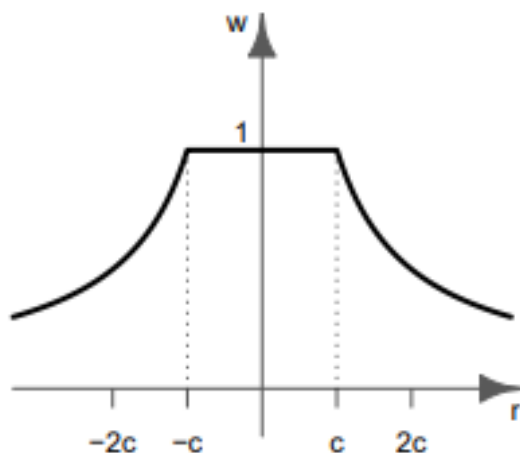


Figura 59: Pesos de la función de Huber.

El valor de c , comúnmente llamada constante de afinamiento de la función de Huber, se define con un valor de 1,35.

Si en el paso 3) no se ha logrado la convergencia de la Ecuación 5.43, la matriz de peso utilizada en el método de los mínimos cuadrados será:

Ecuación 5.46

$$W = \text{diag}([w_1 \ w_2 \ \dots \ w_n])$$

Ecuación 5.47

$$(w_i)_{j+1} = \begin{cases} (w_i)_j, & \left| \frac{(r_i)_j}{\widehat{\sigma}_0} \right| \leq 1,345 \\ (w_i)_j \cdot \frac{1,345}{\left| \frac{(r_i)_j}{\widehat{\sigma}_0} \right|}, & \left| \frac{(r_i)_j}{\widehat{\sigma}_0} \right| > 1,345 \end{cases}$$

Donde j hace referencia al número de iteración del procedimiento explicado anteriormente.

5.3.2 Estimador-M IGGII

Además del estimador-M de Huber, en este trabajo se va a analizar el desempeño del estimador-M IGGIII. En este método el cálculo de los pesos se lleva a cabo mediante la comparación de los residuos estandarizados con un umbral (Yang & Xu, 2016).

Ecuación 5.48

$$(w_i)_{j+1} = \begin{cases} (w_i)_j, & |\tilde{r}_i| \leq 1,345 \\ (w_i)_j \left(\frac{1.345}{|\tilde{r}_i|} \right) \left(\frac{3 - |\tilde{r}_i|}{3 - 1,345} \right)^2, & 1,345 < |\tilde{r}_i| \leq 3 \\ 0, & |\tilde{r}_i| > 3 \end{cases}$$

Los residuos estandarizados se calculan como:

Ecuación 5.49

$$\tilde{r} = \frac{r}{\sigma_0}$$

Donde σ_0 es la desviación estándar de los residuos definida anteriormente.

En este caso, si el valor del residuo estandarizado es mayor que el valor del umbral, se declara tal medida como errónea y se excluye asignándole un peso de valor nulo.

5.4 Simulación y resultados

Para valorar las distintas técnicas de detección de medidas erróneas que se han explicado en las secciones anteriores, se va a utilizar el mismo conjunto de datos que en el capítulo anterior.

A diferenciar de las simulaciones del capítulo anterior, se van a introducir errores en las medidas de los satélites para poder comprobar que las técnicas detectan y excluyen dichas medidas erróneas.

Se van a poner a prueba los métodos de detección en situaciones con uno, dos y tres satélites con medidas erróneas. Además, se van a introducir errores en el pseudorango de dichos satélites de dos formas:

- Valor constante de 10 m.
- Valor creciente desde 10 m hasta 30 m.

De la misma forma que en el capítulo de selección de satélites, se va a ensayar con el receptor estático, siguiendo una trayectoria en 2D y en 3D. Para cada situación se van a comparar los errores de posición que presentan las soluciones de las distintas técnicas después de llevar a cabo la detección y exclusión de medidas erróneas.

Además, en los casos en los que sea posible, se evaluarán los niveles de protección que proporcionan los algoritmos. Se desarrollarán los diagramas de Stanford pertinentes para comparar los niveles de protección con los errores de posición.

5.4.1 Caso estático

5.4.1.1 *Un satélite con error constante en el pseudorango*

Para el caso estático se va a contar con un receptor de posición fija con coordenadas idénticas a las de selección de satélites. La Figura 60 contiene las funciones de densidad de probabilidad para todas las técnicas tratadas en este trabajo a excepción del IBPL, ya que este método no lleva a cabo detección y exclusión de medidas.

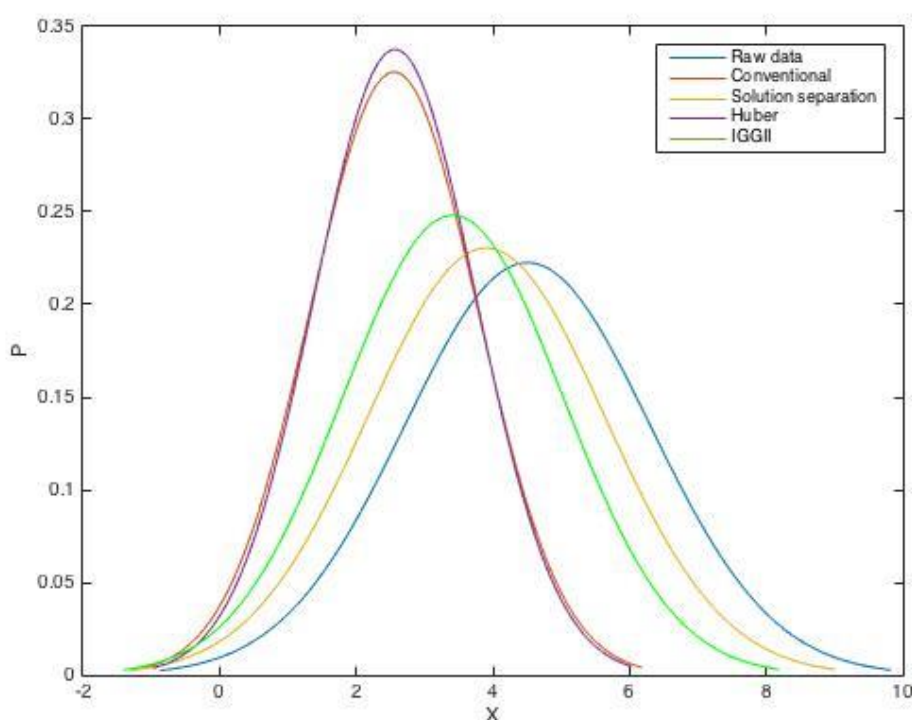


Figura 60: Funciones de densidad de probabilidad del error de posición para el caso estático con un satélite con error constante en el pseudorango.

Como se puede observar, los errores de posición del RAIM convencional y del estimador-M de Huber siguen una distribución muy similar. El estimador-M IGGII proporciona una distribución del error de posición con una media y desviación típica superiores a estos. Por último, el método de RAIM basado en la separación de soluciones presenta la distribución del error menos propicia del conjunto de técnicas evaluadas. La distribución del error si no se aplica ninguna técnica de detección y exclusión cuenta con mayor media y desviación típica, este sería el caso para el método IBPL.

A continuación se presentan las tasas de exclusión de las medidas erróneas introducidas para el RAIM convencional y el basado en separación de soluciones.

Técnica	Medidas erróneas excluidas (%)
<i>RAIM Convencional</i>	76.89
<i>RAIM basado en separación de soluciones</i>	19.2

Tabla 19: Tasas de exclusión de medidas erróneas para el caso estático con un error constante.

El RAIM convencional consigue excluir una mayor proporción de las medidas erróneas introducidas en el pseudorango que el basado en separación de soluciones. Este hecho influye directamente en los errores de posición mostrados en la Figura 60.

5.4.1.1.1 RAIM convencional

A continuación se presentan los niveles de protección (horizontal y vertical) que proporciona el RAIM convencional a lo largo del tiempo de simulación.

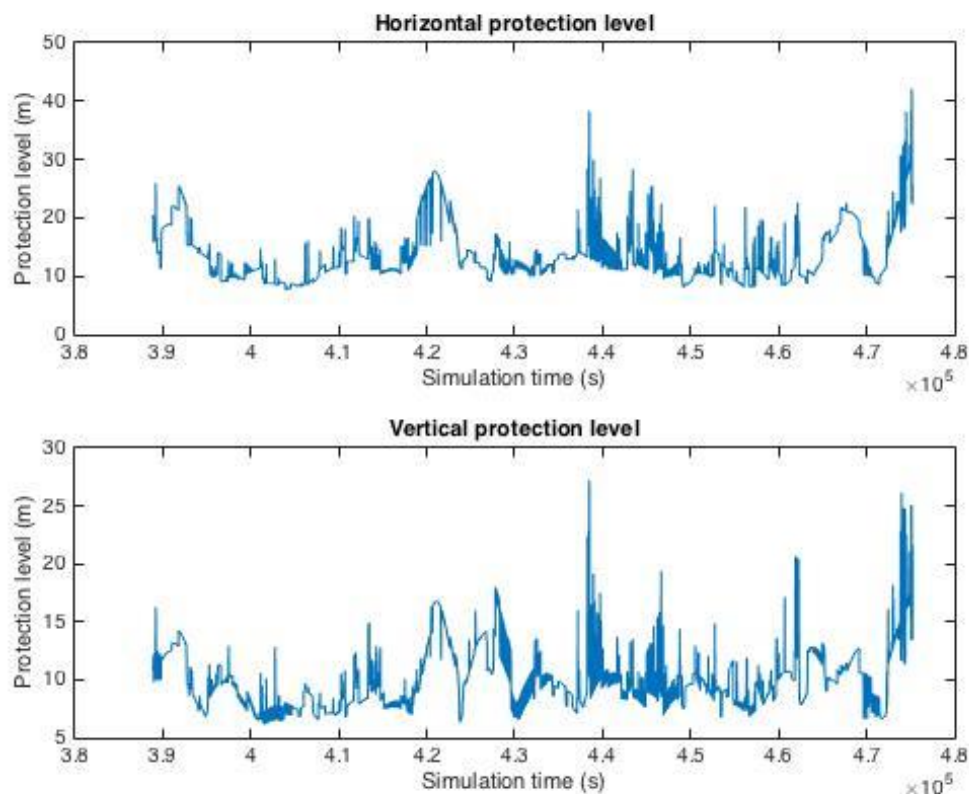


Figura 61: Niveles de protección del RAIM convencional para el caso estático con un satélite con error constante en el pseudorange.

Se puede apreciar como el nivel de protección salta cuando varía el número de satélites disponibles. Por ejemplo, se pueden observar como los valores máximos de ambos niveles de protección se dan cuando el número de satélites visibles alcanza su valor mínimo (17).

Los diagramas de Stanford para este caso se muestran en la Figura 62 y la Figura 63. Es puede apreciar como la nube de puntos se encuentra en la parte superior a la diagonal que atraviesa el diagrama (zona de operación nominal).

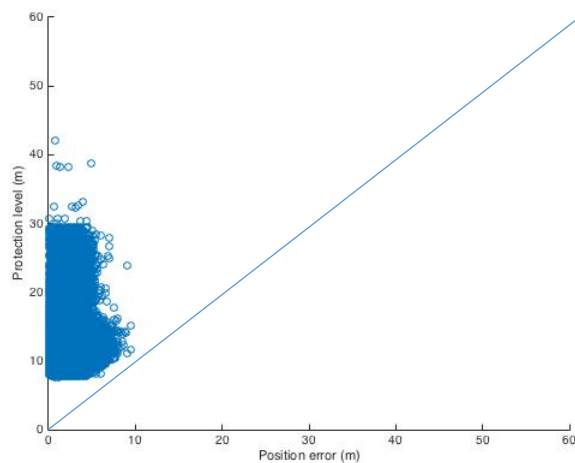


Figura 62: Diagrama de Stanford del componente horizontal del RAIM convencional para el caso estático con un satélite con error constante en el pseudorango.

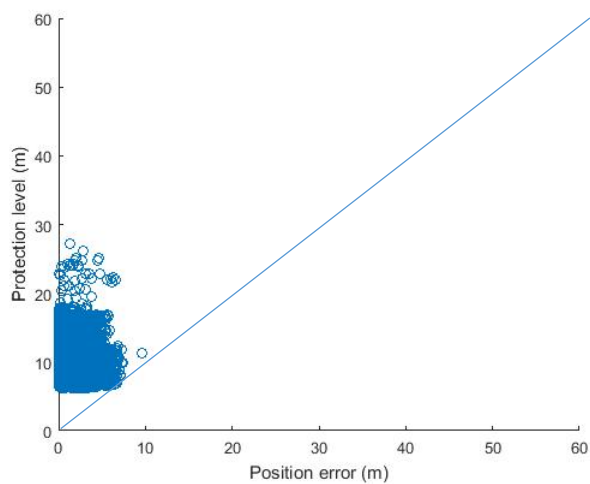


Figura 63: Diagrama de Stanford del componente vertical del RAIM convencional para el caso estático con un satélite con error constante en el pseudorango.

5.4.1.1.2 RAIM basado en la separación de soluciones

A continuación se presentan los niveles de protección (horizontal y vertical) que proporciona el RAIM basado en separación de soluciones a lo largo del tiempo de simulación.

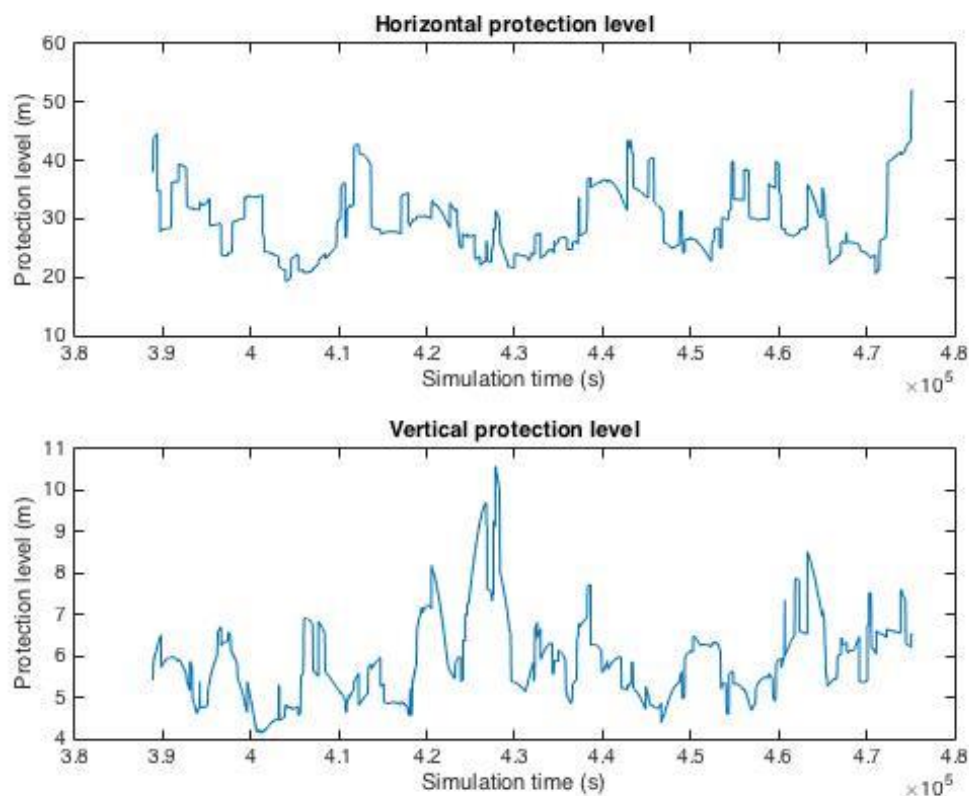


Figura 64: Niveles de protección del RAIM basado en la separación de soluciones para el caso estático con un satélite con error constante en el pseudorange.

En general, los niveles de protección que se obtiene utilizando esta variante del RAIM convencional tienen valores superiores a los obtenidos con el RAIM convencional.

Los diagramas de Stanford (Figura 65 y Figura 66) muestran como mediante la separación de soluciones se consigue trabajar en la zona de operaciones nominales en el caso del componente horizontal, mientras que para el vertical se trabajó bajo la diagonal en ciertos instantes de la simulación.

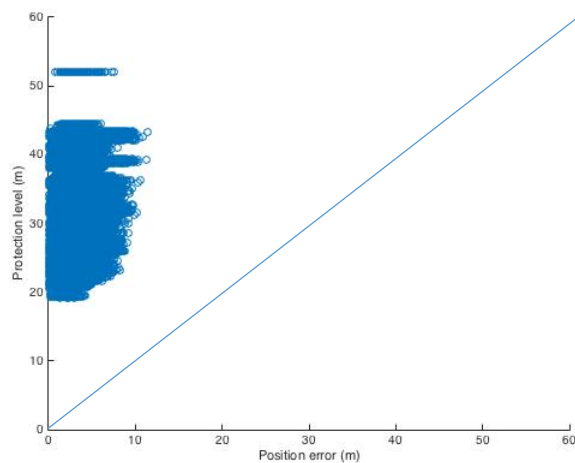


Figura 65: Diagrama de Stanford del componente horizontal del RAIM basado en separación de soluciones para el caso estático con un satélite con error constante en el pseudorango.

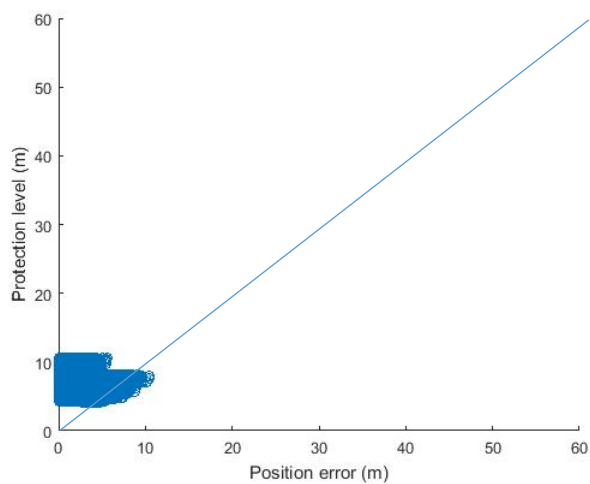


Figura 66: Diagrama de Stanford del componente vertical del RAIM basado en separación de soluciones para el caso estático con un satélite con error constante en el pseudorango.

5.4.1.1.3 IBPL

Como se ha mencionado anteriormente, la técnica del nivel de protección basado en la isotropía no lleva a cabo una detección y exclusión de medidas erróneas.

Se puede observar en la figura siguiente como en los niveles de protección que se obtienen mediante el uso de este método predominan valores superiores a los de los métodos RAIM de las secciones anteriores.

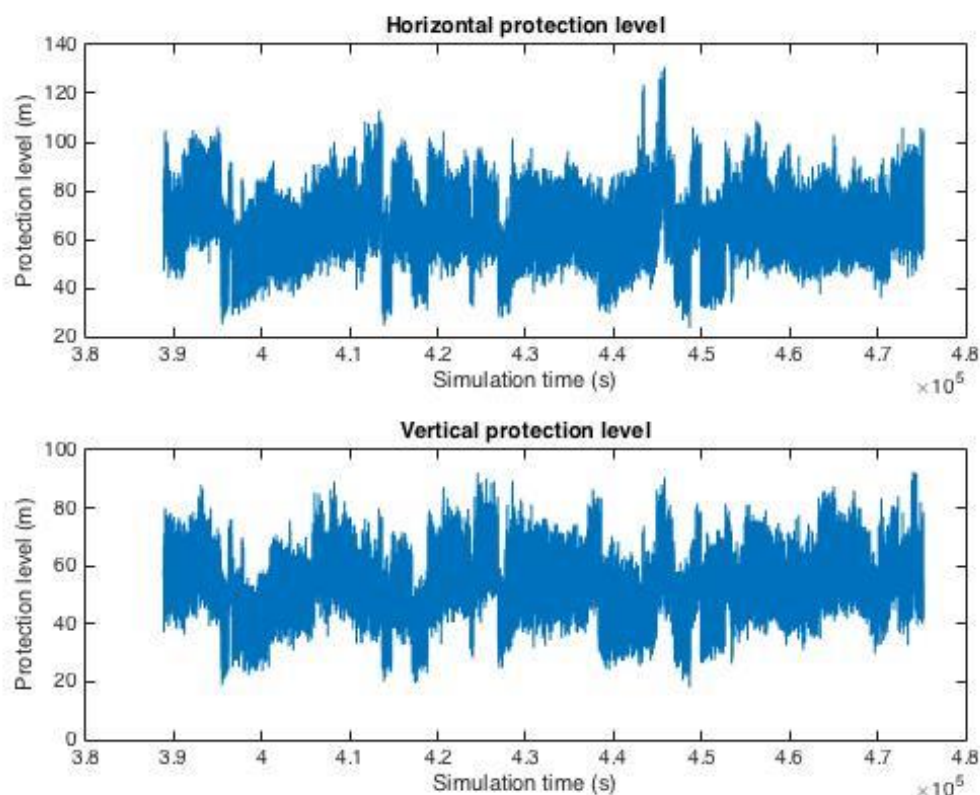


Figura 67: Niveles de protección del IBPL para el caso estático con un satélite con error constante en el pseudorange.

Los diagramas de Stanford de la Figura 68 y la Figura 69 muestran como mediante el método IBPL se asegura la operación en la zona deseada con un margen considerable.

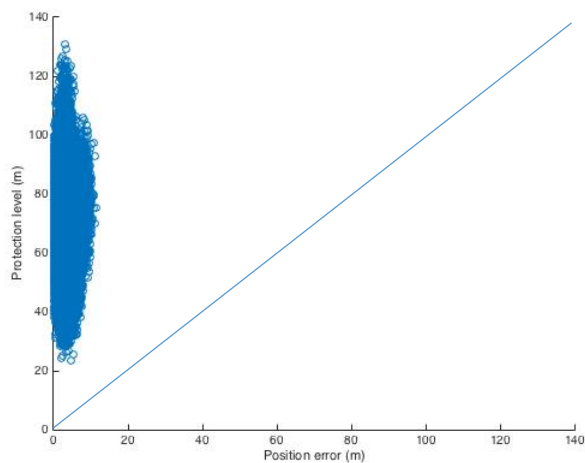


Figura 68: Diagrama de Stanford del componente horizontal del IBPL para el caso estático con un satélite con error constante en el pseudorango.

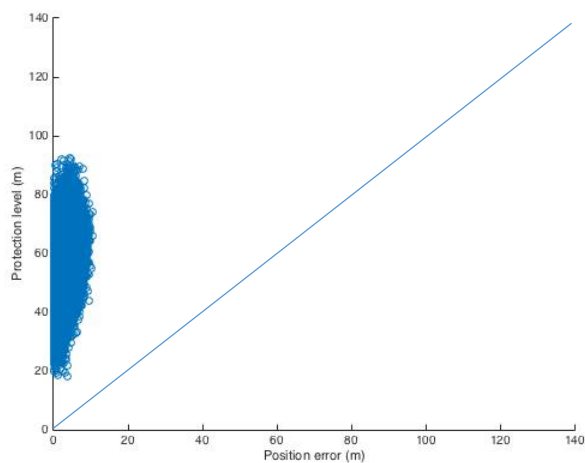


Figura 69: Diagrama de Stanford del componente vertical del IBPL para el caso estático con un satélite con error constante en el pseudorango.

5.4.1.2 *Dos satélites con errores constantes en el pseudorange*

Para poner a prueba las técnicas en una situación de condiciones más adversas, se introducen medidas erróneas de pseudorange en dos de los satélites disponibles. La muestra las funciones de densidad de probabilidad del error de posición obtenidas tras esta prueba.

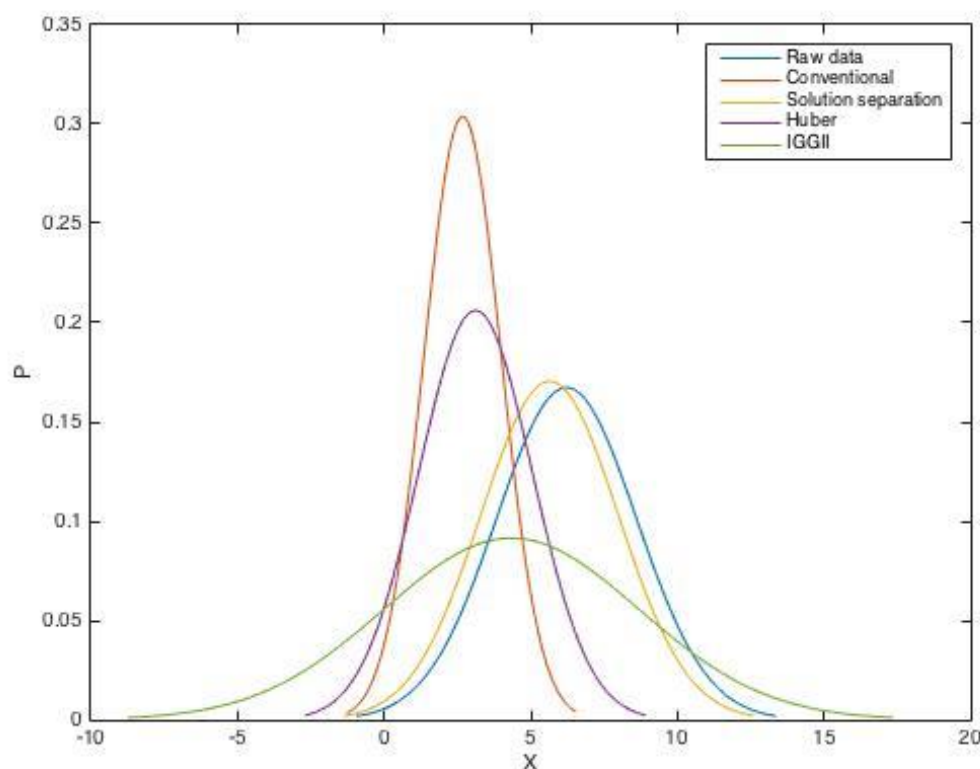


Figura 70: Función de densidad de probabilidad del error de posición para el caso estático con dos satélites con errores constantes en el pseudorange.

El RAIM convencional continúa ofreciendo una distribución del error muy similar al caso anterior con un único satélite erróneo. En cambio, el estimador-M de Huber proporciona una distribución del error de posición con mayor media y desviación típica que en el caso anterior.

El RAIM basado en la separación de posiciones no consigue mejorar apenas la distribución del error, siendo prácticamente idéntica a la del error sin la aplicación de ninguna técnica.

Por último, el estimador-M IGGII ofrece una distribución del error con una media ligeramente más favorable que en el caso del RAIM con separación de soluciones. Sin embargo, la desviación estándar del error es mayor que la obtenida si no se hubiese aplicado el estimador.

5.4.1.2.1 RAIM convencional

En la siguiente figura se pueden observar los niveles de protección horizontal y vertical para el RAIM convencional. Es posible apreciar que estos niveles son prácticamente idénticos a los del caso con un único satélite con el pseudorange erróneo.

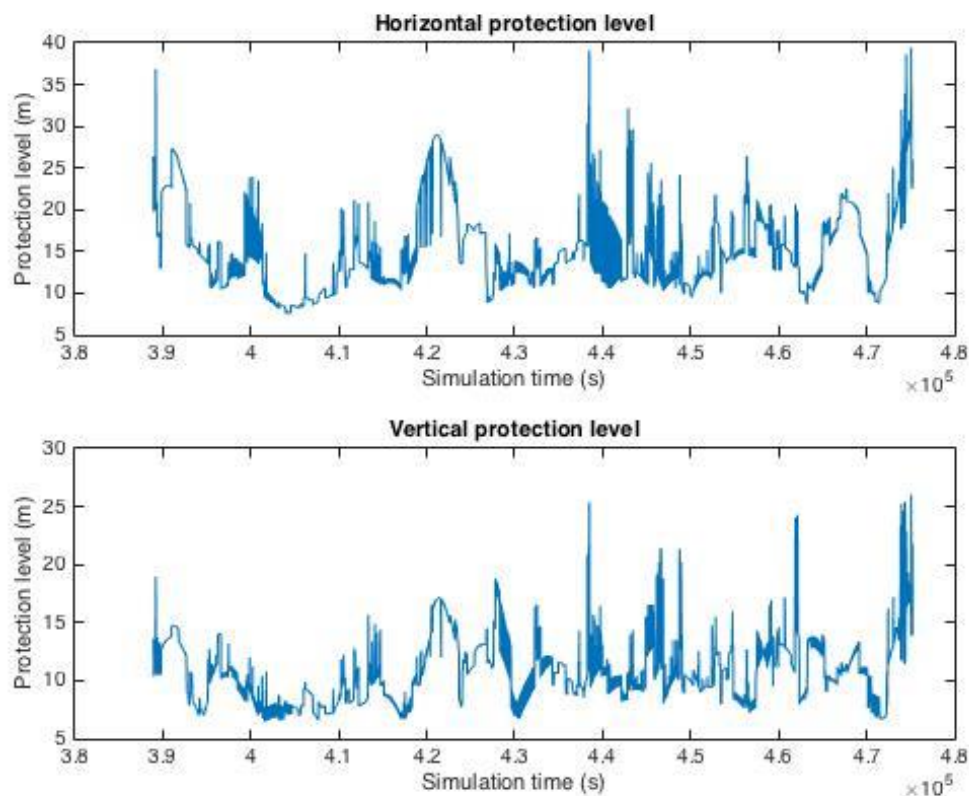


Figura 71: Niveles de protección para el RAIM convencional para el caso estático con dos satélites con errores constantes en el pseudorange.

Debido a que los niveles de protección se han mantenido iguales a pesar de introducir un nuevo satélite erróneo, y que el error de posición ha aumentado, en los diagramas de Stanford (Figura 72 y Figura 73) muestran como la nube de puntos se encuentra más próxima a la diagonal que marca la zona de operación nominal.

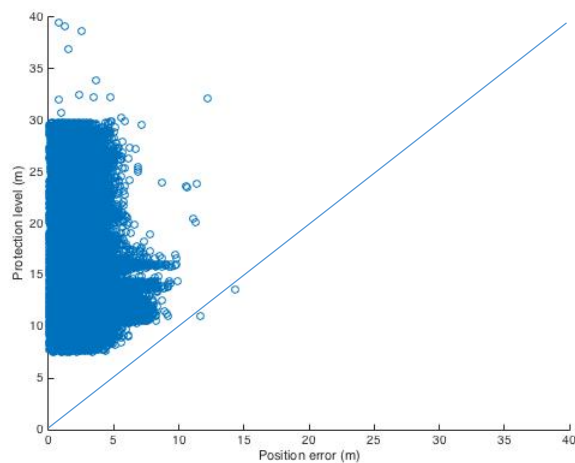


Figura 72: Diagrama de Stanford del componente horizontal para el RAIM convencional para el caso estático con dos satélites con errores constantes en el pseudorange.

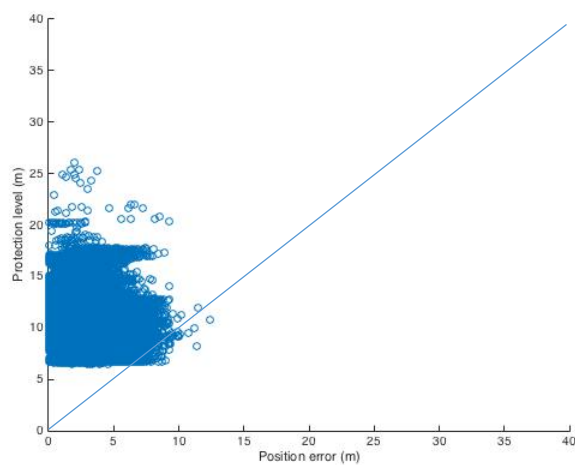


Figura 73: Diagrama de Stanford del componente vertical para el RAIM convencional para el caso estático con dos satélites con errores constantes en el pseudorange.

5.4.1.2.2 RAIM basado en la separación de soluciones

Los niveles de protección que proporciona el RAIM basado en la separación de soluciones al introducir un nuevo error en el pseudorange de otro satélite se muestran en la figura siguiente.

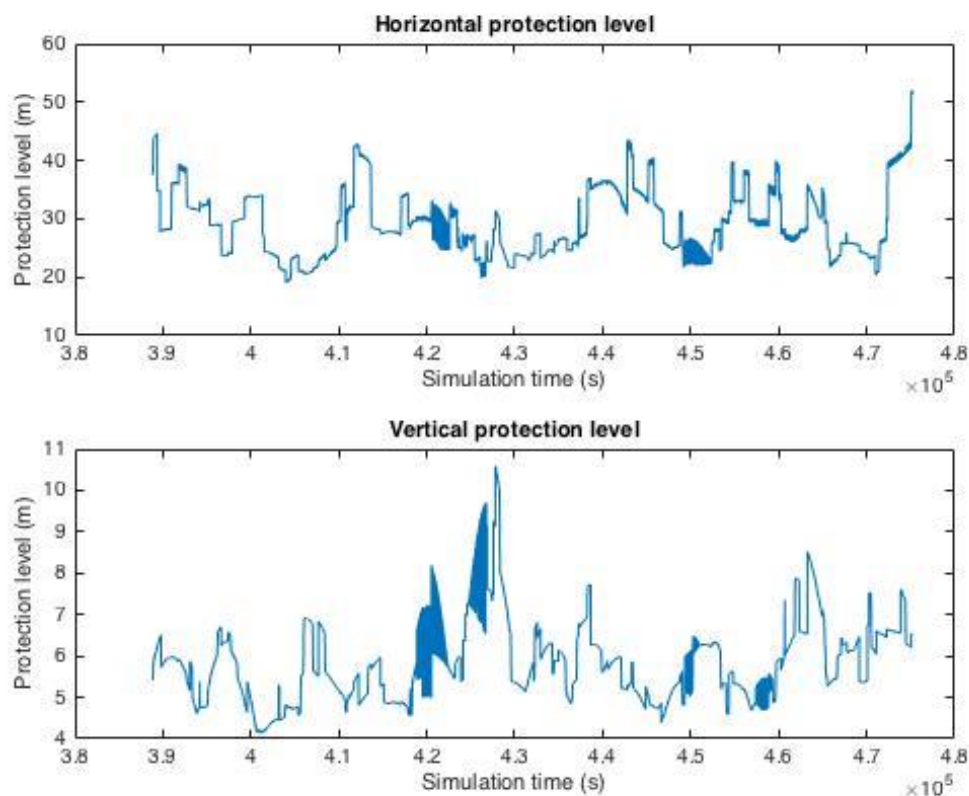


Figura 74: Niveles de protección para el RAIM basado en separación de soluciones para el caso estático con dos satélites con errores constantes en el pseudorange.

Se puede observar como en determinados momentos los niveles de protección se reducen con respecto al caso con un único satélite con medida errónea. Esto se debe a que el algoritmo no consigue detectar las medidas erróneas en todos los casos y por tanto no excluye dichos satélites. Al contar con un mayor número de satélites, los niveles de protección se reducen.

Los diagramas de Stanford muestran como los niveles de protección son muy similares a los del caso anterior. Sin embargo, se puede observar como el error de posición ha aumentado. Debido a esto, para el componente vertical, una proporción considerable de la nube de puntos se encuentra en la zona de operación engañosa.

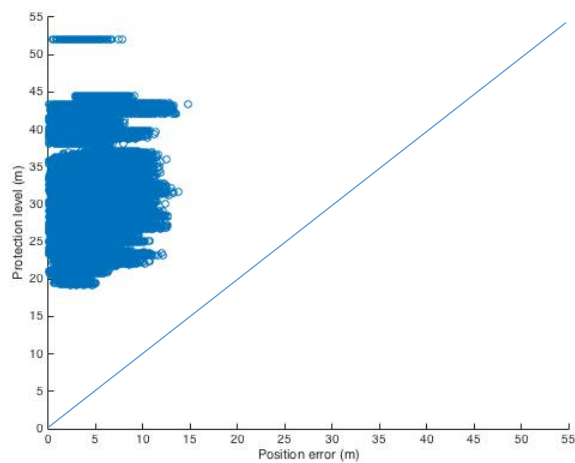


Figura 75: Diagrama de Stanford del componente horizontal para el RAIM basado en separación de soluciones para el caso estático con dos satélites con errores constantes en el pseudorange.

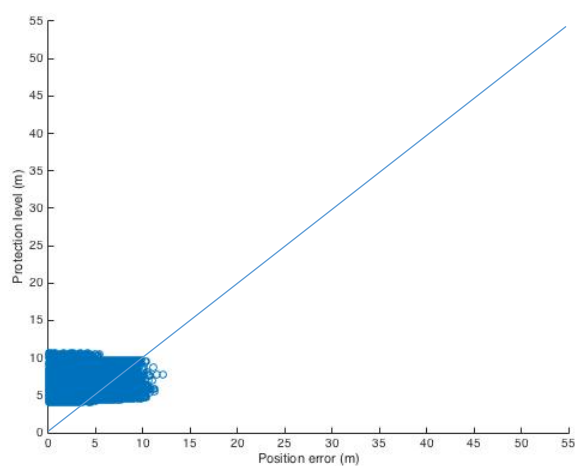


Figura 76: Diagrama de Stanford del componente vertical para el RAIM basado en separación de soluciones para el caso estático con dos satélites con errores constantes en el pseudorange.

5.4.1.2.3 IBPL

Los niveles de protección proporcionados por el IBPL vuelven a ser considerablemente más elevados que para las técnicas RAIM, como se puede observar en la siguiente figura.

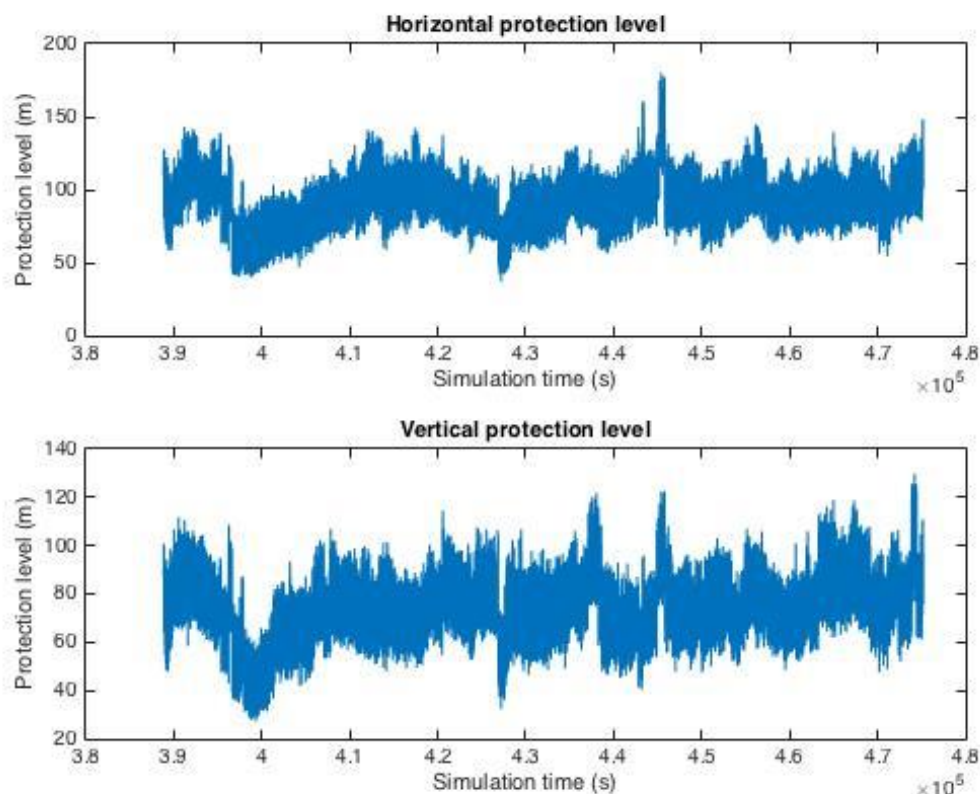


Figura 77: Niveles de protección para el IBPL para el caso estático con dos satélites con errores constantes en el pseudorange.

Evidentemente, los niveles de protección han sufrido un incremento respecto al caso de un solo satélite con medidas erróneas.

En los diagramas de Stanford se puede observar que a pesar de que el error de posición es mayor, los niveles de protección del IBPL aseguran la operación en la zona nominal del diagrama.

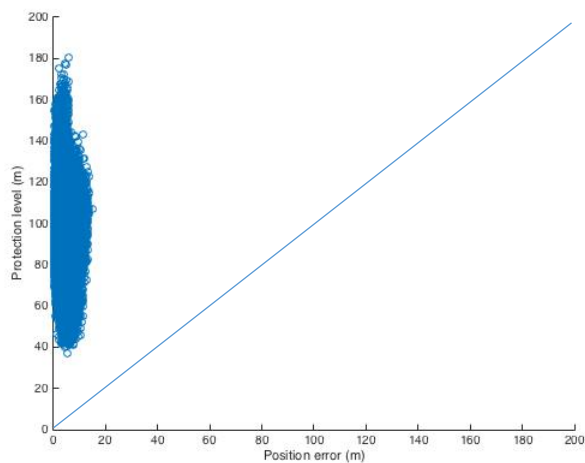


Figura 78: Diagrama de Stanford del componente horizontal para el IBPL para el caso estático con dos satélites con errores constantes en el pseudorange.

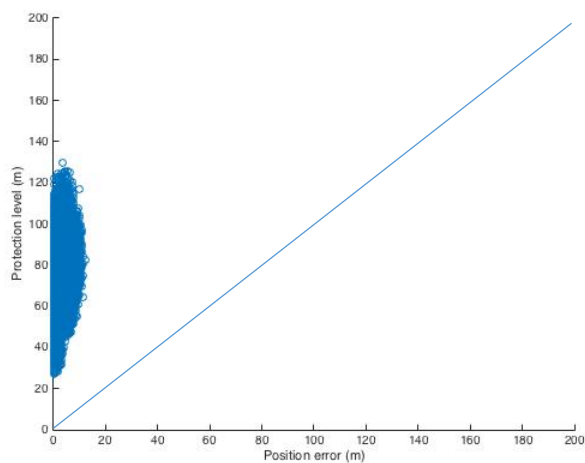


Figura 79: Diagrama de Stanford del componente vertical para el IBPL para el caso estático con dos satélites con errores constantes en el pseudorange.

5.4.1.3 *Tres satélites con errores constantes en el pseudorango*

Si se incrementa a tres el número de satélites con medidas erróneas, la distribución de los errores de posición se ve modificada con respecto a los casos anteriores.

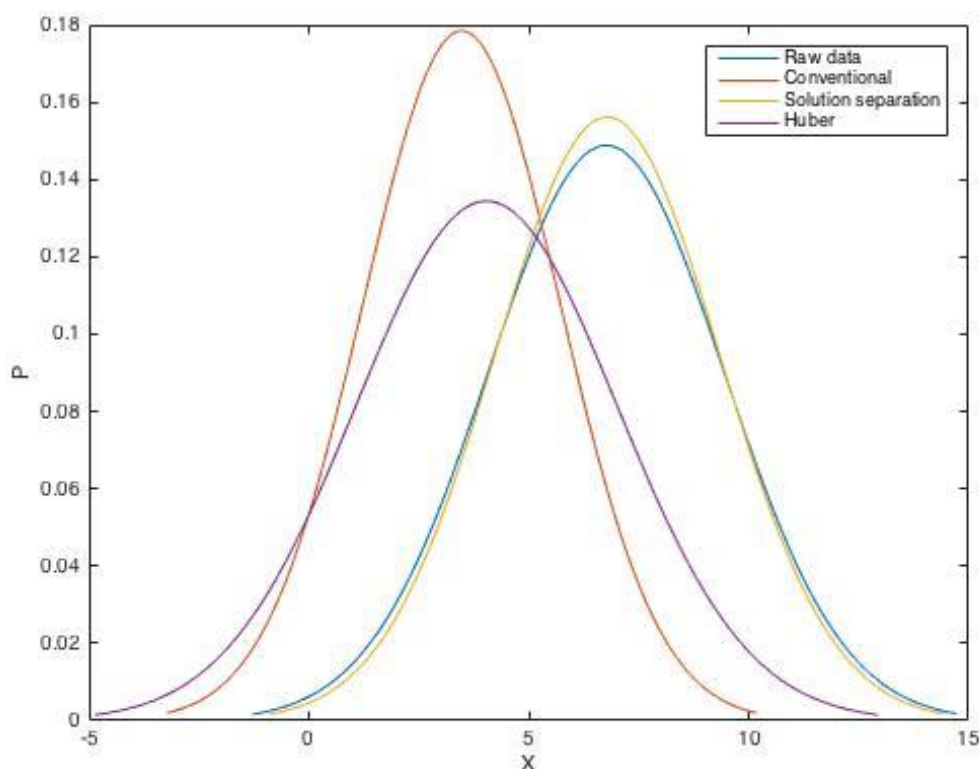


Figura 80: Función de densidad de probabilidad del error de posición para el caso estático con tres satélites con errores constantes en el pseudorango.

Lógicamente, si no se aplica ninguna técnica de detección y exclusión de medidas erróneas la distribución del error contará con un valor medio mayor que en los casos anteriores. Se puede observar en la Figura 80 como la distribución para el RAIM basado en separación de soluciones es prácticamente idéntica a esta.

La técnica del RAIM convencional sigue proporcionando los mejores resultados según la distribución del error, que cuenta con menor media y desviación típica que el resto de métodos.

Al igual que en el caso con dos satélites erróneos, el estimador-M IGGII es el método que proporciona la distribución de error que peor comportamiento tiene. La desviación típica lograda por esta técnica es mayor que la del caso con dos satélites erróneos. La distribución se muestra en la figura siguiente.

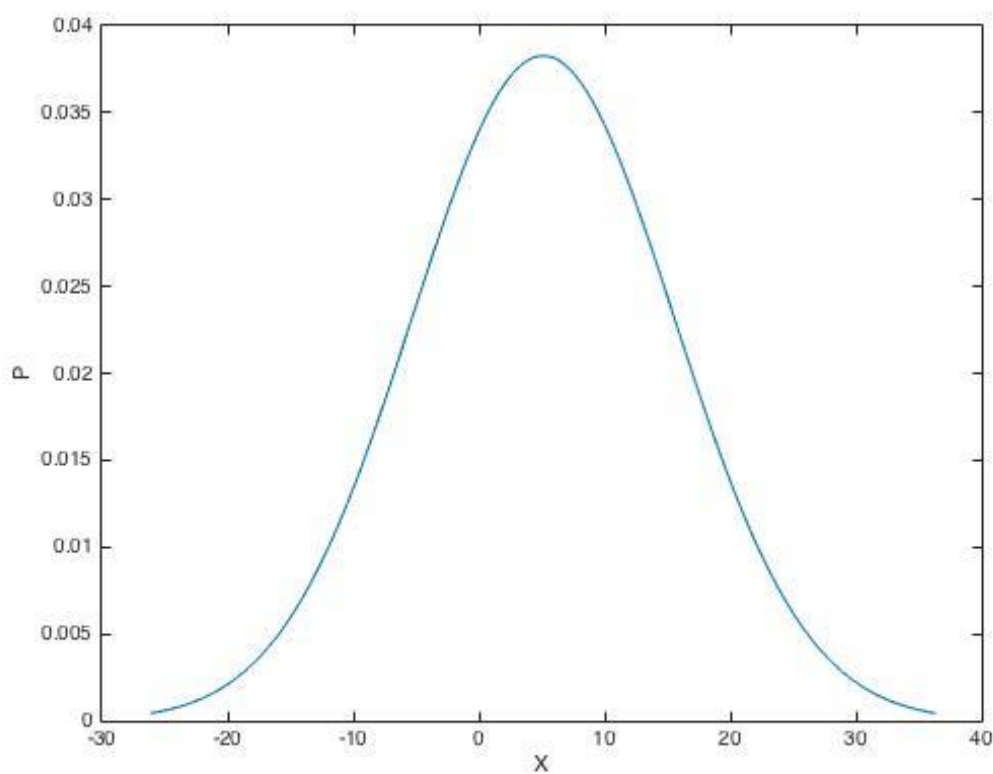


Figura 81: Función de densidad de probabilidad del error de posición para el estimador-M IGGII para el caso estático con tres satélites con errores constantes en el pseudorange.

5.4.1.3.1 RAIM convencional

Los niveles de protección del RAIM convencional sufren un incremento al añadir un tercer satélite con el pseudorango erróneo. Esos incrementos son especialmente apreciables en los momentos en los cuales el sistema cuenta con menor número de satélites visibles.

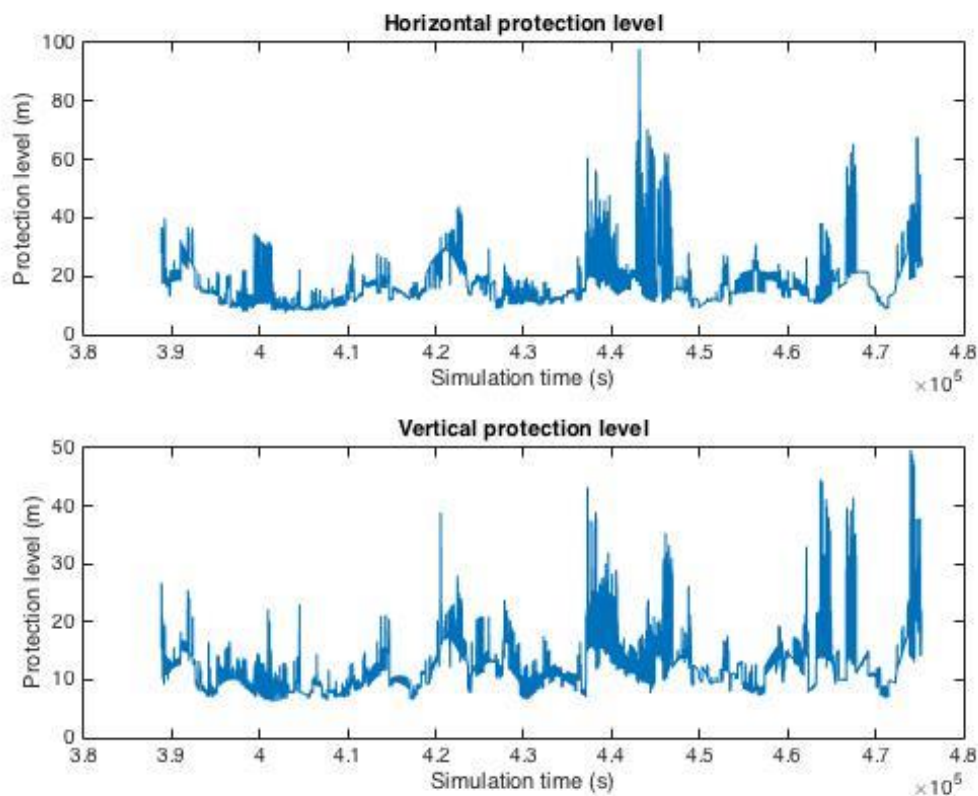


Figura 82: Niveles de protección para el RAIM convencional para el caso estático con tres satélites con errores constantes en el pseudorango.

En los diagramas de Stanford se puede apreciar como esta técnica consigue mantener durante la mayor parte del tiempo operar en la zona nominal. El aumento en el error de posición se compensa con incrementos de los niveles de posición.

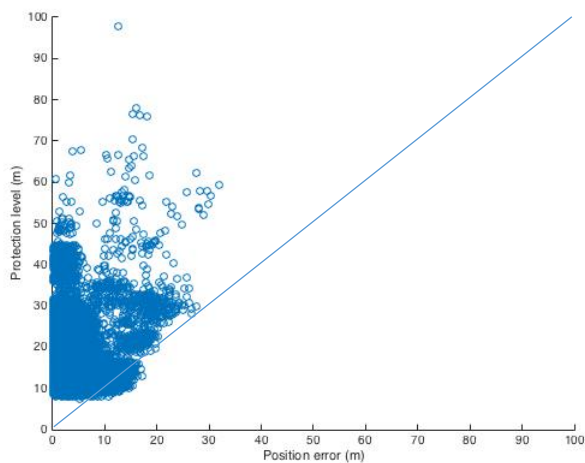


Figura 84: Diagrama de Stanford del componente horizontal para el RAIM convencional para el caso estático con tres satélites con errores constantes en el pseudorango.

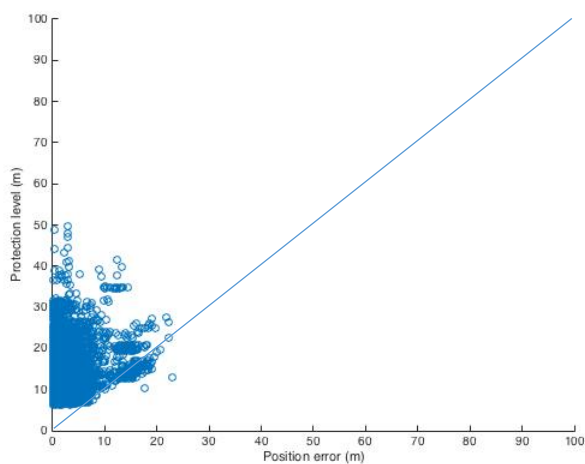


Figura 83: Diagrama de Stanford del componente vertical para el RAIM convencional para el caso estático con tres satélites con errores constantes en el pseudorango.

5.4.1.3.2 RAIM basando en separación de soluciones

Igual que en el caso con dos satélites con dos medidas erróneas, esta técnica no consigue detectar, y por tanto excluir, las medidas erróneas. Esta situación se da con mayor frecuencia que en el caso anterior, por lo que en este caso se calculan los niveles de protección con un mayor número de satélites. En la siguiente figura se pueden observar como los niveles de protección sufren reducciones en instantes donde en los casos anteriores no tenían lugar.

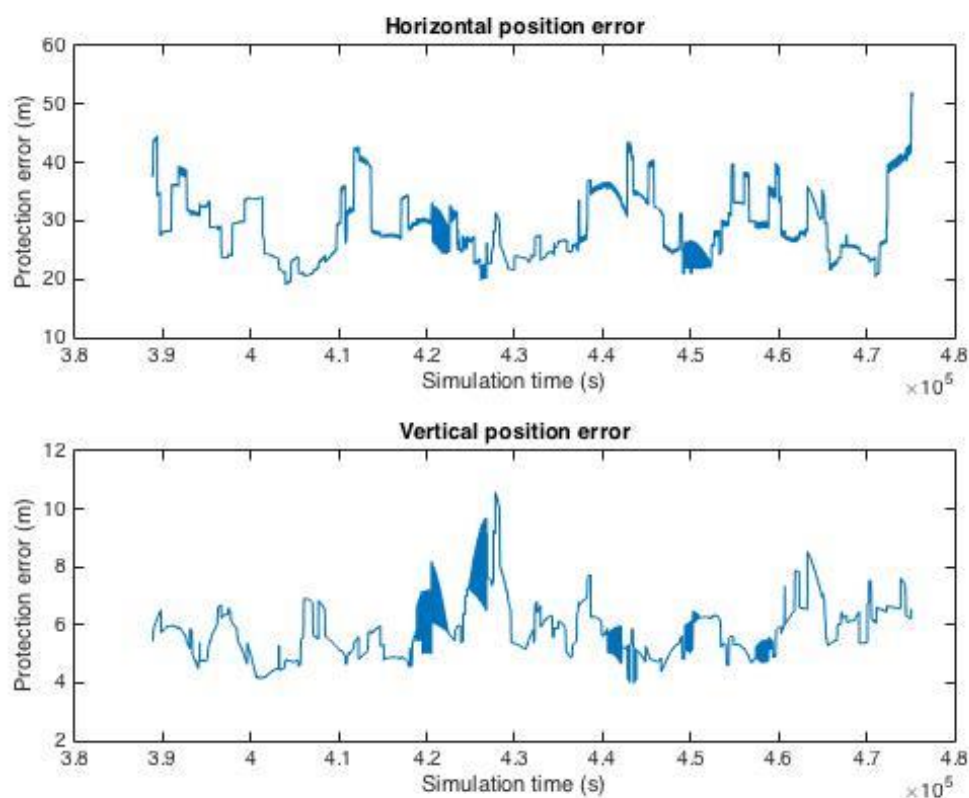


Figura 85: Niveles de protección para el RAIM basado en separación de soluciones para el caso estático con tres satélites con errores constantes en el pseudorange.

Las reducciones puntuales del valor del nivel de protección son especialmente apreciables en el componente vertical. El diagrama de Stanford muestra como un gran número de puntos se encuentran la zona de operación engañosa. En cuanto al componente vertical, aunque se puede apreciar como el error de posición ha sufrido un incremento, este no es suficiente como para evitar que se encuentre en la zona de operaciones nominales.

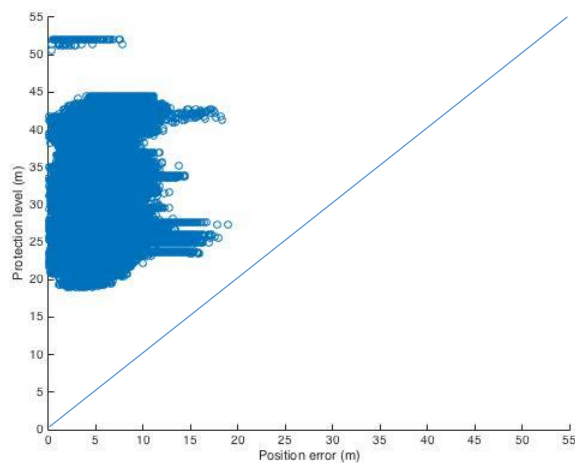


Figura 86: Diagrama de Stanford del componente horizontal para el RAIM basado en separación de soluciones para el caso estático con tres satélites con errores constantes en el pseudorange.

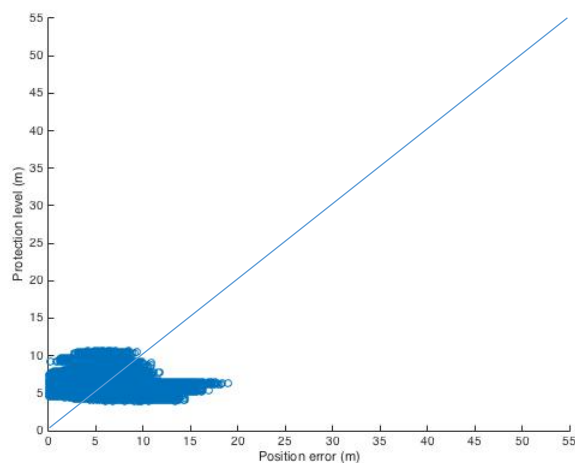


Figura 87: Diagrama de Stanford del componente vertical para el RAIM basado en separación de soluciones para el caso estático con tres satélites con errores constantes en el pseudorange.

5.4.1.3.3 IBPL

Los niveles de protección del IBPL vuelven a sufrir un incremento al introducir otro error en el pseudorango de un satélite. Los valores de estos niveles de protección son significativamente superiores a los proporcionados por otros métodos.

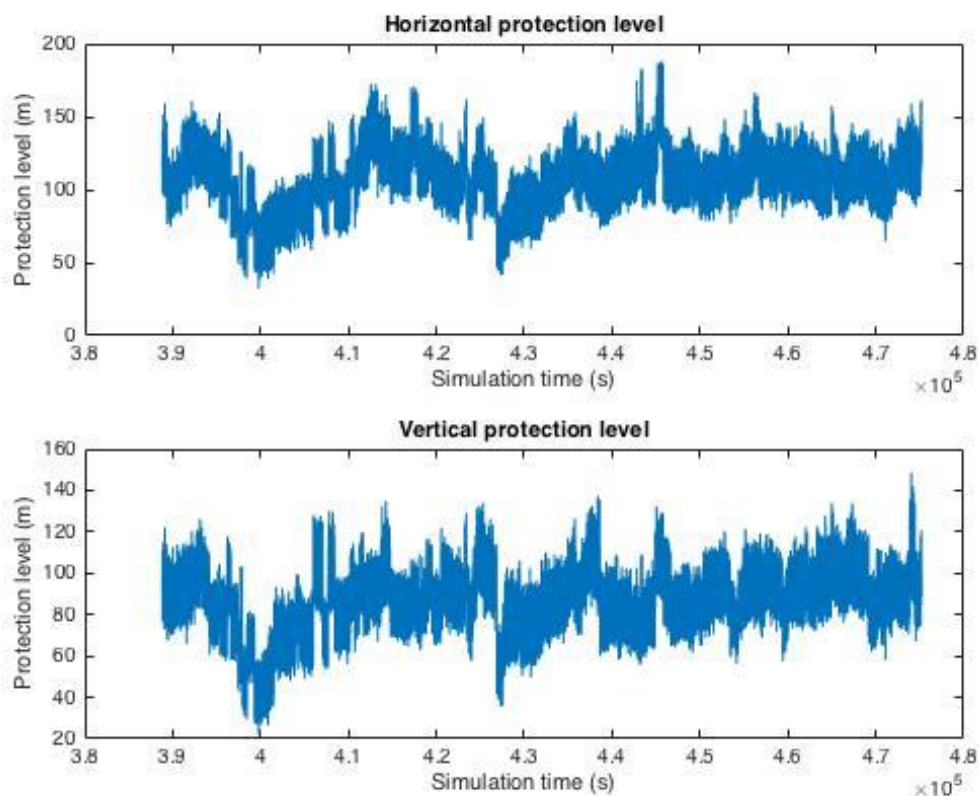


Figura 88: Niveles de protección para el IBPL para el caso estático con tres satélites con errores constantes en el pseudorango.

A pesar del aumento en los componentes horizontal y vertical del error de posición, los elevados niveles de protección aseguran la zona de operaciones nominales.

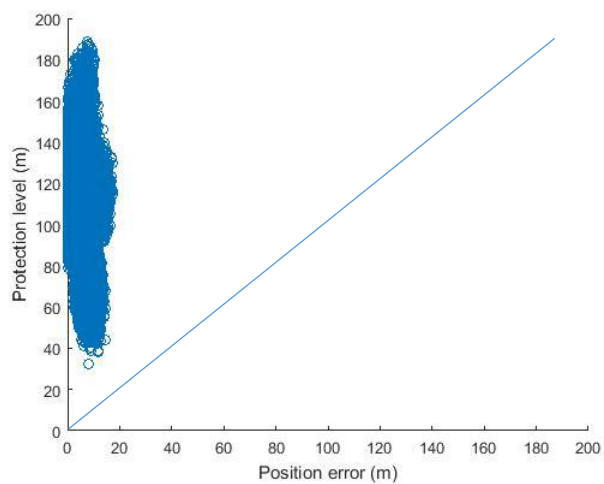


Figura 89: Diagrama de Stanford del componente horizontal para el IBPL para el caso estático con tres satélites con errores constantes en el pseudorango.

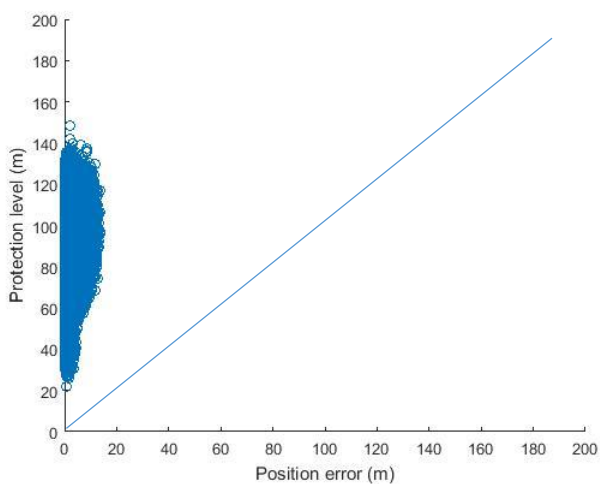


Figura 90: Diagrama de Stanford del componente vertical para el IBPL para el caso estático con tres satélites con errores constantes en el pseudorango.

5.4.1.4 Un satélite con error creciente en el pseudorango

Además de comprobar el desempeño de los métodos en situaciones con satélites que contienen errores constantes en el pseudorango, se va a introducir un error en el pseudorango cuyo módulo vaya incrementando de forma constante.

En este caso no va a resultar necesario llevar a cabo la simulación de 24 horas de longitud. Se ha escogido un periodo menor que dos horas en el que es posible observar la influencia del error del pseudorango en un satélite ($3.888\text{--}3.927 [10^5 \text{ s}]$) y luego en otro satélite distinto ($3.927\text{--}3.938 [10^5 \text{ s}]$).

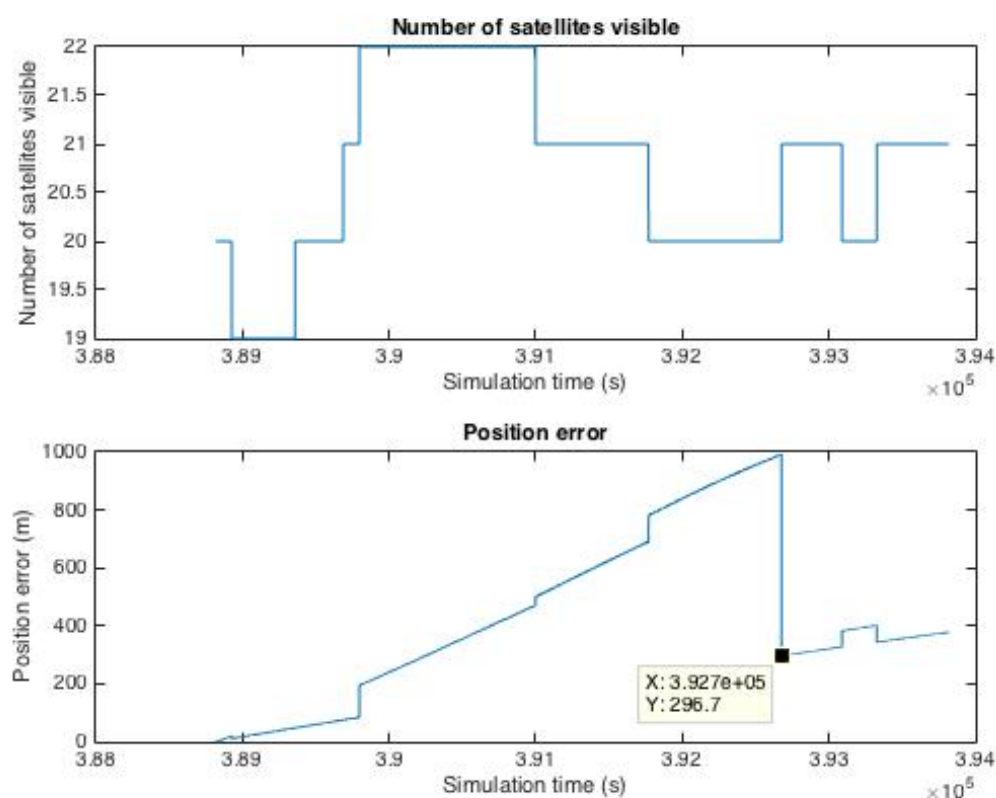


Figura 91: Número de satélites visibles y error de posición para el caso estático con error creciente.

Las funciones de densidad de probabilidad del error de posición muestran como el RAIM basado en separación de soluciones y el estimador-M IGGII no son capaces de hacer frente a un error en el pseudorango cuyo módulo aumenta con el tiempo.

Con ambas técnicas se obtiene una distribución del error de posición menos favorable que si no se aplicasen los métodos. Estos algoritmos detectan y excluyen medidas correctas, reduciendo considerablemente el número de medidas a utilizar en el cálculo de la posición.

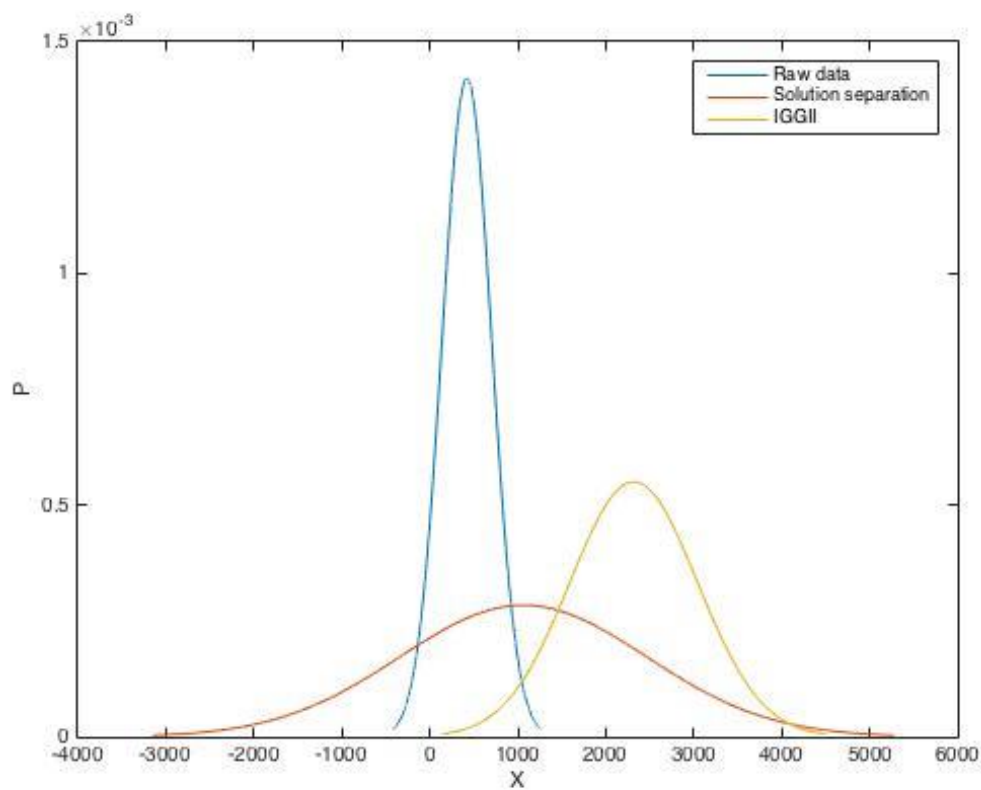


Figura 92: Funciones de densidad de probabilidad del error de posición para el caso estático con error creciente (1).

Por otro lado, el RAIM convencional y el estimador-M de Huber ofrecen resultados aceptables. Las distribuciones del error son similares a las de los casos con errores constantes.

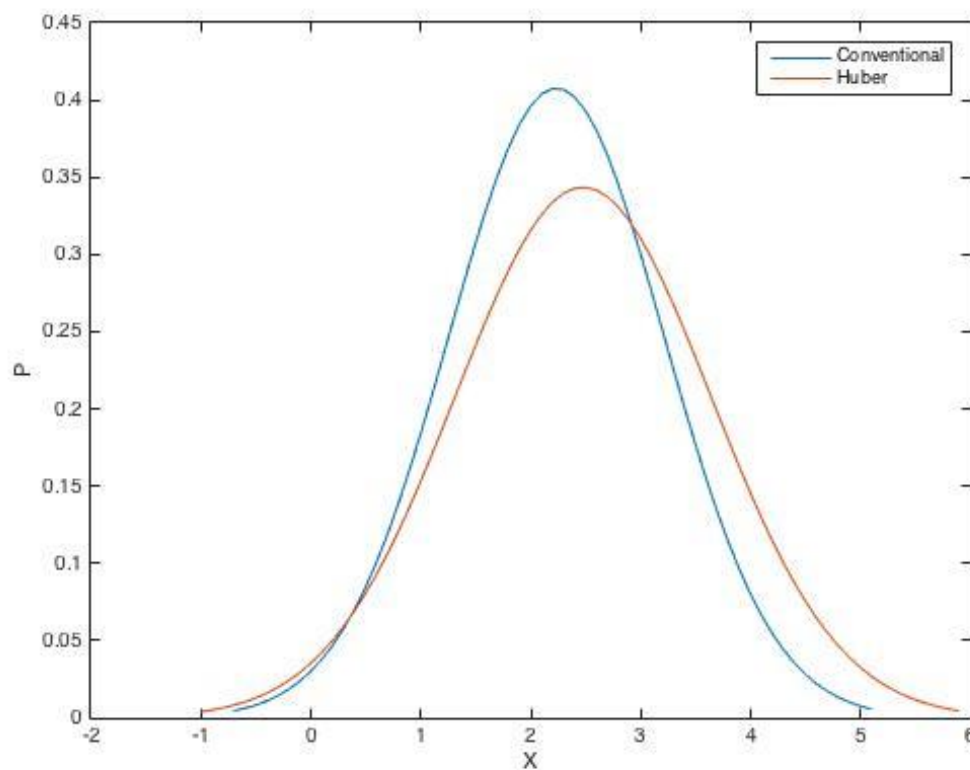


Figura 93: Funciones de densidad de probabilidad del error de posición para el caso estático con error creciente (2).

5.4.1.4.1 RAIM convencional

Los niveles de protección en la situación de un error cuyo módulo va aumentando con el tiempo se asemejan considerablemente a los del caso con un error de módulo constante.

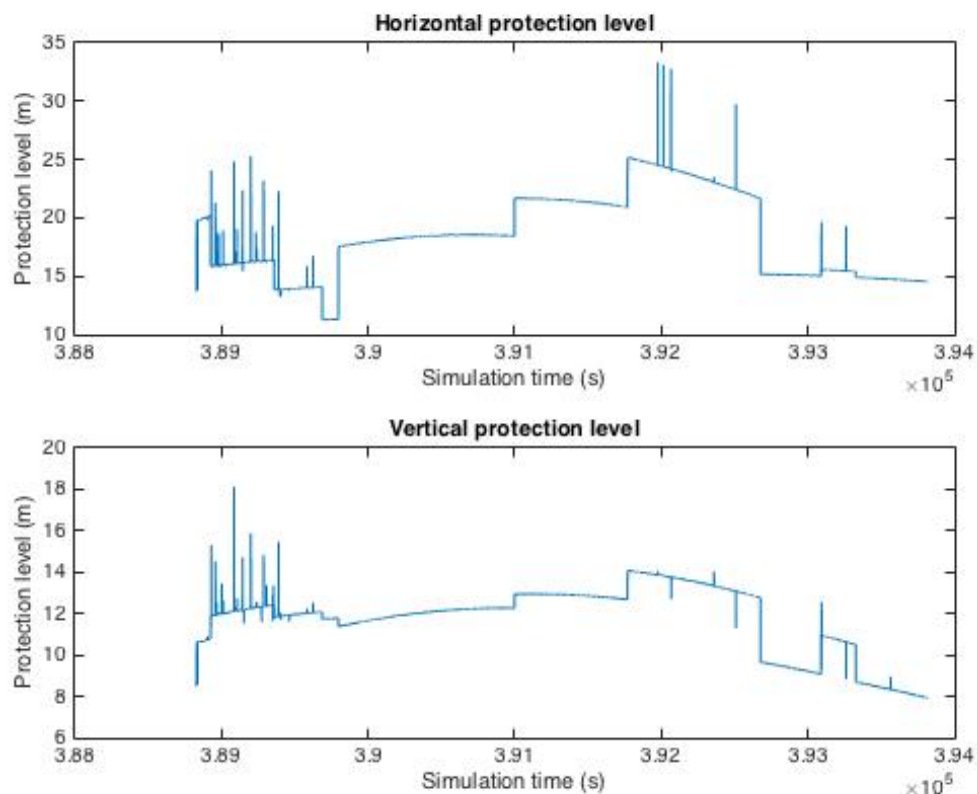


Figura 94: Niveles de protección para el RAIM convencional para el caso estático con error creciente.

Si se compara con la parte superior de la Figura 91, se puede apreciar como los niveles de protección se ven modificados con el número de satélites visibles. Lógicamente, cuando se reduce el número de satélites visibles el nivel de protección aumenta, y viceversa.

En las gráficas de los niveles de protección se pueden apreciar ciertos picos individuales que no aparecían en el caso del error constante. Esta diferencia reside en que debido al incremento en el tamaño del error, el método detecta y por tanto excluye de manera errónea ciertas medidas. Esta exclusión provoca incrementos puntuales en los niveles de protección debido a que el RAIM cuenta con un número menor de satélites para el cómputo del nivel.

Como se puede observar en la Figura 95, el número de satélites utilizados para el cómputo del nivel de protección suele ser uno menos que el número de satélites visibles. Esto se debe a que el método consigue detectar y excluir al satélite con la medida errónea. Sin embargo, en ciertos casos esta diferencia es de dos satélites, lo que provoca los incrementos puntuales de los niveles de protección.

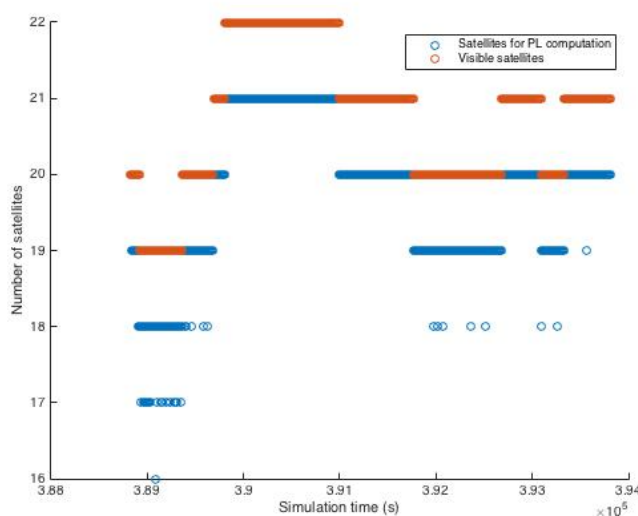


Figura 95: Número de satélites visibles y utilizados para el cálculo del nivel de protección para el RAIM convencional para el caso estático con error creciente.

Los diagramas de Stanford son idénticos a los del caso de un satélite con error constante en el pseudorango. Esto se debe a que el RAIM convencional consigue detectar y excluir la medida errónea en una mayoría de los casos por lo que el error se encuentra contenido sin llegar a superar los 10 m en ningún momento. Esta característica puede resultar tremendamente beneficiosa como resistencia al *spoofing*.

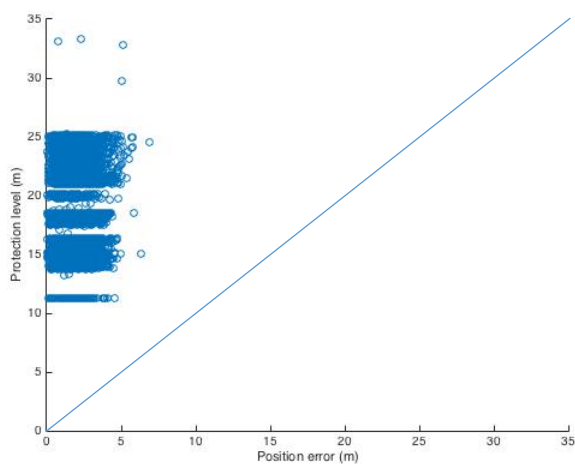


Figura 96: Diagrama de Stanford del componente horizontal para el RAIM convencional para el caso estático con error creciente.

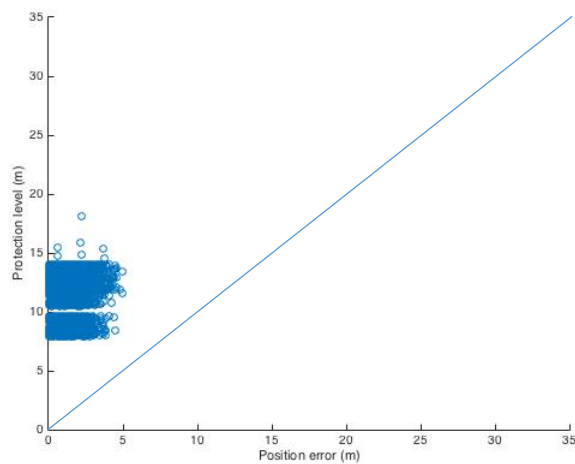


Figura 97: Diagrama de Stanford del componente vertical para el RAIM convencional para el caso estático con error creciente.

Además, es posible apreciar en los diagramas de Stanford los incrementos puntuales en los niveles de protección mencionados anteriormente.

5.4.1.4.2 RAIM basado en separación de soluciones

Como se ha comentado anteriormente, el RAIM basado en la separación de soluciones no responde de forma adecuada a la introducción de un error en el pseudorango que aumenta con el tiempo de la simulación. Se puede observar como los niveles de protección van decreciendo hasta alcanzar un valor nulo. Se da un valor nulo cuando el método no es capaz de calcular el nivel de protección.

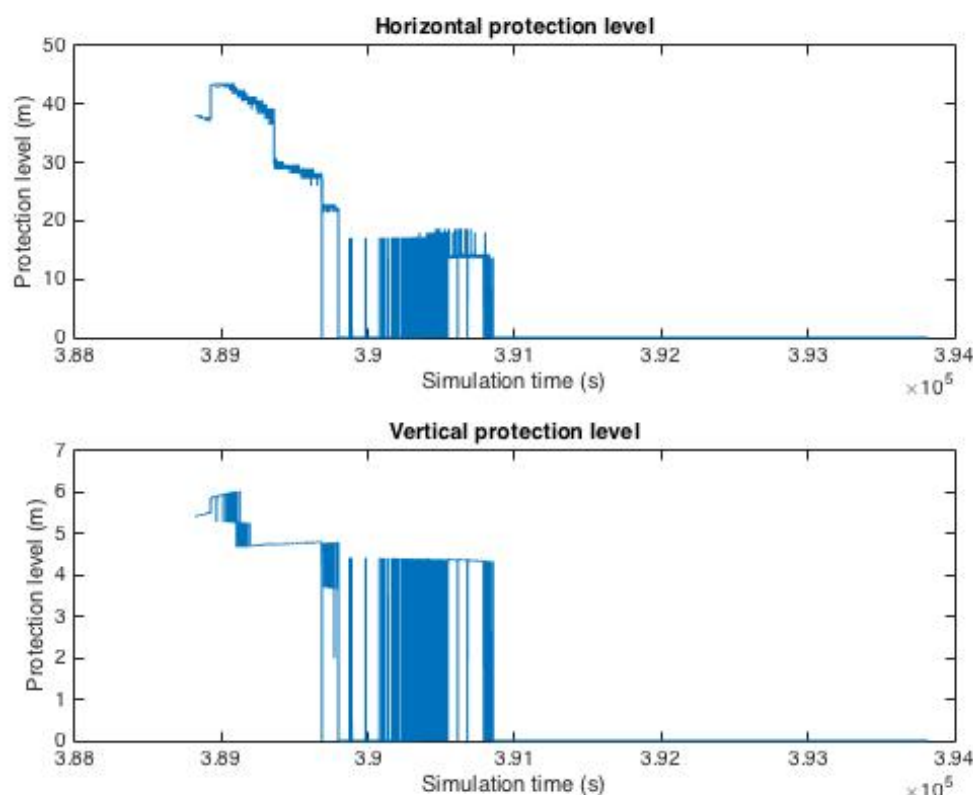


Figura 98: Niveles de protección para el RAIM basado en separación de soluciones para el caso estático con error creciente.

La incapacidad para calcular el nivel de protección se debe a que la técnica cuenta con un número de satélites insuficiente para llevarlo a cabo (menos de 6 para un sistema que utiliza dos constelaciones). En la figura siguiente se puede observar el número de satélites que utiliza el RAIM basado en la separación de soluciones para calcular el nivel de protección.

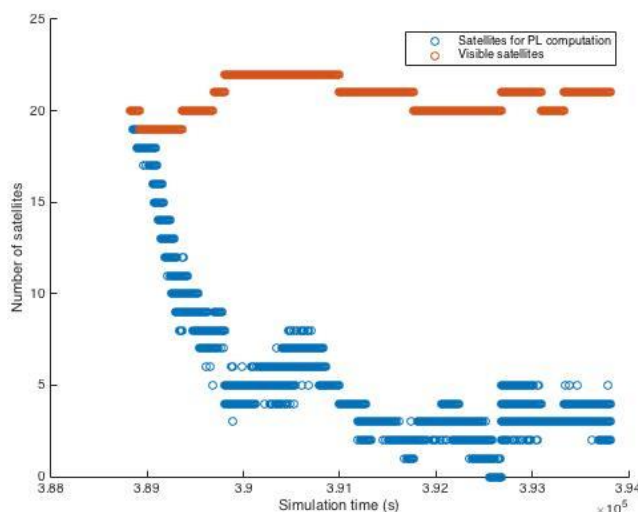


Figura 99: Número de satélites visibles y utilizados para el cálculo del nivel de protección para el RAIM basado en separación de soluciones para el caso estático con un error creciente.

Se puede observar una tendencia general decreciente en el número de satélites a utilizar para el cálculo de los niveles de protección. La exclusión de un número tan elevado de satélites se debe al tamaño del error en el pseudorange. La magnitud del error provoca que incluso los discriminadores de los satélites que no tienen medidas erróneas se vean incrementados hasta superar el valor de sus respectivos umbrales. Con el crecimiento del error, más satélites son excluidos de manera errónea, lo que reduce los satélites disponibles para calcular el nivel de precisión y la solución de posicionamiento. Esto explica la distribución del error tan desfavorable que proporciona este método.

Lógicamente, al ir excluyendo satélites que el método cree que contienen medidas erróneas, los niveles de protección van disminuyendo. Aunque se puede observar una tendencia decreciente en los niveles de protección, se producen aumentos y disminuciones del nivel en cada instante debido a que el método excluye y recupera satélites de forma constante.

Los diagramas de Stanford de las figuras siguientes muestran como debido a los bajos niveles de protección logrados, esta técnica se encuentra en la zona de operación engañosa.

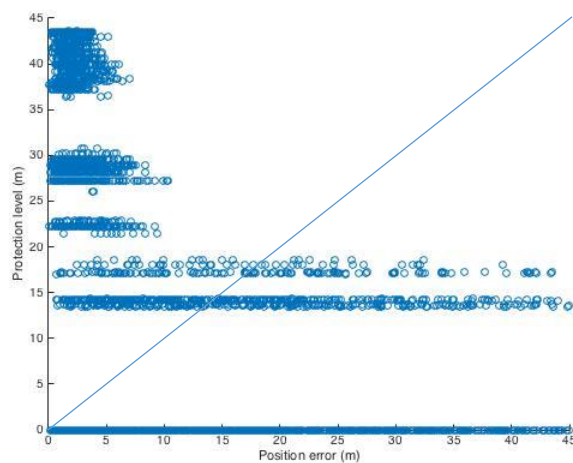


Figura 100: Diagrama de Stanford del componente horizontal para el RAIM basado en separación de soluciones para el caso estático con error creciente.

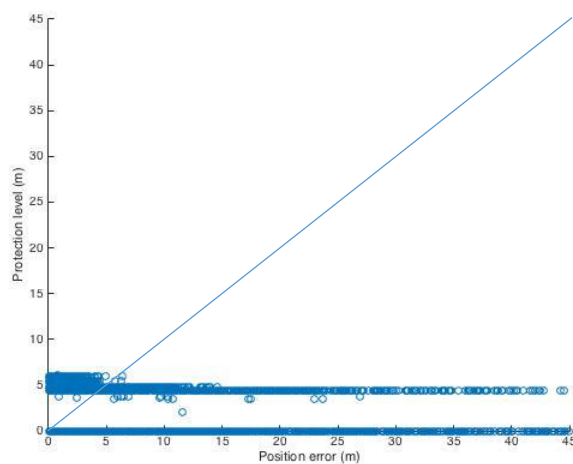


Figura 101: Diagrama de Stanford del componente vertical para el RAIM basado en separación de soluciones para el caso estático con error creciente.

5.4.1.4.3 IBPL

Como era de esperar, los niveles de protección aumentan de forma proporcional al error introducido. Aun así, se pueden apreciar los saltos que causan la variación del número de satélites disponibles y el instante en el que el error cambia de satélite.

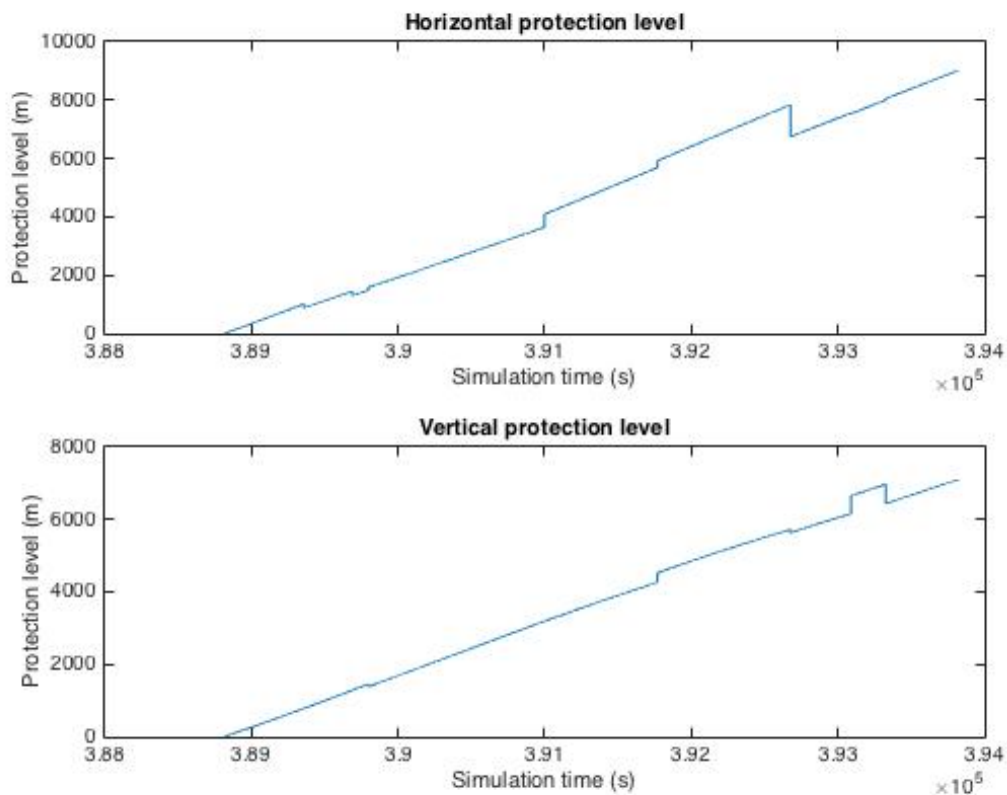


Figura 102: Niveles de protección para el IBPL para el caso estático con error creciente.

Lógicamente, con estos elevados niveles de protección, el algoritmo asegura la zona de operación nominal tanto para el componente horizontal como el vertical.

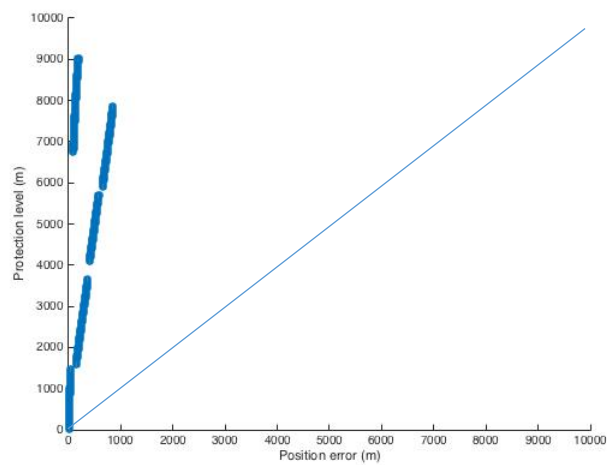


Figura 103: Diagrama de Stanford del componente horizontal para el IBPL para el caso estático con error creciente.

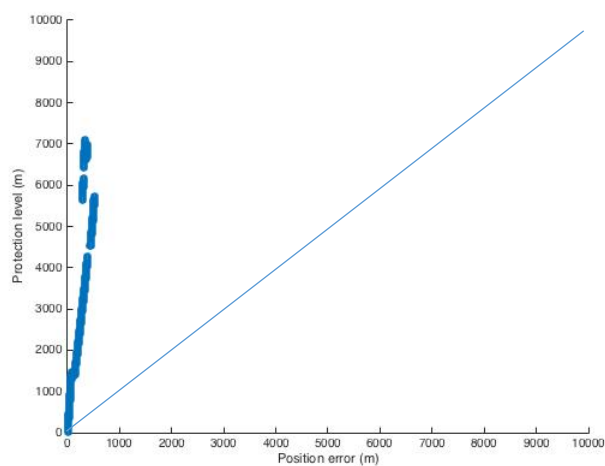


Figura 104: Diagrama de Stanford del componente vertical para el IBPL para el caso estático con error creciente.

5.4.2 Caso 2D

Para el caso bidimensional se utiliza la misma trayectoria elíptica que en el capítulo de selección de satélites (ver Figura 46).

Debido a que se utilizan los mismos satélites y se introducen errores idénticos en los mismos satélites que en el caso estático, los resultados obtenidos este caso son completamente análogos a los del caso estático.

Los resultados obtenidos en la evaluación individual para los distintos algoritmos se encuentran contenidos en el apéndice de este trabajo, con el objetivo de no sobrecargar en demasía la memoria y teniendo en cuenta que las conclusiones extraídas de dichas evaluaciones no difieren de los explicado para el caso estático en la sección 5.4.1. Los resultados en cuestión aparecen en Apéndices en la sección 8.2.1.

5.4.3 Caso 3D

Para el caso tridimensional se utiliza la misma trayectoria que en el capítulo de selección de satélites (ver Figura 50).

Debido a que se utilizan los mismos satélites y se introducen errores idénticos en los mismos satélites que en el caso estático, los resultados obtenidos este caso son completamente análogos a los del caso estático.

Los resultados obtenidos en la evaluación individual para los distintos algoritmos se encuentran contenidos en el apéndice de este trabajo, con el objetivo de no sobrecargar en demasía la memoria y teniendo en cuenta que las conclusiones extraídas de dichas evaluaciones no difieren de los explicado para el caso estático en la sección 5.4.1. Los resultados en cuestión aparecen en Apéndices en la sección 8.2.2.

5.5 Comparativa y valoración

La introducción de varios errores constantes en las medidas de los satélites, además de un error cuyo valor aumenta con el tiempo de la simulación, permite comparar el desempeño de los métodos de detección de medidas erróneas que se han tratado en este trabajo. La variedad en las medidas erróneas y las distintas situaciones del receptor (estático, trayectoria bidimensional y tridimensional) se adecúan a la realidad en la navegación de las aplicaciones de hoy en día.

El RAIM convencional ofrece una tasa de detección y exclusión de medidas erróneas elevadas independientemente del número de satélites con medidas erróneas. En consecuencia, mediante este método se consigue mantener unos errores de posición contenidos, que junto a los niveles de protección de este método aseguran la operación en la zona nominal del diagrama de Stanford.

La existencia de un error con módulo creciente no altera el funcionamiento ni el desempeño de la técnica convencional. En estas circunstancias el RAIM convencional excluye de manera equivocada ciertas medidas debido al tamaño del error introducido, lo que provoca variaciones puntuales en los niveles de protección. A pesar de ello, la solución de posicionamiento proporcionada tiene una buena prestación ya que el algoritmo se recupera excluyendo el satélite adecuado o incrementa el valor del nivel de protección, permitiendo alertar al usuario de que la solución se está viendo comprometida.

Por otro lado, el RAIM basado en la separación de soluciones no consigue hacer frente a la aparición de varios errores de forma simultánea. Con la introducción de dos o tres errores en distintos satélites, esta técnica no es capaz de detectar las medidas erróneas, por lo que no excluye dichas medidas. Esta no exclusión provoca que los niveles de protección sufran disminuciones al contar con un mayor número de satélites para su cómputo. Al no lograr la exclusión de las medidas, los errores de posición se ven afectados de manera negativa aumentando la media y desviación típica de la distribución.

La reducción de los niveles de protección y el aumento de los errores de posición provoca que, en los diagramas de Stanford desarrollados, se trabaje por momentos en la zona de operación engañosa. Esto significa que con este algoritmo se podrían producir, en esta situación, errores de valor elevado sin que se alertase al usuario. Este comportamiento introduce el riesgo de que, al usar este algoritmo, el posicionamiento por GNSS podría verse comprometido de forma inadvertida.

Además, el RAIM basado en separación de soluciones tampoco alcanza a detectar un error cuyo módulo va aumentando con el tiempo. El crecimiento del error provoca la exclusión equivocada de medidas correctas de manera paulatina hasta que el método no cuenta con un número suficiente de satélites para calcular los niveles de protección. Ante un error considerablemente grande, el RAIM basado en separación de soluciones no es capaz ni de detectar y excluir la medida errónea, ni de computar el nivel de protección.

El estimador-M de Huber se posiciona como el segundo mejor método (de entre los evaluados en este trabajo) para la detección y exclusión de medidas erróneas. De forma similar al RAIM convencional, esta técnica reacciona de forma favorable ante la inclusión de varias medidas erróneas consiguiendo excluirlas con éxito. Asimismo, la técnica también detecta y excluye la medida errónea independientemente del tamaño del error. Por ello, los errores de posición se mantienen entre márgenes aceptables.

Sin embargo, el estimador-M IGGII no ofrece resultados favorables como el Huber. Al igual que el RAIM basado en separación de soluciones, al añadir varios errores en las medidas el estimador IGGII no es capaz de detectar y excluir las medidas pertinentes. Esta circunstancia se ve reflejada en los errores de posición y en su distribución, no siendo comparables con los del RAIM convencional y el estimador de Huber. Debido a la forma de función IGGII, al incrementarse el tamaño del error introducido, se eliminan de manera equívoca medidas correctas. Además, también se atenúa la aportación de otras, por lo que

el cálculo de la posición mediante los mínimos cuadrados cuenta con escasos satélites, proporcionando una solución con errores de posición elevados.

La última técnica analizada en este trabajo ha sido el IBPL, que a pesar de no llevar a cabo detección y exclusión alguna, proporciona unos niveles de protección que pueden ser evaluados. Los niveles de protección del IBPL cuentan siempre con valores mayores que los niveles de las técnicas RAIM. Estos elevados niveles de protección aseguran en todas las circunstancias valoradas en este trabajo la operación en la zona nominal del diagrama de Stanford. Es decir, los niveles de protección calculados por el IBPL son siempre superiores a los errores de posición. Sin embargo, para ciertas aplicaciones en las que se establece un nivel de alerta para limitar el error de posición o el nivel de protección, este método no sería válido.

El método IBPL es por tanto una técnica conservadora ,pero que puede resultar de utilidad para la detección de ataques de spoofing o de situaciones de error. Puede resultar muy eficaz a la hora de detectar desviaciones en la solución de posicionamiento. Pero requiere de un reajuste en la definición de los niveles de alerta respecto del RAIM convencional.

La realización de simulaciones en los casos de receptores que siguen una trayectoria bidimensional y tridimensional han servido como confirmación de que los métodos evaluados en este trabajo para la detección de medidas erróneas también son capaces de actuar en estas situaciones, que concuerdan con la aplicación real de los sistemas GNSS. Los resultados ofrecidos por estas simulaciones son idénticos a los del caso estático, con la excepción de un ligero aumento del error de posición en algunos de los casos tratados.

La tabla que se muestra a continuación contiene la comparación de las distintas técnicas de detección de medidas erróneas evaluadas en este trabajo. Además, presenta para cada técnica el caso con un satélite con medida errónea y el caso con tres satélites con medidas erróneas.

<i>Técnica [número de satélites con medidas erróneas]</i>	Media error posición (m)	Desviación típica error posición (m)	Nivel de protección horizontal medio (m)	Nivel de protección vertical medio (m)
<i>RAIM Convencional [1]</i>	2.4753	1.2127	13.2235	9.7750
<i>RAIM Convencional [3]</i>	3.3994	2.2195	16.4134	12.0225
<i>RAIM basado en separación de soluciones [1]</i>	3.8536	1.7422	29.3871	5.8822
<i>RAIM basado en separación de soluciones [3]</i>	6.7479	2.5642	29.2176	5.8596
<i>Estimador Huber [1]</i>	2.4984	1.1710	-	-
<i>Estimador Huber [3]</i>	3.9914	2.9906	-	-
<i>Estimador IGGII [1]</i>	3.3664	1.6093	-	-
<i>Estimador IGGII [3]</i>	5.7552	31.2332	-	-
<i>IBPL [1]</i>	4.4758	1.8073	66.4405	53.8869
<i>IBPL [3]</i>	6.7470	2.6859	107.6247	87.1284

Tabla 20: Comparación del caso estático de las técnicas de detección de medidas erróneas.

Los valores de la media y la desviación típica de los errores de posición evidencian que las técnicas RAIM basado en separación de soluciones y el estimador IGGII no son capaces de hacer frente a más de una medida errónea. En cambio, el RAIM convencional y el estimador Huber presentan resultados adecuados con la introducción de medidas erróneas. Lógicamente, al no llevar a cabo ni detección ni exclusión de medidas erróneas, el error de posición proporcionado por el método IBPL se ve severamente afectado por la introducción de un mayor número de medidas erróneas. Por tanto, resultaría interesante aplicar el IBPL al conjunto de medidas después de realizar una detección y exclusión por medio del RAIM convencional o el estimador de Huber.

Si se comparan los niveles de protección, es posible apreciar como el valor medios de los niveles de protección del RAIM convencional sufren un incremento en el caso de tres satélites con medidas erróneas. Este cambio se debe a que el algoritmo detecta un mayor número de satélites con medidas erróneas y los elimina consiguientemente, lo que provoca un aumento en los niveles de protección.

Este no es el caso del RAIM basado en separación de soluciones, que no es capaz de detectar las medidas erróneas y por tanto no excluye los satélites, por lo que los niveles de protección se mantienen constantes. El método IBPL presenta unos valores de los niveles de protección sumamente elevados para hacer frente a los errores de posición mencionados anteriormente.

CAPÍTULO 6:

Conclusiones

Las conclusiones presentadas en este capítulo pretenden completar las ya existentes en los apartados de “Valoración y comparación” que aparecen al final de cada uno de los tres capítulos anteriores.

En este trabajo se han evaluado varias maneras de optimizar la navegación por satélite, intentando aprovechar las ventajas que proporcionan los sistemas multiconstelación. Para ello, se han llevado a cabo distintas simulaciones en el entorno matemático Matlab. Para poner a prueba los algoritmos y técnicas de optimización se han utilizado las medidas transmitidas por satélites de las constelaciones GPS y Galileo. Estas medidas han sido generadas por un simulador de navegación por satélite desarrollado internamente en GMV, que incluye todos los errores e influencias físicas de la navegación por satélite con el objetivo de asemejarse lo máximo a la realidad.

La presencia de saltos de ciclo en las medidas provoca una pérdida en la precisión del sistema, por lo que interesa detectar esta presencia para descartar las medidas afectadas. Las simulaciones llevadas a cabo en este trabajo han dejado patente las ventajas que los algoritmos de frecuencia múltiple sobre los de frecuencia simple.

Los algoritmos de frecuencia simple requieren de una modificación continua del valor del umbral para adaptarse a los distintos tamaños de saltos de ciclo. En la práctica, esto es completamente inviable ya que es imposible estimar el tamaño del salto de ciclo. Además, estos algoritmos no son capaces de hacer frente a los efectos de la ionosfera.

En cambio, los algoritmos de frecuencia múltiple son capaces de lidiar con los efectos de la ionosfera, y en el caso del Melbourne-Wübbena, también de discriminar variaciones rápidas de ionosfera de errores en las medidas. La combinación de Melbourne-Wübbena presenta un grado de detección de saltos muy elevado. Además, la inmunidad frente al ruido de este algoritmo es muy favorable, operando de manera adecuada incluso en situaciones con satélites

de poca elevación, y por tanto, niveles de ruido térmico significativamente elevados.

En la Tabla 21 se muestra un resumen de la valoración de los algoritmos de detección de saltos de ciclo evaluados en este trabajo.

Técnica	Probabilidad detección correcta	Inmunidad frente al ruido	Valoración
Ajuste polinómico	Baja	Media	Presenta resultados poco favorables y requiere la modificación del valor del umbral.
Media y desviación	Muy baja	Muy baja	Presenta resultados muy inferiores al resto y requiere la modificación del valor del umbral.
Combinación libre de geometría	Muy alta	Alta	Presenta muy buenos resultados excepto frente a variaciones rápidas de la ionosfera.
Combinación Melbourne-Wübbena	Alta	Muy alta	Presenta muy buenos resultados excepto cuando existen dos saltos de ciclo con el mismo tamaño.

Tabla 21: Resumen valoración algoritmos de detección de saltos de ciclo.

La cantidad elevada de satélites disponibles que maneja un sistema multiconstelación provoca que se cuente con una mejor geometría para llevar a cabo la navegación y se obtiene mayor precisión en la solución de esta. Sin embargo, el uso de un número excesivamente elevado de satélites para la navegación provoca un incremento en la carga computacional del sistema, debido a que el método de mínimos cuadrados se ve obligado a invertir matrices de órdenes considerablemente elevados. Además, en ocasiones las medidas se utilizan en otros algoritmos de navegación lo que puede llevar a un severo perjuicio en término de tiempo de ejecución. Por esto, para necesaria la introducción de un método que seleccione un subconjunto de satélites de entre los disponibles que sea capaz de proporcionar una precisión adecuada y que no requiera de unos recursos computacionales elevados.

A lo largo de este trabajo se han estudiado y evaluado seis algoritmos de selección de satélites distintos. Las simulaciones llevadas a cabo han dejado patente el hecho de que en un sistema multiconstelación no se puede llevar a cabo la selección por la vía óptima, es decir, probando todas las combinaciones posibles de satélites para encontrar la que proporcione mayor precisión en la solución. La cantidad de satélites disponibles que ofrece un sistema multiconstelación provoca que si se sigue este método, se necesite de un tiempo considerablemente elevado para llevar a cabo la selección.

De entre los algoritmos estudiados, dos de ellos (WPDOP y EPR) presentan soluciones de navegación con características muy similares al método óptimo, pero necesitando de menos carga computacional y por tanto, llevando a cabo la selección en tiempos más breves. Además el algoritmo WPDOP es capaz de modificar el número de satélites de la selección, lo que le permite adaptarse a distintas necesidades de precisión.

En la Tabla 22 se muestra un resumen de la valoración de los algoritmos de detección de saltos de ciclo evaluados en este trabajo.

Técnica	Error de posición	Tiempo requerido	Valoración
OSSA	Muy favorable	Muy poco favorable	Requiere de un tiempo excesivo para llevar a cabo la selección.
HEA	Poco favorable	Muy favorable	Presenta resultados poco favorable con respecto al error de posición.
MVA	Muy poco favorable	Muy favorable	Utiliza únicamente cuatro satélites lo que le priva de los beneficios de los sistemas multiconstelación.
Dartboard DOP	Poco favorable	Muy favorable	Presenta resultados poco favorable con respecto al error de posición.
WPDOP	Muy favorable	Favorable	Presenta muy buenos resultados y ofrece flexibilidad para el número de satélites en la selección.
EPR	Muy favorable	Favorable	Presenta muy buenos resultados aprovechando las ventajas de otros algoritmos.

Tabla 22: Resumen valoración algoritmos de selección de satélites.

Otra ventaja importante de los sistemas multiconstelación, es el uso de las medidas redundantes disponibles para llevar a cabo la detección y exclusión de medidas erróneas o anómalas. Mediante una técnica de detección de medidas erróneas, el sistema de navegación puede descartar medidas que de otra forma influirían de forma negativa en la navegación. Debido al cada vez más común uso de la navegación por satélite en entornos urbanos, resulta conveniente el análisis con la existencia de más de una medida errónea en el conjunto de medidas.

Para ello, en este trabajo se han evaluado cinco métodos diferentes. Se han analizado dos técnicas basadas en la estimación robusta. Si la distribución que siguen los errores de navegación no se asemeja a una normal, el comportamiento de los residuos no es el adecuado. Debido a que las técnicas RAIM se basan en estos residuos, los resultados pueden no ser favorables en todos los casos. Por ello, los estimadores robustos se presentan como una

alternativa interesante, ya que son capaces de adaptarse a cualquier distribución del error.

Las pruebas llevadas a cabo en este trabajo han evidenciado la importancia de la función del estimador escogido. Los dos estimador evaluados en este trabajo han ofrecido resultados muy dispares en las simulaciones realizadas. A diferencia de los métodos RAIM, los estimadores robustos modifican la matriz de peso que se utiliza en método de los mínimos cuadrados, lo que significa que son capaces de alterar las contribuciones de las medidas del sistema a la hora de computar la solución de navegación.

La introducción de errores múltiples en el conjunto de datos y de un error cuyo módulo va aumentando con el tiempo de navegación evidenciaron que el RAIM basado en separación de soluciones y el estimador IGGII no son capaces de hacer frente a estas situaciones. Por otro lado, el RAIM convencional y estimador de Huber si completan la detección y exclusión de medidas erróneas con éxito, aprovechándose de la redundancia de medidas de tal forma que el error de posición se ve afectado ligeramente por la introducción de errores en las medidas.

El análisis de la técnica IBPL (Integrity Based Protection Level) contrasta con el resto de métodos evaluados ya que no lleva a cabo ni detección ni exclusión de medidas erróneas, limitándose a computar los niveles de protección para valorar la integridad del sistema. Aunque esta técnica asegura la operación en la zona nominal del diagrama de Stanford (donde el nivel de protección mayor que el error de posición), no es adecuada para cualquier aplicación que establezca límites de alerta en el nivel de protección o en el error de posición.

En la Tabla 23 se muestra un resumen de la valoración de los algoritmos de detección de saltos de ciclo evaluados en este trabajo.

Técnica	Error de posición	Niveles de protección	Valoración
RAIM convencional	Muy favorable	Normalmente superiores al error de posición	Presenta muy buenos resultados independientemente del número de errores y su tamaño.
RAIM separación soluciones	Poco favorable	No siempre superiores al error de posición	No consigue hacer frente a múltiples errores y para errores de cierto tamaño excluye gran parte de los satélites
Estimador de Huber	Muy favorable		Presenta muy buenos resultados independientemente del número de errores y su tamaño.
Estimador IGGII	Poco favorable		No consigue hacer frente a múltiples errores ni errores de gran tamaño.
IBPL	Muy poco favorable	Siempre superiores al error de posición	No lleva a cabo detección ni exclusión de medidas erróneas por lo que el error de posición es siempre elevado.

Tabla 23: Resumen valoración algoritmos de detección de medidas erróneas.

Varias de las técnicas evaluadas en este trabajo han demostrado poder hacer frente a los principales problemas presentes en la navegación por satélite de hoy en día. Los sistemas multiconstelación pueden ser de gran interés ya que la mayor disponibilidad de satélites significa una mejora en la precisión de la solución de navegación y permite al sistema ofrecer una mayor disponibilidad de servicio. Además, se puede hacer uso de la redundancia de medidas para llevar a cabo tareas de detección y exclusión de medidas erróneas que aseguren la integridad del sistema sin comprometer el desempeño del mismo.

Se puede afirmar por tanto, que el uso de sistema multiconstelación conlleva a sistemas de navegación superiores en términos de precisión, integridad y disponibilidad de servicio. Por otro lado, es posible llevar a cabo la detección de saltos de ciclo mediante el uso de métodos que aseguran un elevado grado de detección y una operación adecuada incluso en presencia de niveles de ruido significativos, de forma que las medidas afectadas por los saltos puedan ser descartadas.

CAPÍTULO 7: Bibliografía

- Li, X., Zhang, X., Ren, X., Fritsche, M., Wickert, J., & Schuh, H. (2015). Precise positioning with current multiconstellation Global Navigation Satellite Systems: GPS, GLONASS, Galileo and BeiDou. *Scientific Reports*.
- Azaola, M., Calle, D., Mozo, A., & R. Piriz. (2016). *Autonomous Isotropy-Based Integrity Using GPS and GLONASS*. GMV.
- Blanco-Delgado, N., & Nunes, F. D. (2010). *Satellite Selection based on WDOP Concept and Convex Geometry*.
- D'Angelo, P., Diez, J., Fernández, A., Gabaglio, V., & Marradi, L. (2005). *Navigation Algorithm Optimisation for Combined Galileo/GPS Receivers with GRANADA Environment and Navigation Simulator*.
- Fernández, A., D'Angelo, P., Diez, J., Marradi, L., & Gabaglio, V. (2005). Navigation Algorithm Optimisation for Combined Galileo/GPS Receivers with the GRANADA Environment and Navigation Simulator. ION GNSS 18th International Technical Meeting of the Satellite Division.
- Gaglione, S., Innac, A., Pastore, S., & Angrisano, A. (2017). *Robust estimation methods applied to GPS in harsh environments*. Navigation Conference (ENC), 14-25.
- Gaglione, S., Innac, A., Pastore, S., & Angrisano, A. (2017). *RObust estimation methods applied to GPS in harsh environments*.
- Gao, X., Dai, W., Song, Z., & Cai, C. (2016). Reference satellite selection method for GNSS high-precision relative positioning. *Geodesy and Geodynamics*.
- GMV. (2011). *Navipedia*. Obtenido de Integrity: <http://www.navipedia.net/index.php/Integrity>
- Innac, A., Bhuiyan, M. Z., Söderholm, S., Kuusniemi, H., & Gaglione, S. (2016). Reliability testing for multiple GNSS measurement outlier detection.
- J. Sanz Subirana, J. J.-P. (2012). *Navipedia*. Recuperado el 2017, de GPS and Galileo Satellite Coordinates Computation: http://www.navipedia.net/index.php/GPS_and_Galileo_Satellite_Coordinates_Computation
- J. Sanz Subirana, J. J.-P. (2012). *Navipedia*. Recuperado el 2017, de GLONASS Satellite Coordinates Computation: http://www.navipedia.net/index.php/GLONASS_Satellite_Coordinates_Computation

- J. Sanz Subirana, J. J.-P. (2013). *Navipedia*. Recuperado el 2017, de Examples of single frequency Cycle-Slip Detectors: http://www.navipedia.net/index.php/Examples_of_single_frequency_Cycle-Slip_Detectors
- J. Sanz Subirana, J. J.-P. (2013). *Navipedia*. Recuperado el 2017, de Detector based in code and carrier phase data: The Melbourne-Wübbena combination: http://www.navipedia.net/index.php/Detector_based_in_code_and_carrier_phase_data:_The_Melbourne-W%C3%BCbbena_combination
- J. Sanz Subirana, J. J.-P. (2013). *Navipedia*. Recuperado el 2018, de Tropospheric Delay: http://www.navipedia.net/index.php/Tropospheric_Delay
- J. Sanz Subirana, J. J.-P. (2014). *Navipedia*. Recuperado el 2017, de Detector based in carrier phase data: The geometry-free combination: http://www.navipedia.net/index.php/Detector_based_in_carrier_phase_data:_The_geometry-free_combination
- J. Sanz Subirana, J. J.-P. (2016). *Navipedia*. Recuperado el 2018, de Ionospheric Delay: http://www.navipedia.net/index.php/Ionospheric_Delay
- Joerger, M., Chan, F.-C., & Pervan, B. (2014). Solution Separation Versus Residual-Based RAIM.
- Kaplan, E. D., & Hegarty, C. J. (2006). *Understanding GPS. Principles and Applications*. Artech House.
- Knight, N. L., & Wang, J. (2009). *A Comparison of Outlier Detection Procedures and Robust Estimation Methods in GPS Positioning*.
- Kong, J., Mao, X., & Li, S. (2014). *BDS/GPS Satellite Selection Algorithm based on Ployhedron Volumetric Method*.
- Martineau, A., Macabiau, C., Nikiforov, I., & Roturier, B. (2014). Performance of Receiver Autonomous Integrity Monitoring (RAIM) for vertically guided approaches. HAL.
- Meneghini, C., & Parente, C. (2017). *Advantages of Multi GNSS Constellation: GDOP Analysis for GPS, GLONASS and Galileo Combinations*.
- Meng, F., Zhu, B., & Wang, C. (2016). *Research of Fast Satellite Selection Algorithm for Multi-Constellation*.

- Meng, F., Zhu, B., & Wang, S. (2013). *A New Fast Satellite Selection Algorithm for BDS-GPS Receivers*.
- Miaoyan, Z., Jun, Z., & Yong, Q. (2008). *Satellite Selection for Multi-Constellation*.
- National Oceanic and Atmospheric Administration. (s.f.). *Space Weather Prediction Center*. Recuperado el 2018, de Ionospheric Scintillation: <http://www.swpc.noaa.gov/phenomena/ionospheric-scintillation>
- Nie, Z., Gao, Y., Wang, Z., & Ji, S. (2016). *A new method for satellite selection with controllable PDOP threshold*.
- Pan, L., Cai, C., Santerre, R., & Zhang, X. (2016). *Performance evaluation of single-frequency point positioning with GPS, GLONASS, BeiDou and Galileo*.
- Roongpiboonsopit, D., & Karimi, H. A. (2009). *A Multi-Constellations Satellite Selecion Algorithm for Integrated Global Navigation Satellite Systems*. Journal of Intelligent Transportation Systems: Technology, Planning, and Operations, 127-141.
- Roongpiboonsopit, D., & Karimi, H. A. (2013). *A Multi-Constellations Satellite Selection Algorithm for Integrated Global Navigation Satellite Systems*.
- Ruckstuhl, A. (2016). *Robust Fitting of Parametric Models Based on M-Estimation*.
- Shabnam, M., Chowdhury, I. H., Tushar, Z. H., Sultana, S., & Hossam-E-Haider, M. (2017). *Performance Evaluation of GNSS Receiver in Multi-constellation System*. International Conference on Electrical, Computer and Communication Engineering (ECCE).
- Teng, Y., & Wang, J. (2014). *New Characteristics of Geometric Dllution of Precision (GDOP) for Multi-GNSS Constellations*.
- Teng, Y., & Wang, J. (2015). *A closed-form formula to calculate geomtric dilution of precision (GDOP) for multi-GNSS constellations*.
- Teng, Y., & Wang, J. (2016). *Some Remarks on PDOP and TDOP for Multi-GNSS Constellations*.
- Tian, A., Fu, C., Zhang, S., & Zhou, M. (2013). *A New Satellite Selection Algorithm in GNSS*.
- Xu, G. (2007). *GPS. Theory, Algorithms and Applications*. Springer.

- Xu, G. (2007). *GPS. Theory, Algorithms and Applications*. Springer.
- Yang, Y., & Xu, J. (2016). GNSS receiver autonomous integrity monitoring (RAIM) algorithm based on robust estimation. *Geodesy and Geodynamics*.

CAPÍTULO 8: Apéndices

8.1 Selección de satélites

En esta sección se van a presentar las figuras correspondientes a los casos de trayectoria bidimensional y tridimensional para los algoritmos de selección de satélites.

8.1.1 Caso 2D

8.1.1.1 *Highest Elevation Algorithm (HEA)*

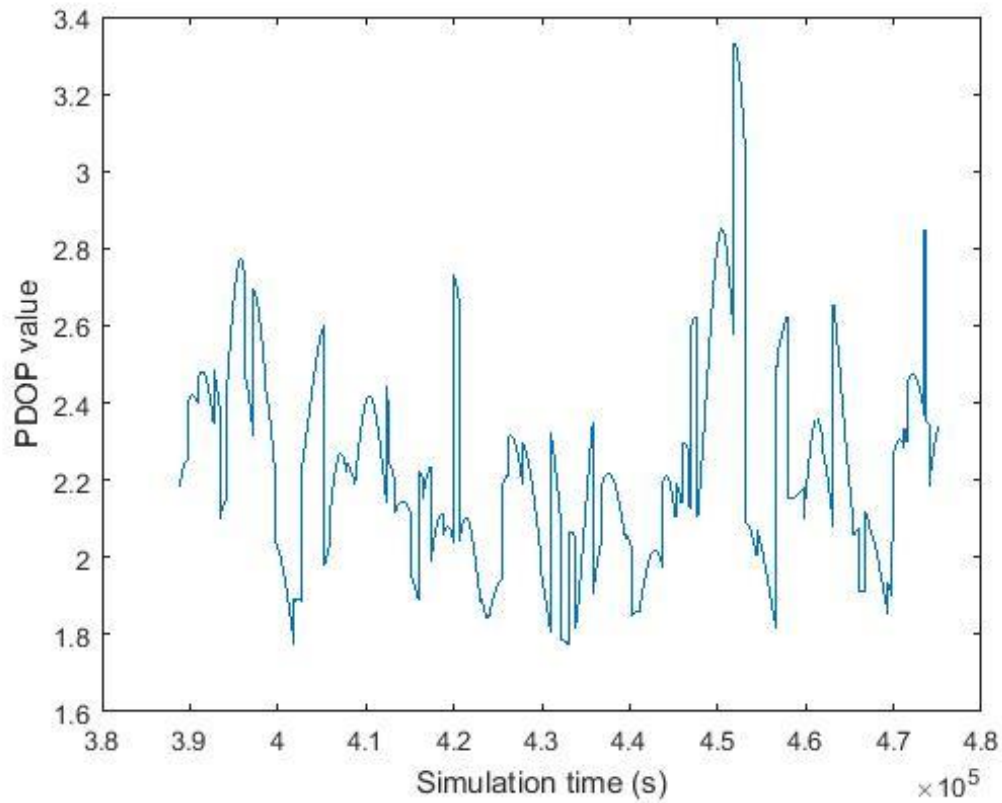


Figura 106: Valores del PDOP para el algoritmo HEA para el caso de trayectoria bidimensional.

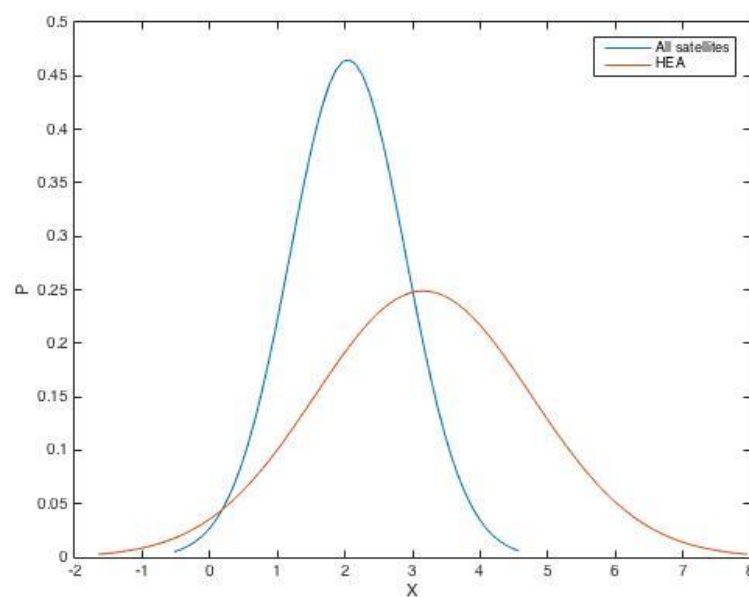


Figura 105: FDP del error de posición para el algoritmo HEA para el caso de trayectoria bidimensional.

8.1.1.2 *Maximum Volume Algorithm (MVA)*

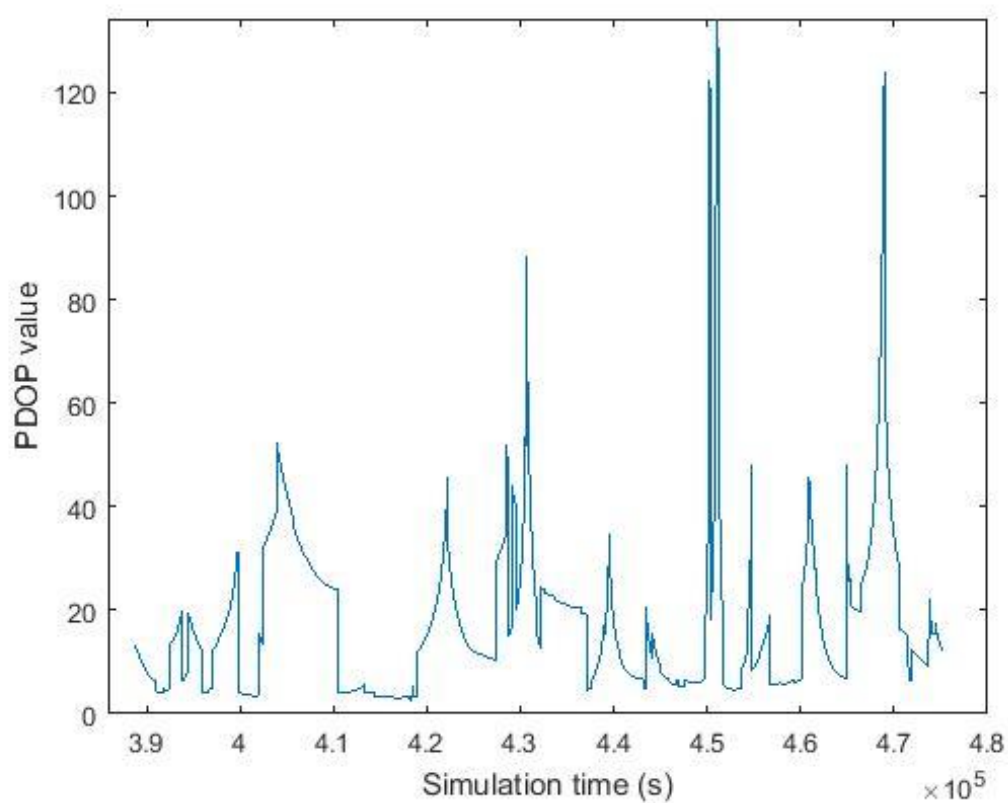


Figura 107: Valores del PDOP para el algoritmo MVA para el caso de trayectoria bidimensional.

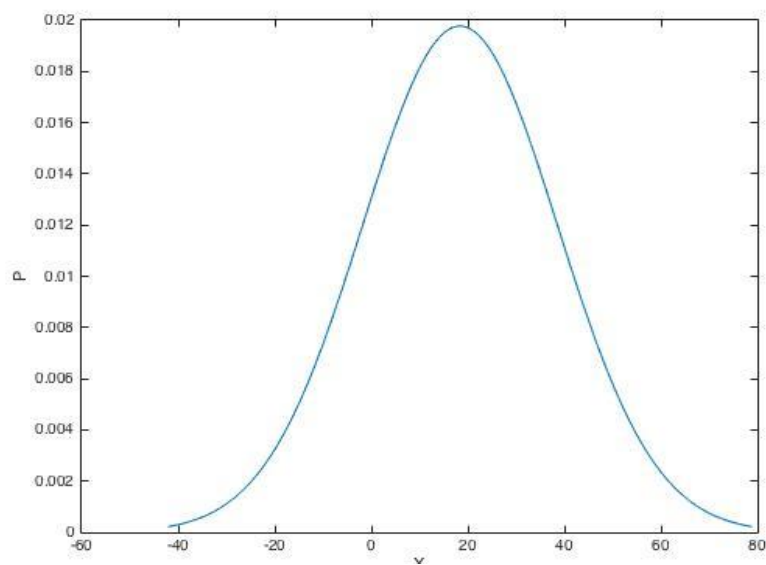


Figura 108: FDP del error de posición para el algoritmo MVA para el caso de trayectoria bidimensional.

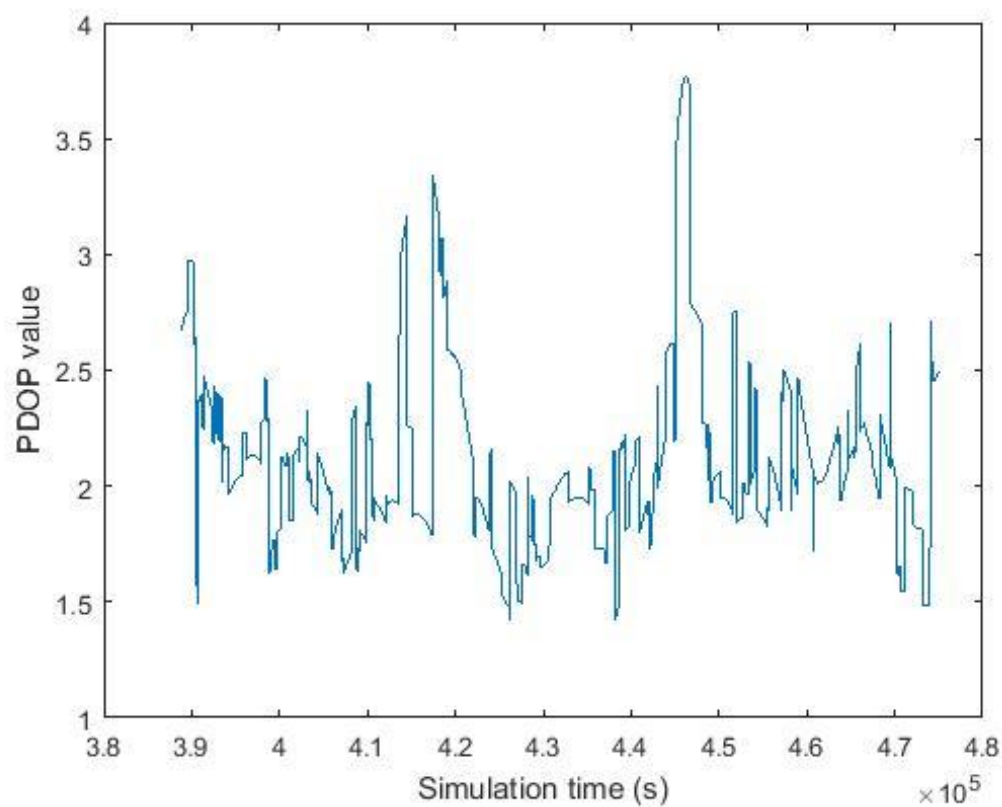


Figura 110: Valores del PDOP para el algoritmo Dartboard DOP para el caso de trayectoria bidimensional.

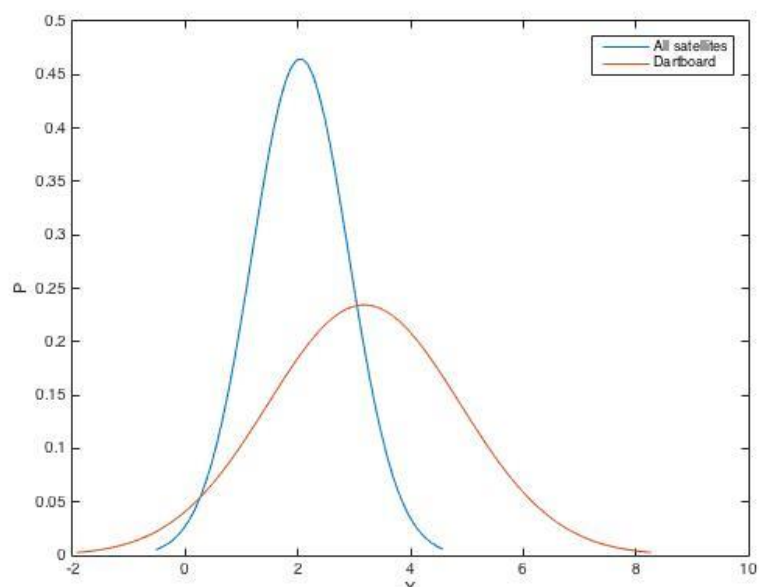


Figura 109: FDP del error de posición para el algoritmo Dartboard DOP para el caso de trayectoria bidimensional.

8.1.1.3 Dartboard DOP Algorithm

8.1.1.4 WPDOP Algorithm

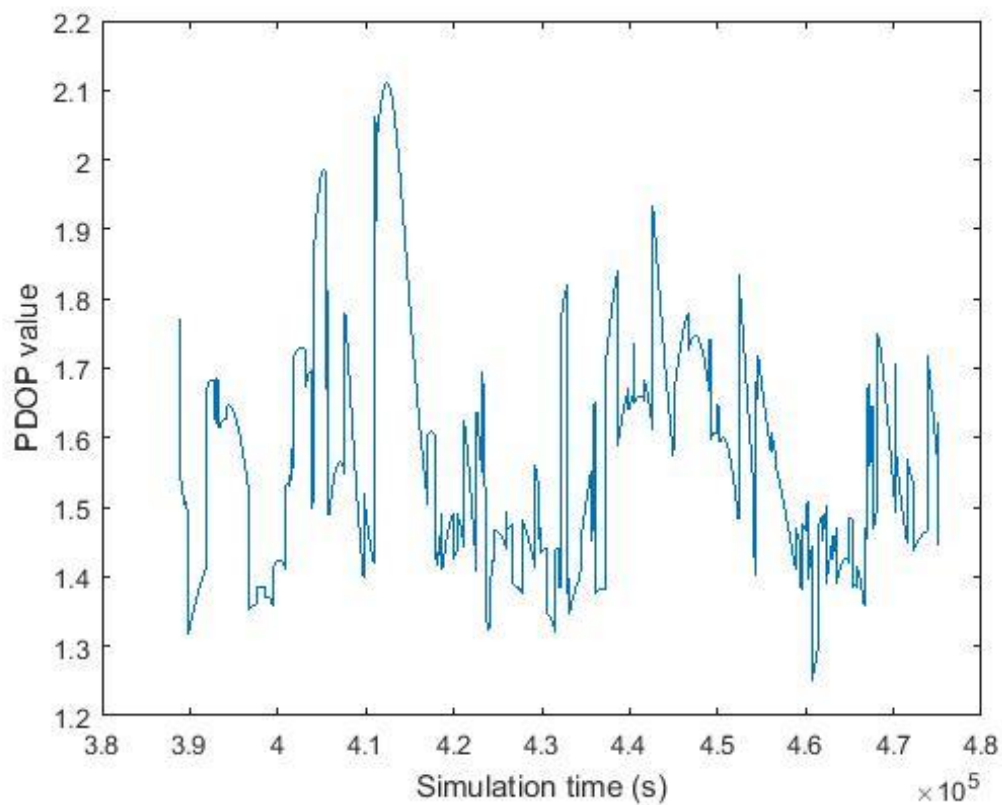


Figura 111: Valores del PDOP para el algoritmo WPDOP para el caso de trayectoria bidimensional.

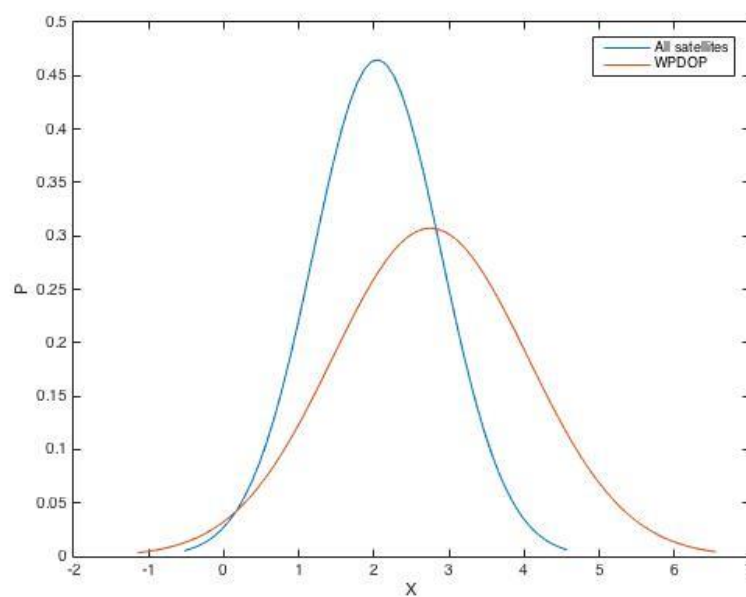


Figura 112: FDP del error de posición para el algoritmo WPDOP para el caso de trayectoria bidimensional.

8.1.1.5 EPR

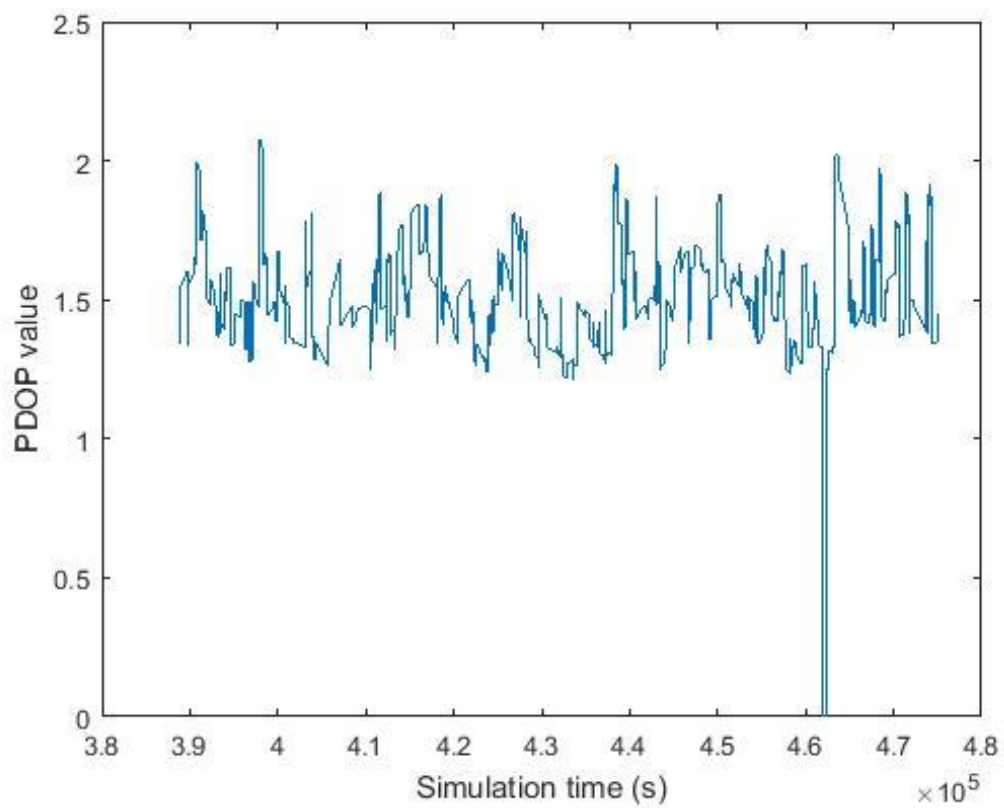


Figura 113: Valores del PDOP para el algoritmo EPR para el caso de trayectoria bidimensional.

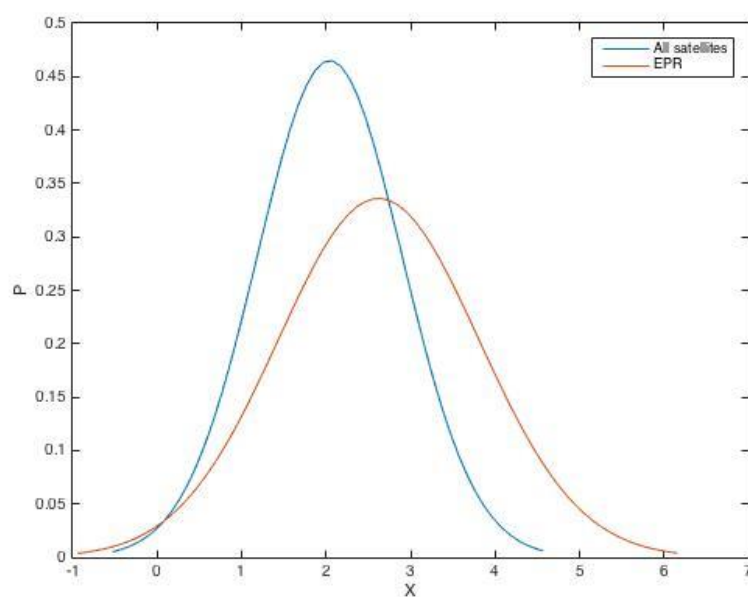


Figura 114: FDP del error de posición para el algoritmo EPR para el caso de trayectoria bidimensional.

8.1.2 Caso 3D

8.1.2.1 *Highest Elevation Algorithm (HEA)*

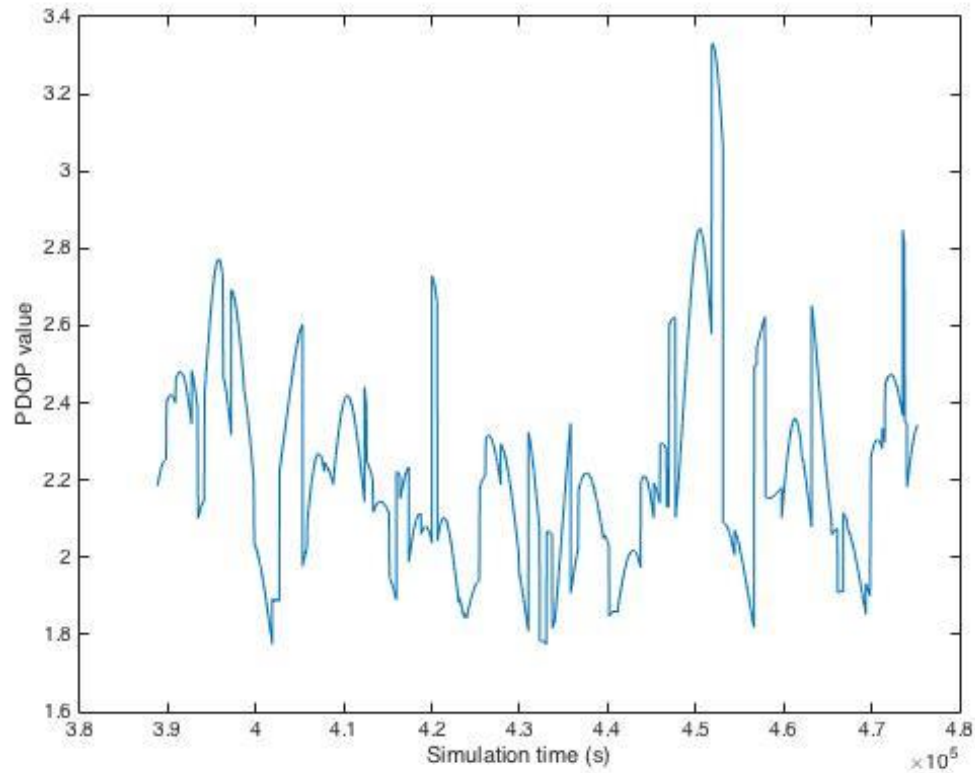


Figura 115: Valores del PDOP para el algoritmo HEA para el caso de trayectoria tridimensional.

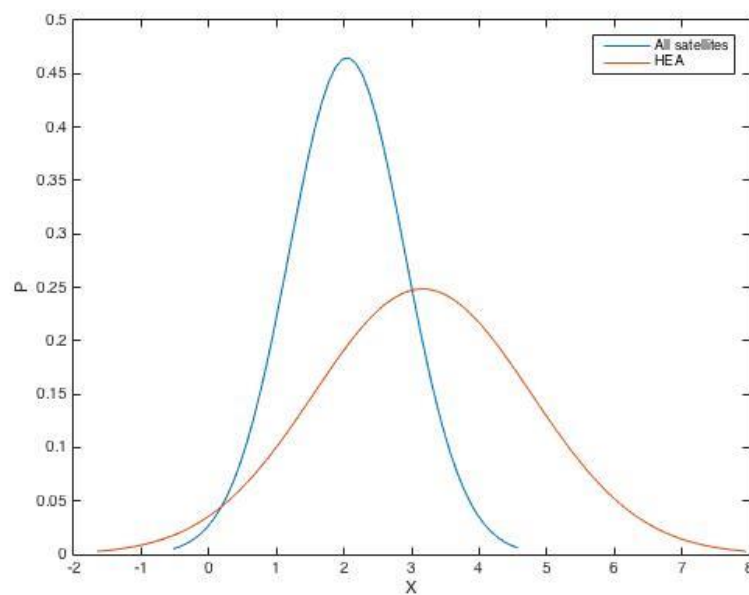


Figura 116: FDP del error de posición para el algoritmo HEA para el caso de trayectoria tridimensional.

8.1.2.2 Maximum Volume Algorithm (MVA)

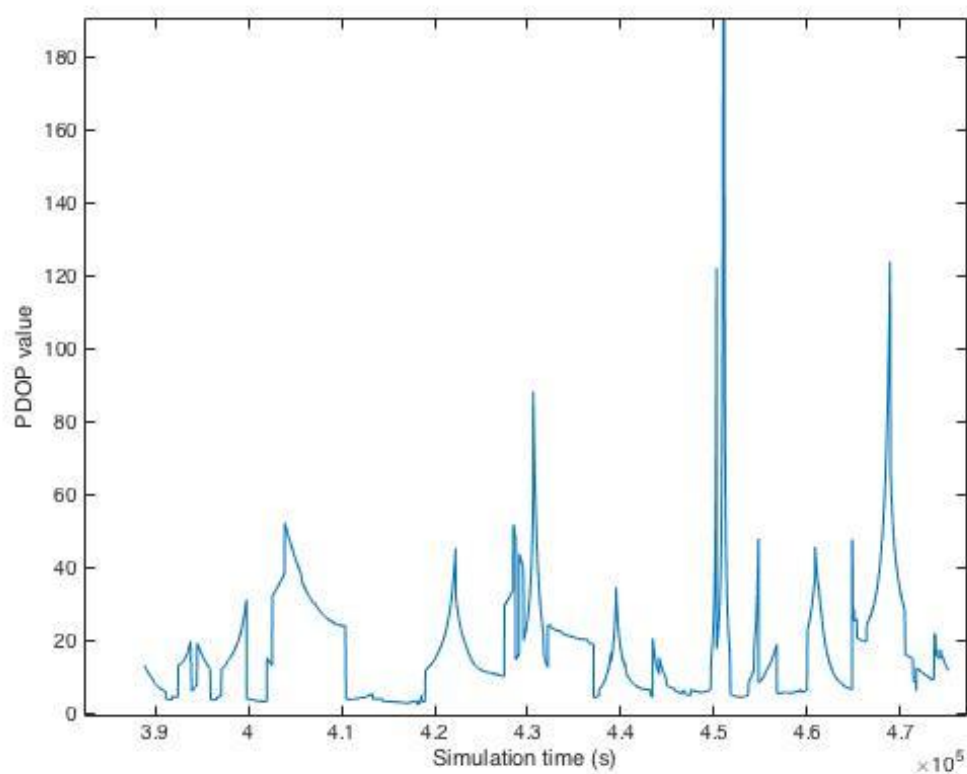


Figura 117: Valores del PDOP para el algoritmo MVA para el caso de trayectoria tridimensional.

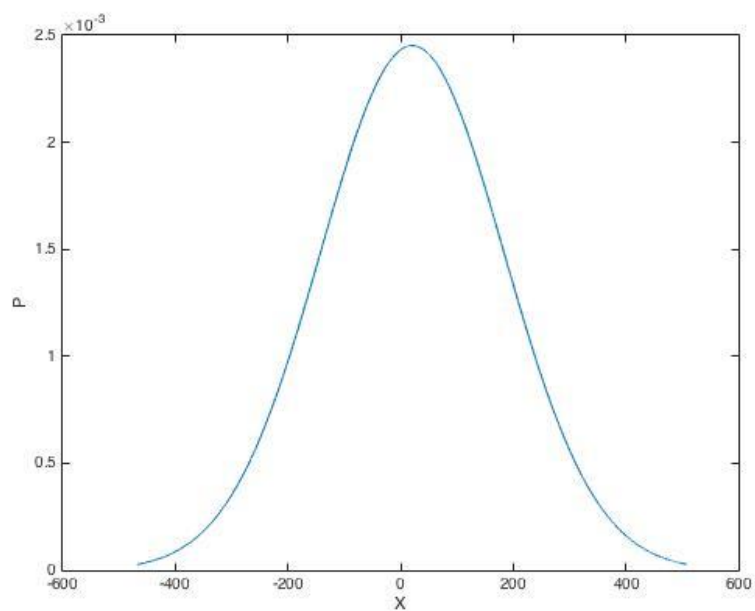


Figura 118: FDP del error de posición para el algoritmo MVA para el caso de trayectoria tridimensional.

8.1.2.3 Dartboard DOP algorithm

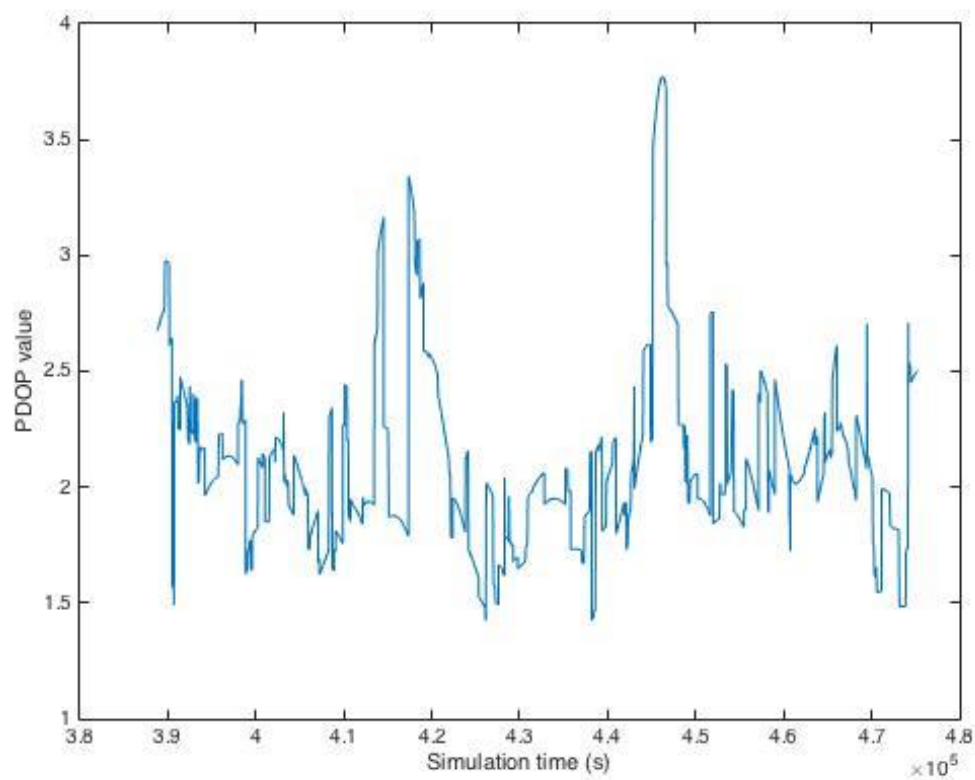


Figura 119: Valores del PDOP para el algoritmo Dartboard DOP para el caso de trayectoria tridimensional.

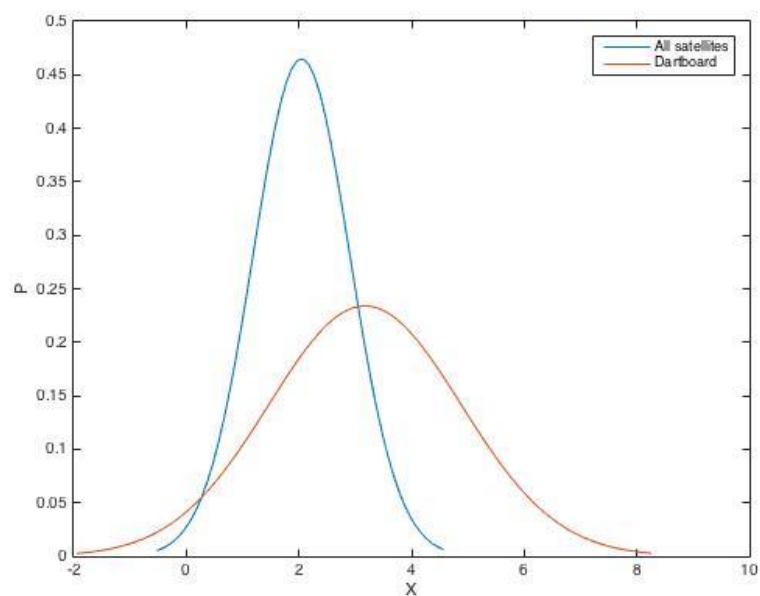


Figura 120: FDP del error de posición para el algoritmo Dartboard DOP para el caso de trayectoria tridimensional.

8.1.2.4 WPDOP algorithm

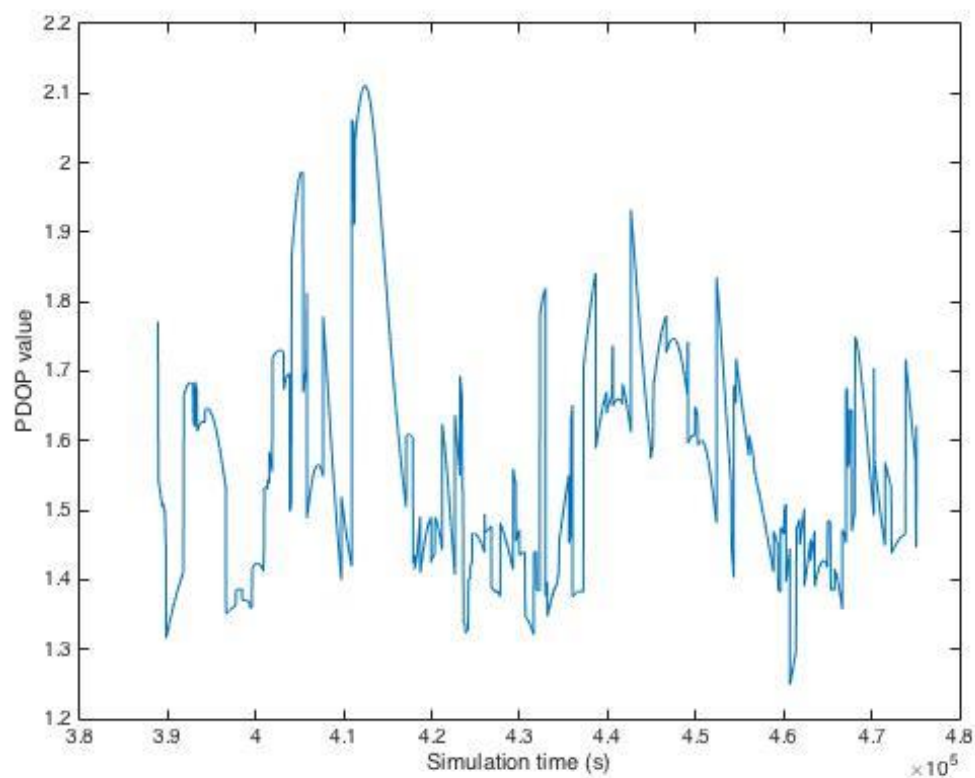


Figura 121: Valores del PDOP para el algoritmo WPDOP para el caso de trayectoria tridimensional.

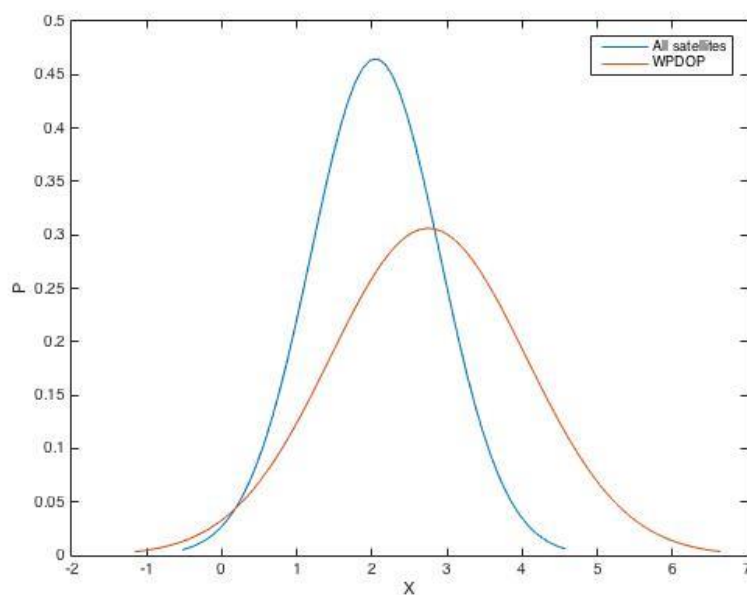


Figura 122: FDP del error de posición para el algoritmo WPDOP para el caso de trayectoria tridimensional.

8.1.2.5 *EPR algorithm*

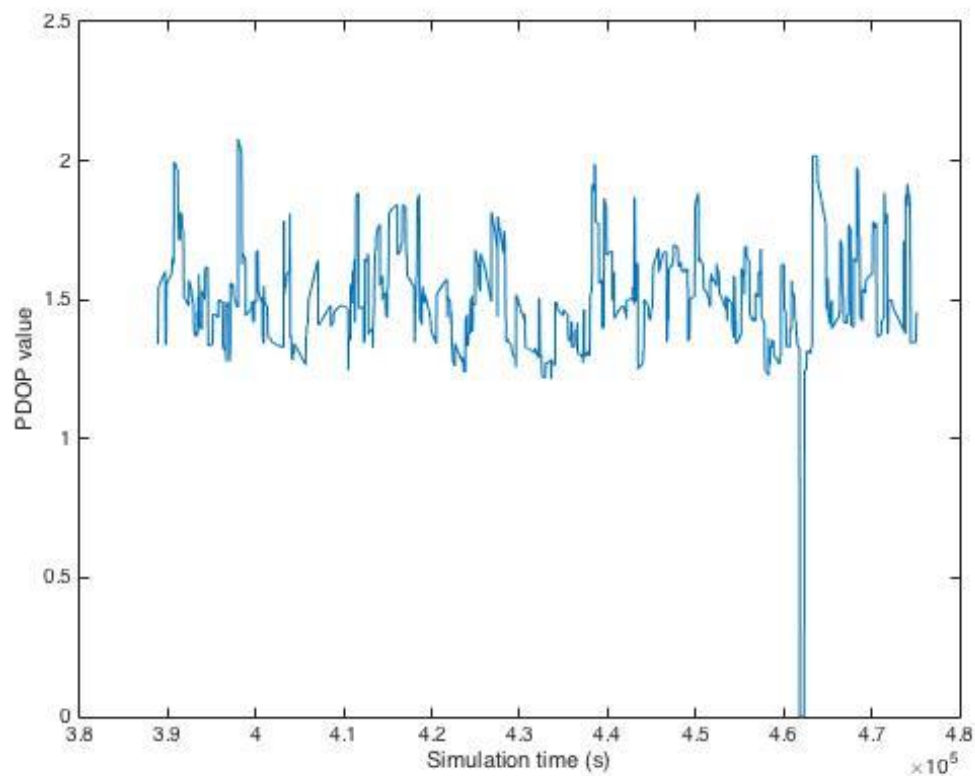


Figura 123: Valores del PDOP para el algoritmo EPR para el caso de trayectoria tridimensional.

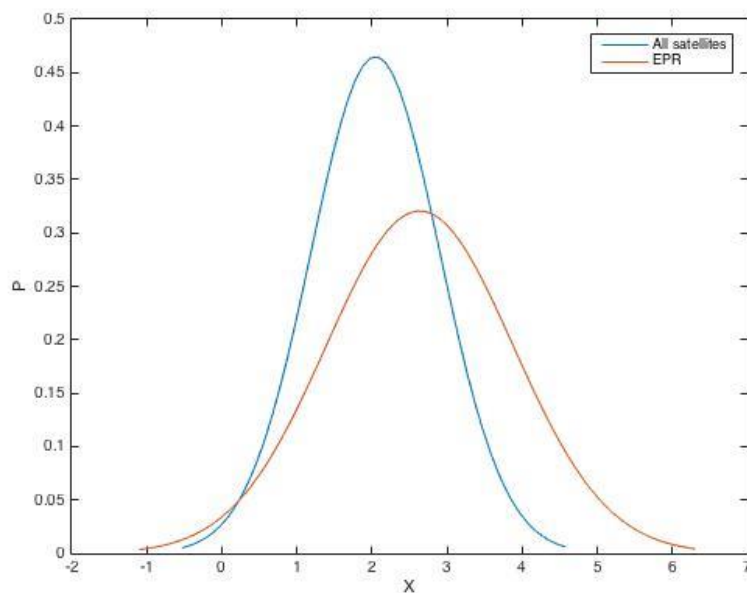


Figura 124; FDP del error de posición para el algoritmo EPR para el caso de trayectoria tridimensional.

8.2 Detección de medidas erróneas

En esta sección se van a presentar las figuras correspondientes a los casos de trayectoria bidimensional y tridimensional para los métodos de detección de medidas erróneas.

8.2.1 Caso 2D

8.2.1.1 *Un satélite con error constante en el pseudorange*

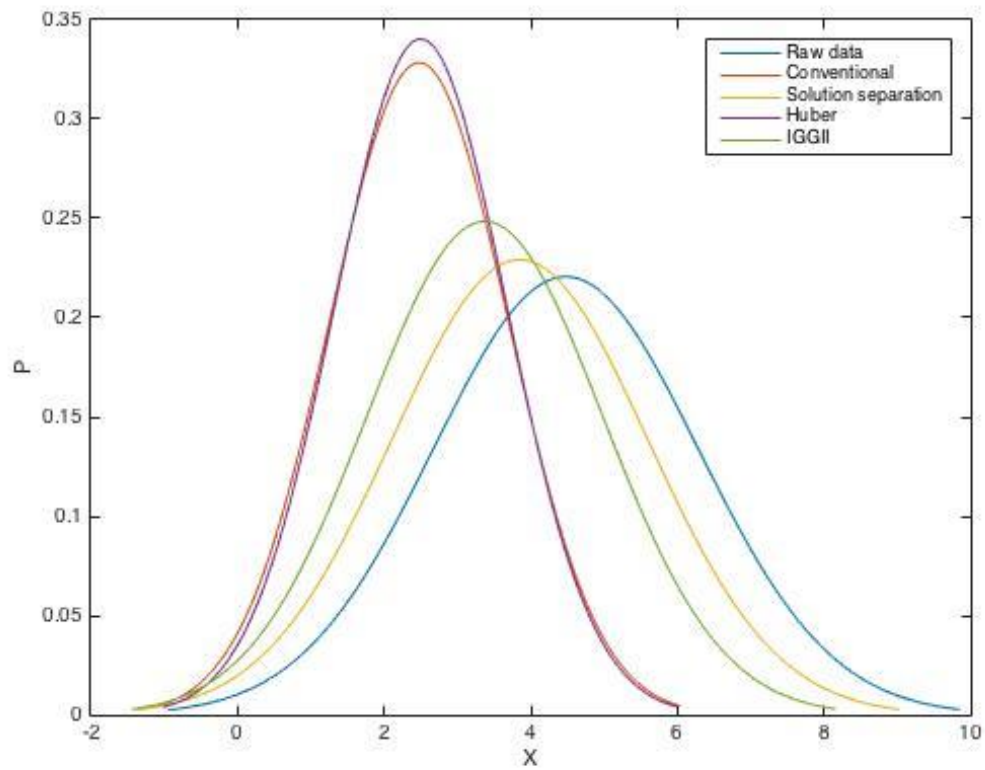


Figura 125: Funciones de densidad de probabilidad del error de posición con un satélite con error constante para el caso de trayectoria bidimensional.

8.2.1.1.1 RAIM convencional

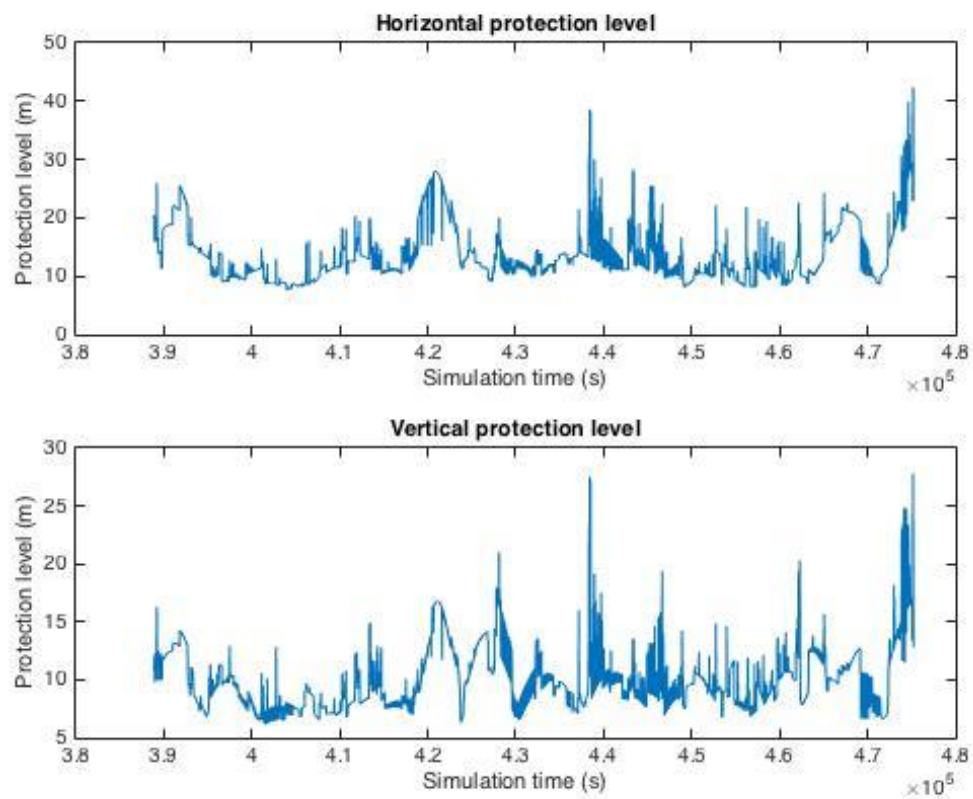


Figura 126: Niveles de protección para el RAIM convencional con un satélite con error constante para el caso de trayectoria bidimensional.

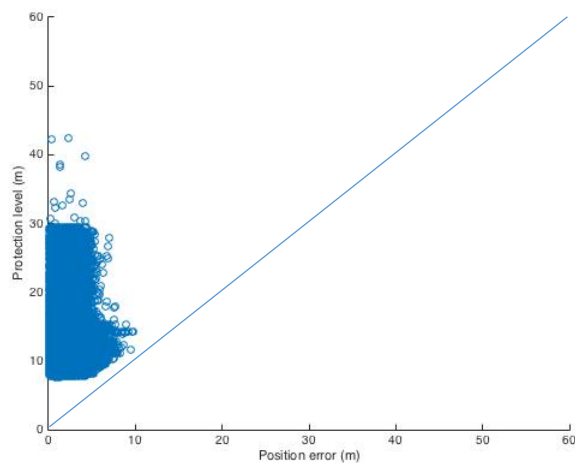


Figura 127: Diagrama de Stanford del componente horizontal para el RAIM convencional con un satélite con error constante para el caso de trayectoria bidimensional.

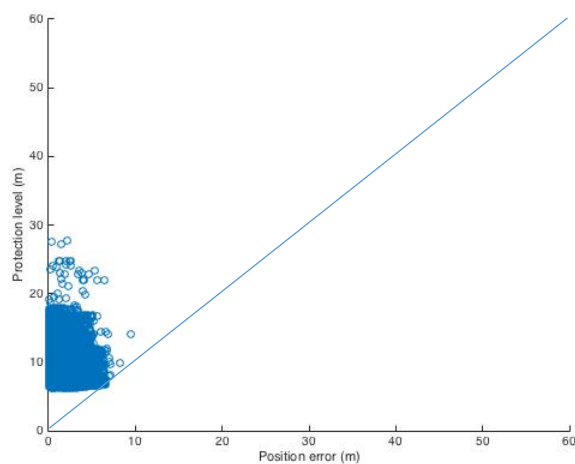


Figura 128: Diagrama de Stanford del componente horizontal para el RAIM convencional con un satélite con error constante para el caso de trayectoria bidimensional.

8.2.1.1.2 RAIM basado en separación de soluciones

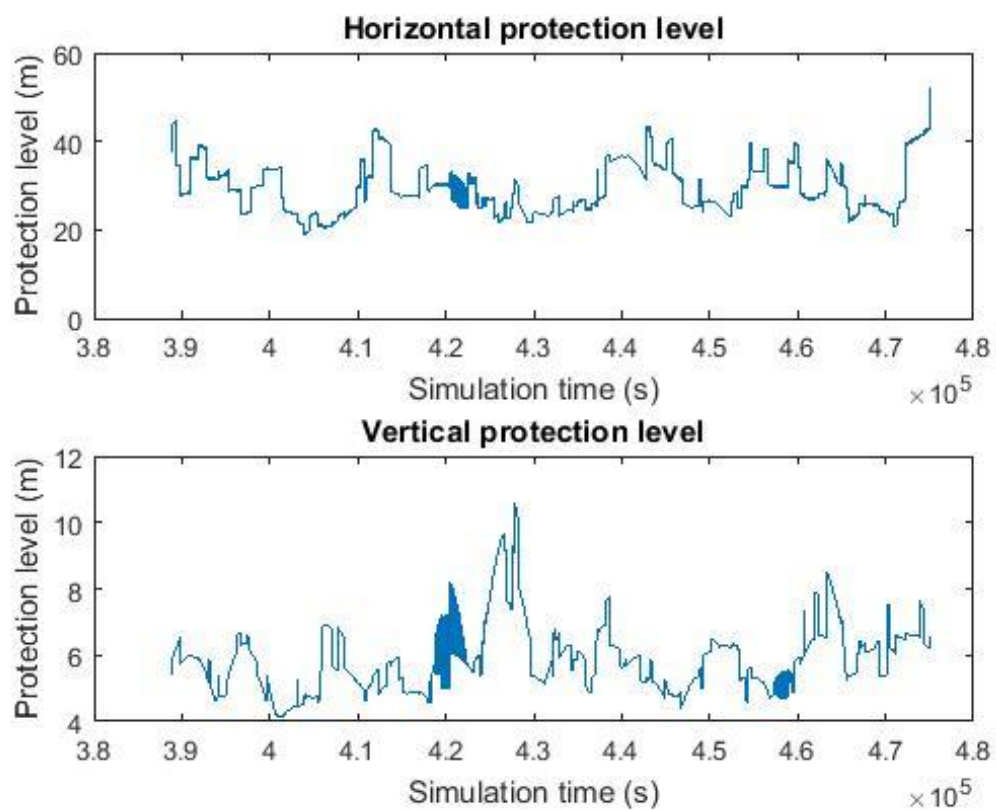


Figura 129: Niveles de protección del RAIM basado en separación de soluciones para el caso de trayectoria bidimensional con un satélite con error constante en el pseudorange.

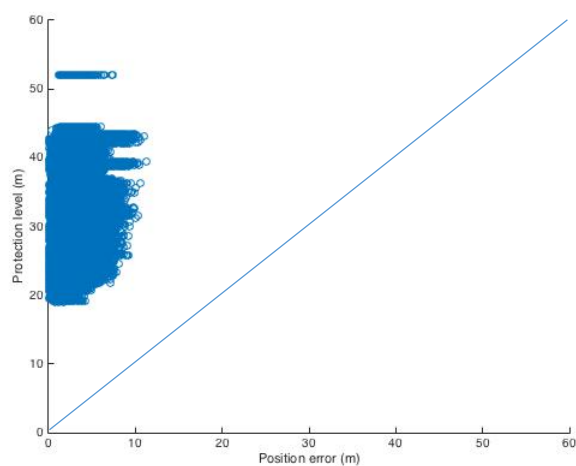
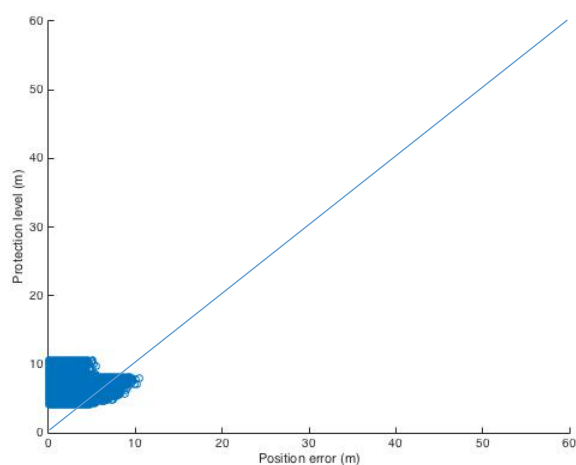


Figura 130: Diagrama de Stanford del componente horizontal para el RAIM basado en separación de soluciones con un satélite con error constante para el caso de trayectoria bidimensional.

Figura 131: Diagrama de Stanford del componente vertical para el RAIM basado en separación de



soluciones con un satélite con error constante para el caso de trayectoria bidimensional.

8.2.1.1.3 IBPL

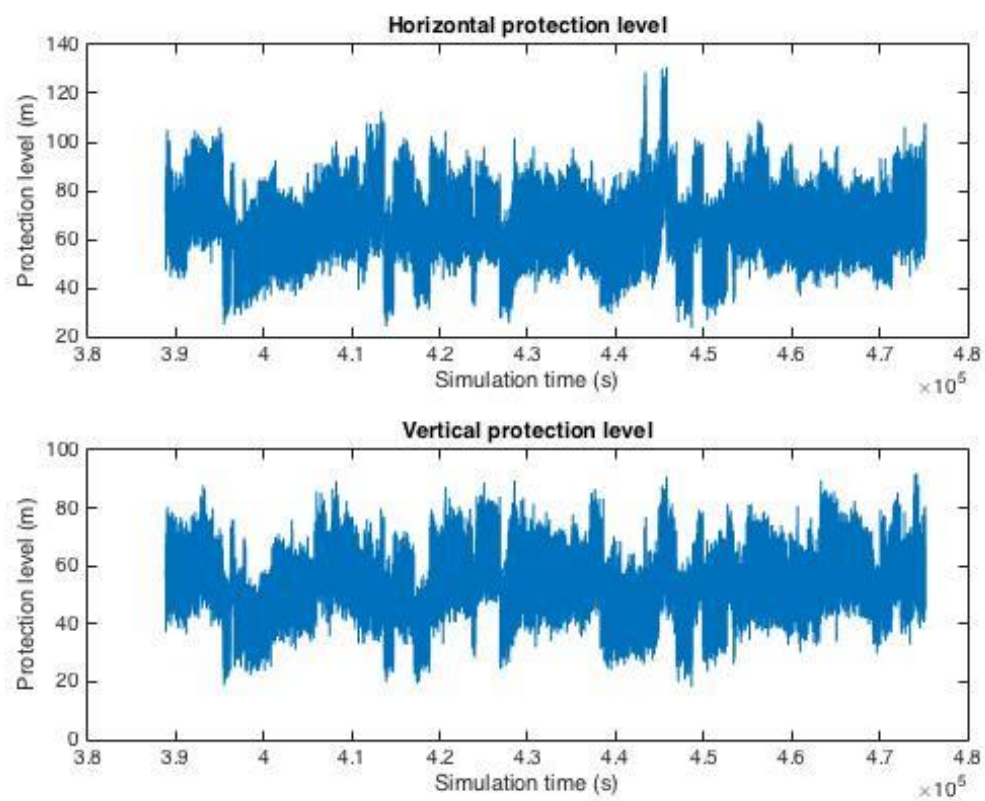


Figura 132: Niveles de protección del IBPL para el caso de trayectoria bidimensional con un satélite con error constante en el pseudorange.

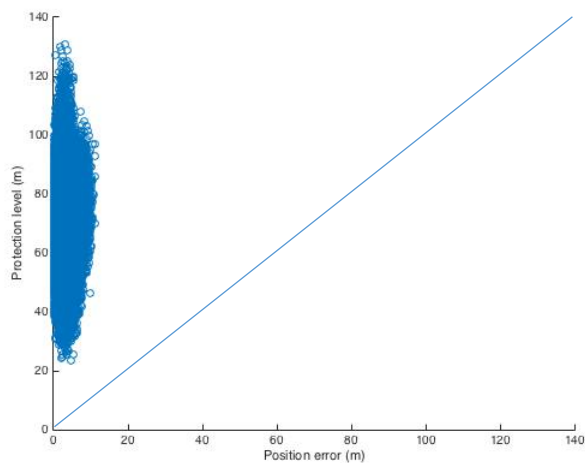
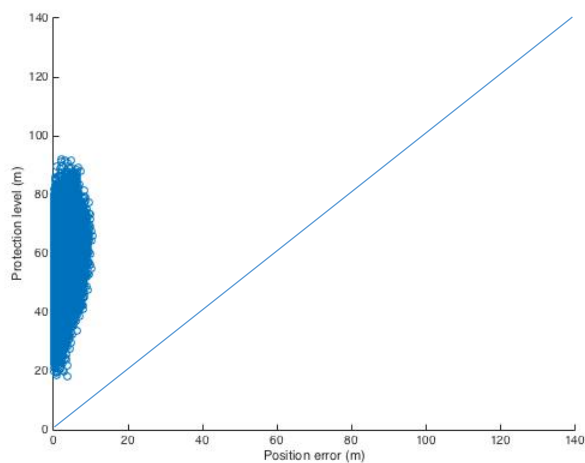


Figura 133: Diagrama de Stanford del componente horizontal para el IBPL con un satélite con error constante para el caso de trayectoria bidimensional.

Figura 134: Diagrama de Stanford del componente vertical para el IBPL con un satélite con error constante



para el caso de trayectoria bidimensional.

8.2.1.2 *Dos satélites con errores constantes en el pseudorango*

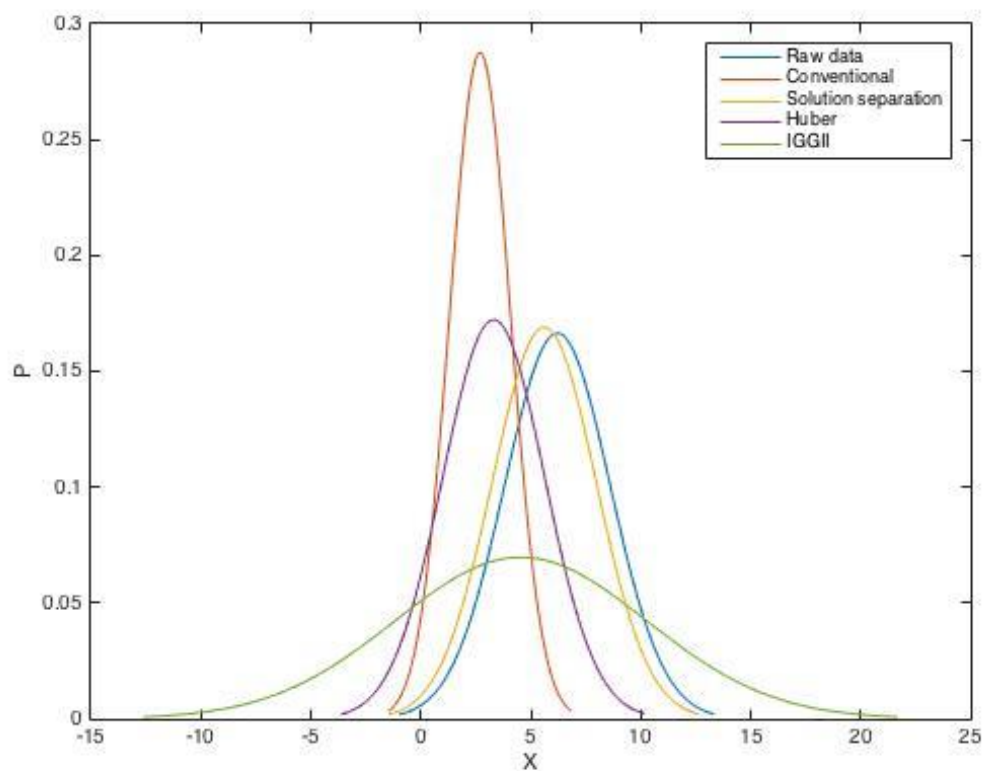


Figura 135: Funciones de densidad de probabilidad del error de posición con dos satélites con errores constantes para el caso de trayectoria bidimensional.

8.2.1.2.1 RAIM convencional

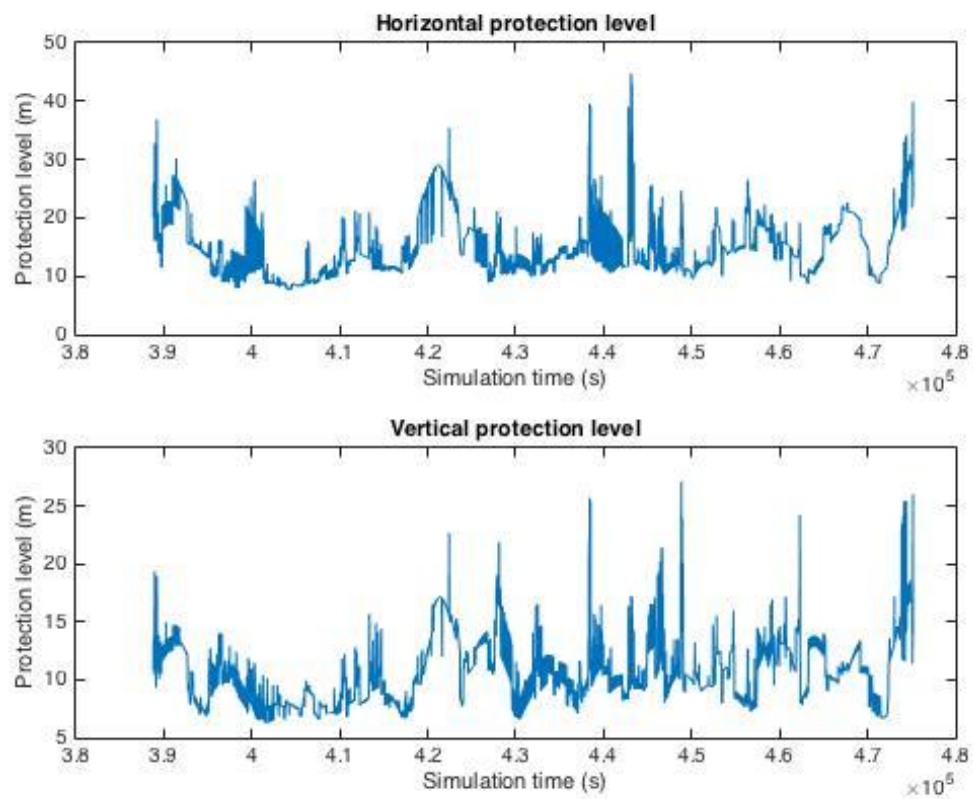


Figura 136: Niveles de protección para el RAIM convencional con dos satélites con errores constantes para el caso de trayectoria bidimensional.

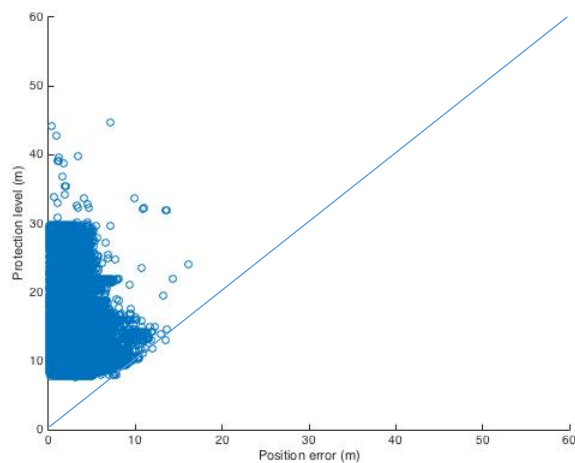


Figura 137: Diagrama de Stanford del componente horizontal para el RAIM convencional con dos satélites con errores constantes para el caso de trayectoria bidimensional.

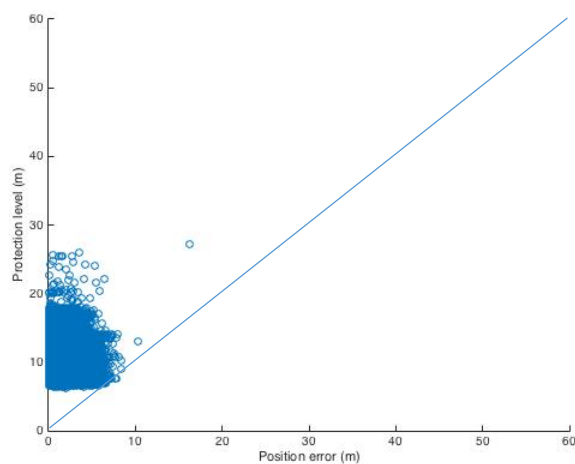


Figura 138: Diagrama de Stanford del componente horizontal para el RAIM convencional con dos satélites con errores constantes para el caso de trayectoria bidimensional.

8.2.1.2.2 RAIM basado en separación de soluciones

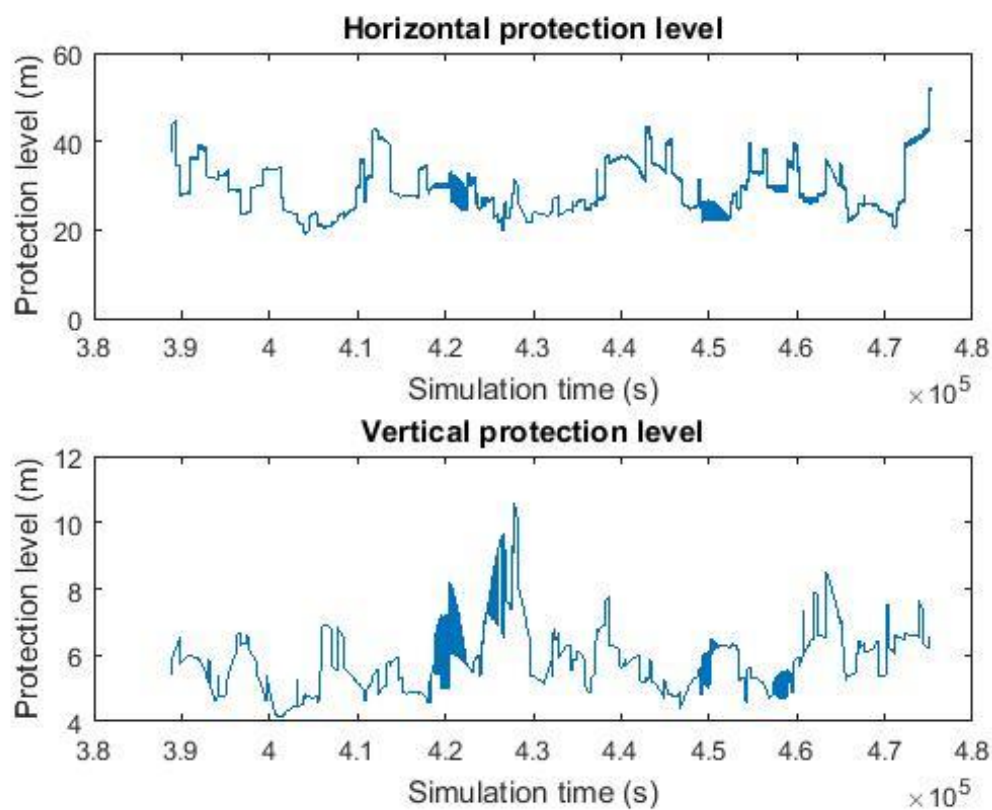


Figura 139: Niveles de protección del RAIM basado en separación de soluciones para el caso de trayectoria bidimensional con dos satélites con errores constantes en el pseudorange.

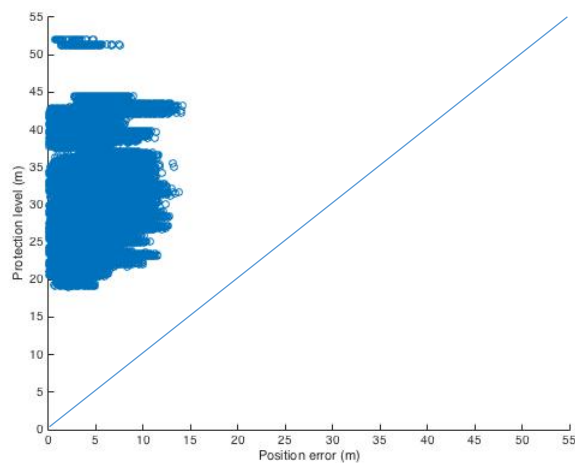


Figura 140: Diagrama de Stanford del componente horizontal para el RAIM basado en separación de soluciones con dos satélites con errores constantes para el caso de trayectoria bidimensional.

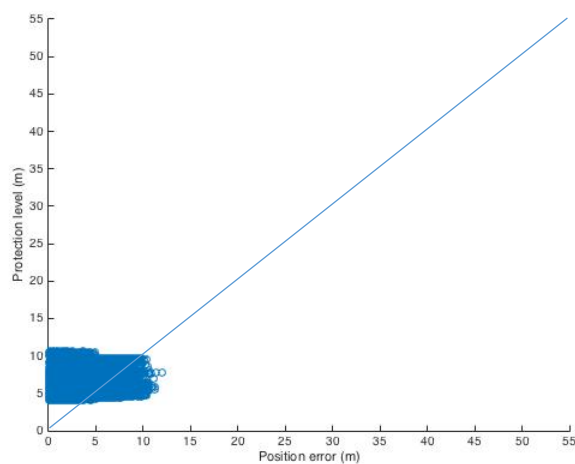
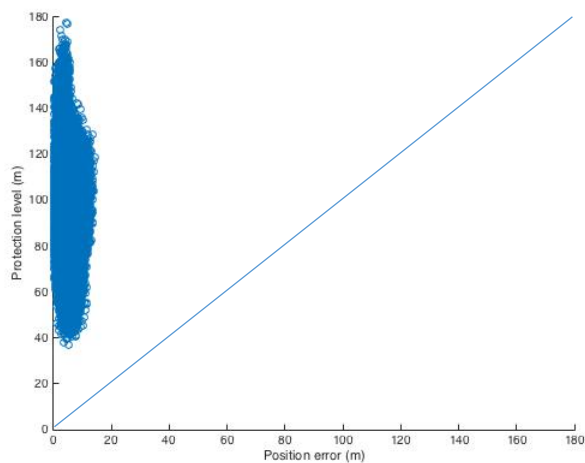


Figura 141: Diagrama de Stanford del componente vertical para el RAIM basado en separación de soluciones con dos satélites con errores constantes para el caso de trayectoria bidimensional.



8.2.1.2.3 IBPL

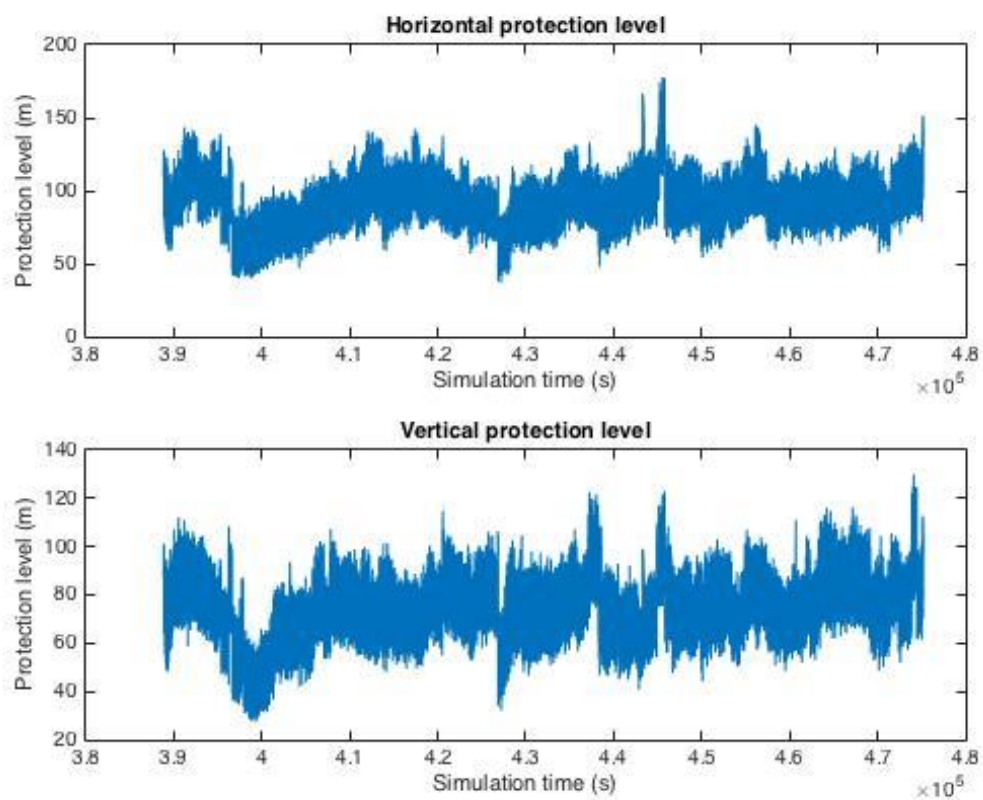
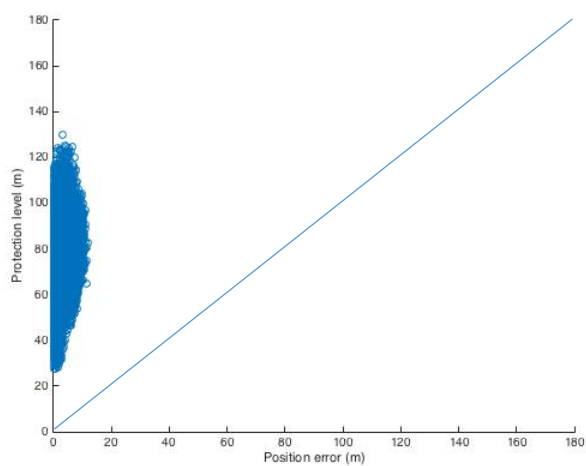


Figura 142: Niveles de protección del IBPL para el caso de trayectoria bidimensional con un satélite con error constante en el pseudorango.

Figura 143: Diagrama de Stanford del componente horizontal para el IBPL con dos satélites con errores constantes para el caso de trayectoria bidimensional.

Figura 144: Diagrama de Stanford del componente vertical para el IBPL con dos satélites con errores constantes para el caso de trayectoria bidimensional.



8.2.1.3 Tres satélites con errores constantes en el pseudorango

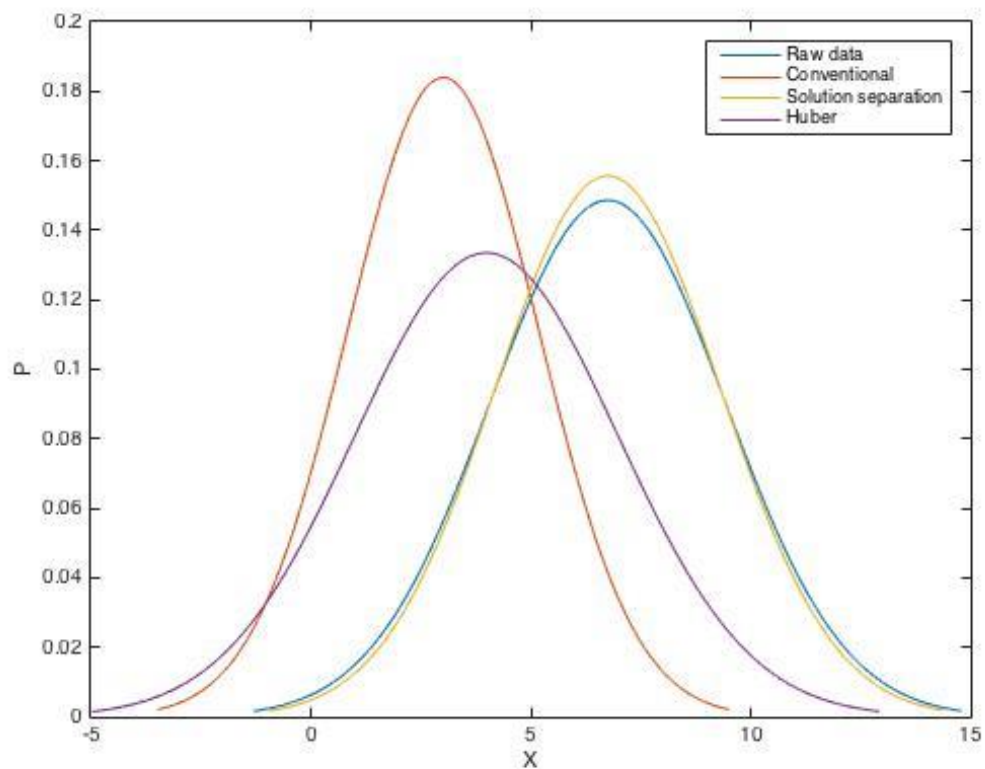


Figura 145: Funciones de densidad de probabilidad del error de posición con tres satélites con errores constantes para el caso de trayectoria bidimensional.

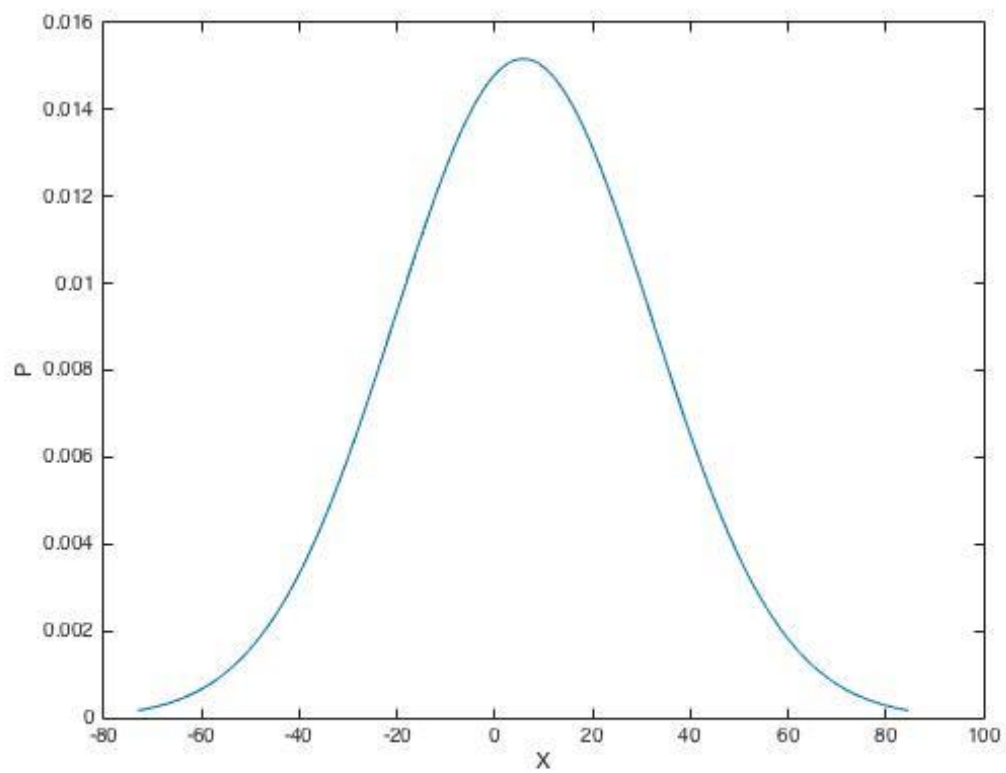


Figura 146: Función de densidad de probabilidad del error de posición para el estimador IGGII con tres satélites con errores constantes para el caso de trayectoria bidimensional.

8.2.1.3.1 RAIM convencional

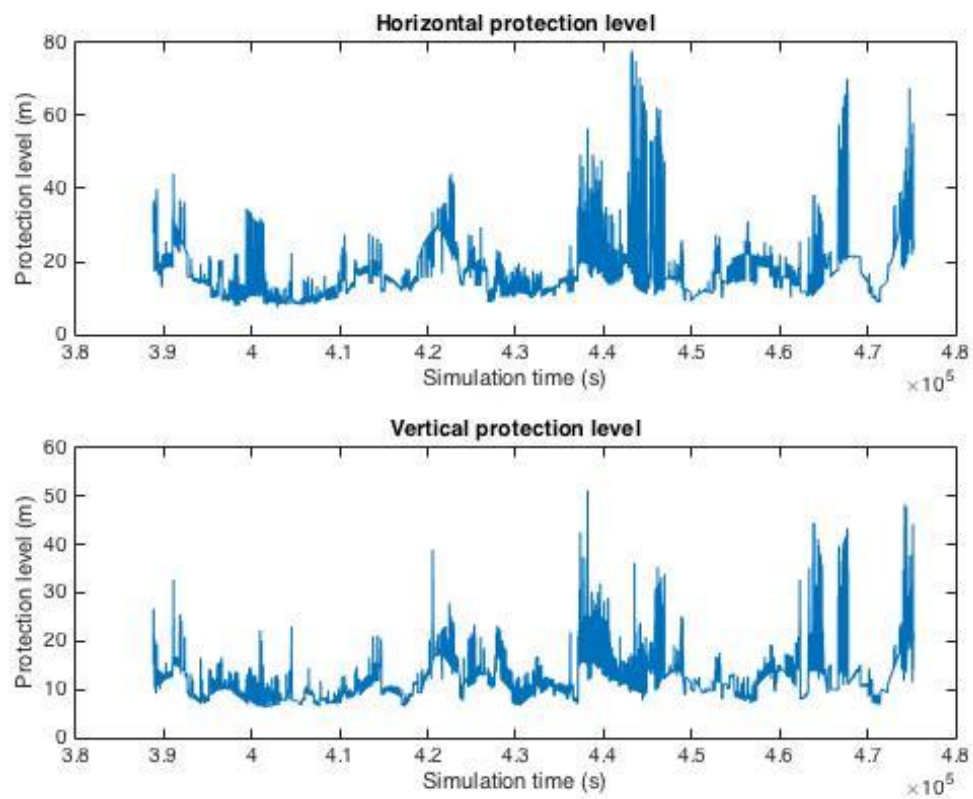


Figura 147: Niveles de protección para el RAIM convencional con tres satélites con errores constantes para el caso de trayectoria bidimensional.

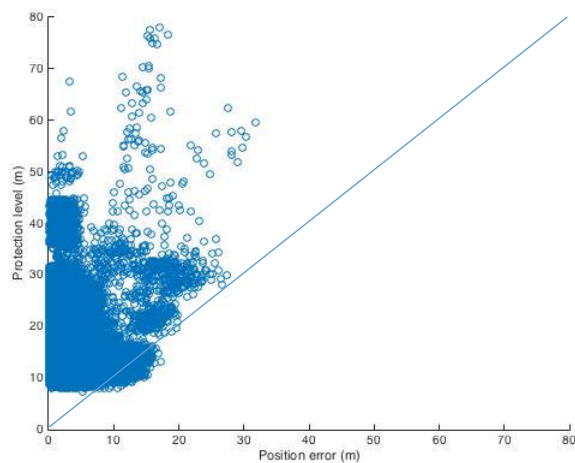
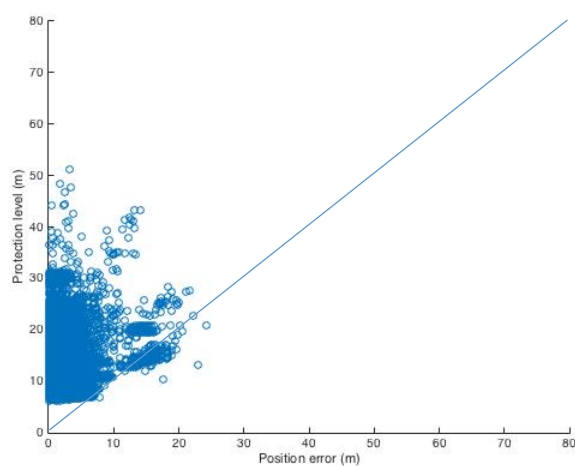


Figura 148: Diagrama de Stanford del componente horizontal para el RAIM convencional con tres satélites con errores constantes para el caso de trayectoria bidimensional.

Figura 149: Diagrama de Stanford del componente vertical para el RAIM convencional con tres satélites con



errores constantes para el caso de trayectoria bidimensional.

8.2.1.3.2 RAIM basado en separación de soluciones

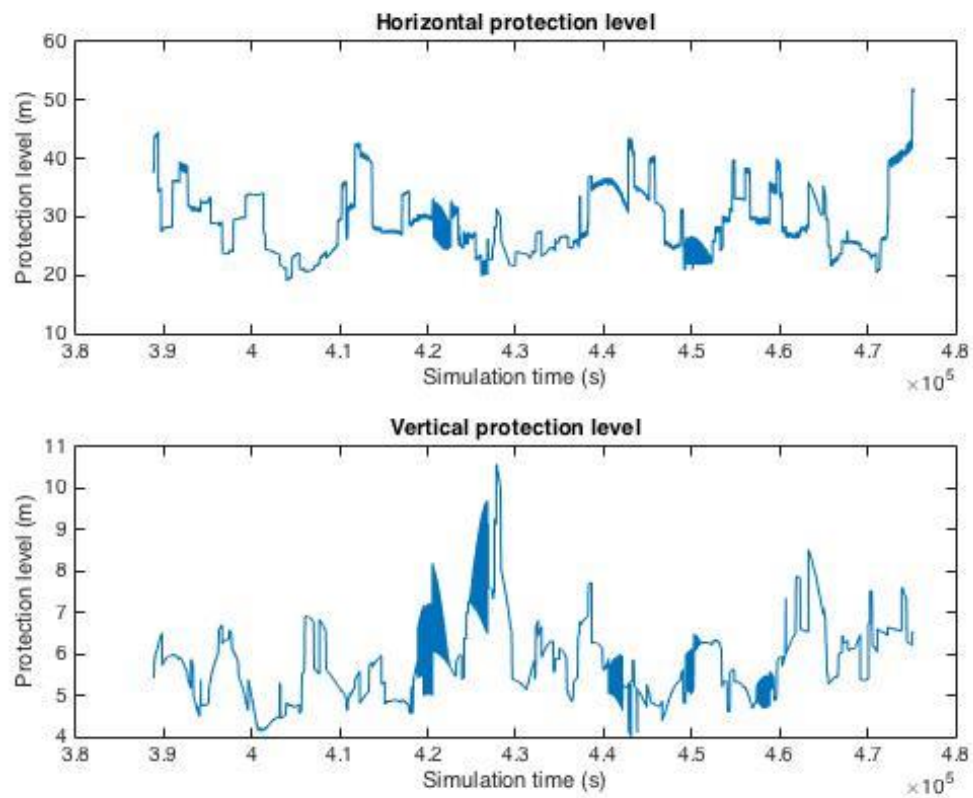


Figura 150: Niveles de protección para el RAIM basado en separación de soluciones con tres satélites con errores constantes para el caso de trayectoria bidimensional.

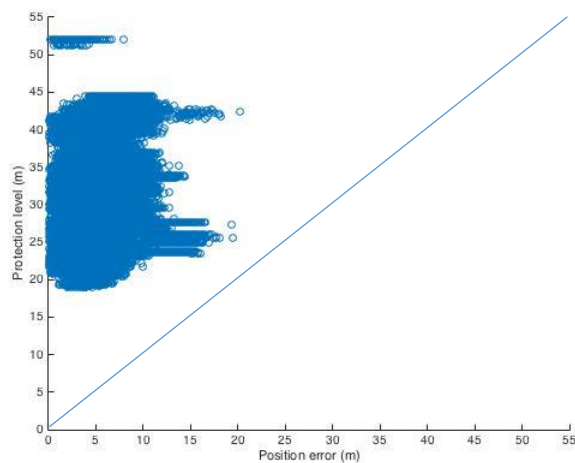
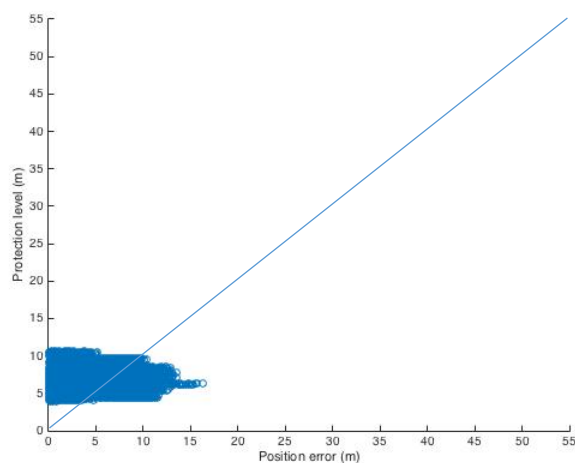


Figura 151: Diagrama de Stanford del componente horizontal para el RAIM basado en separación de soluciones con tres satélites con errores constantes para el caso de trayectoria bidimensional.

Figura 152: Diagrama de Stanford del componente vertical para el RAIM basado en separación de



soluciones con tres satélites con errores constantes para el caso de trayectoria bidimensional.

8.2.1.3.3 IBPL

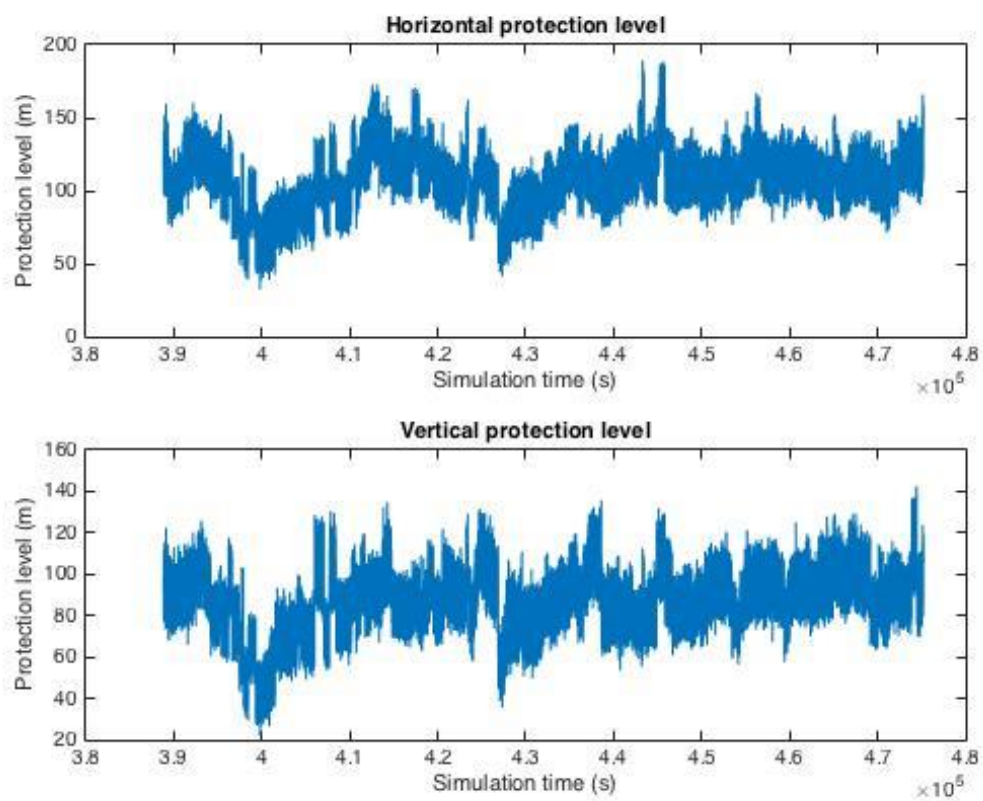


Figura 153: Niveles de protección para el IBPL con tres satélites con errores constantes para el caso de trayectoria bidimensional.

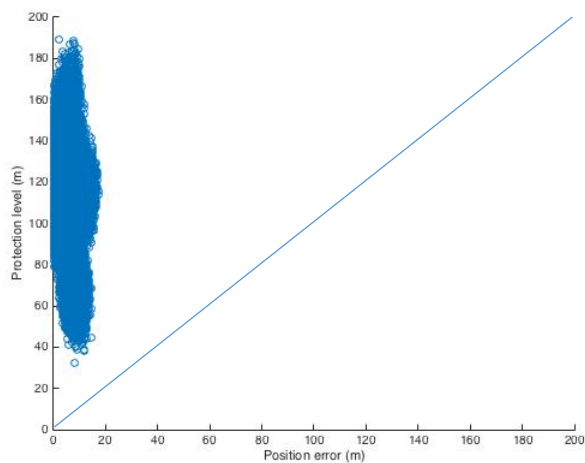
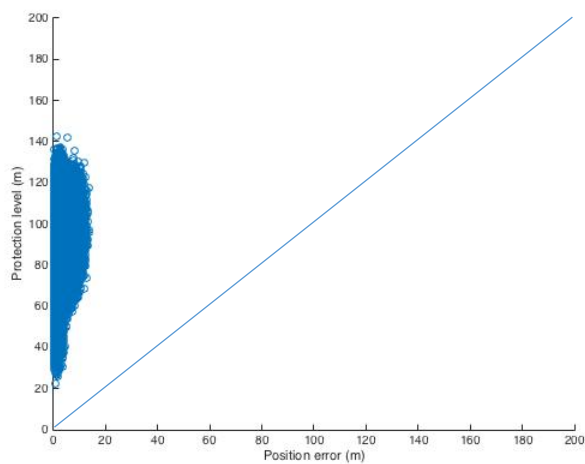


Figura 154: Diagrama de Stanford del componente horizontal para el IBPL con tres satélites con errores constantes para el caso de trayectoria bidimensional.

Figura 155: Diagrama de Stanford del componente vertical para el IBPL con tres satélites con errores



constantes para el caso de trayectoria bidimensional.

8.2.2 Caso 3D

8.2.2.1 *Un satélite con error constante en el pseudorange*

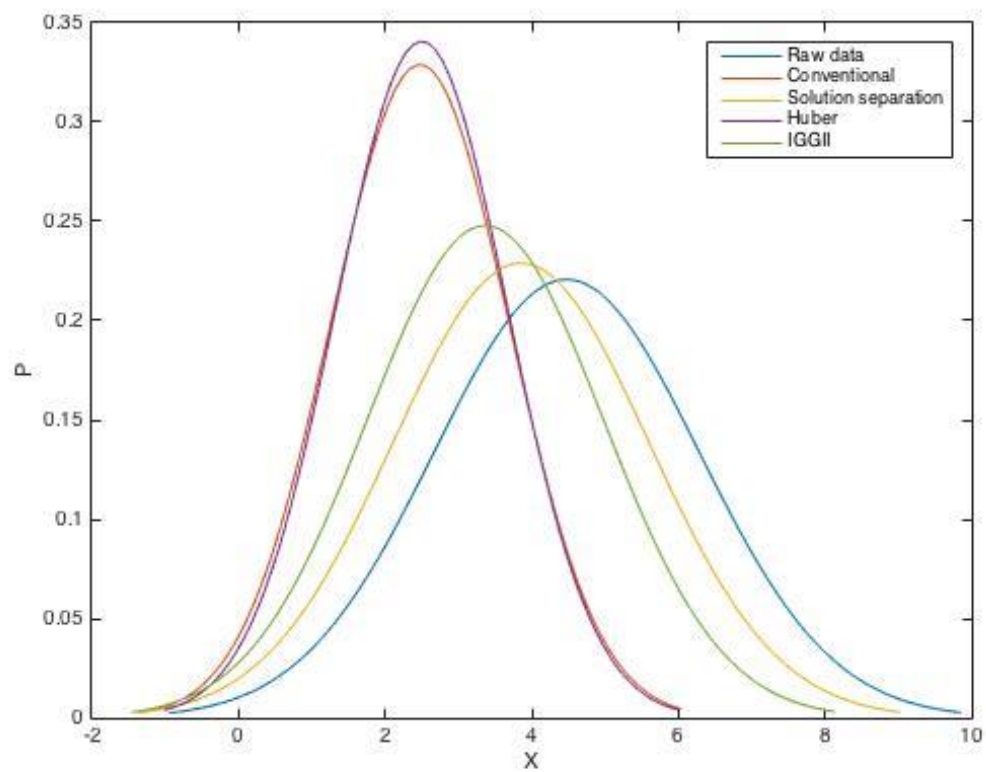


Figura 156: Funciones de densidad de probabilidad del error de posición con un satélite con error constante para el caso de trayectoria bidimensional.

8.2.2.1.1 RAIM convencional

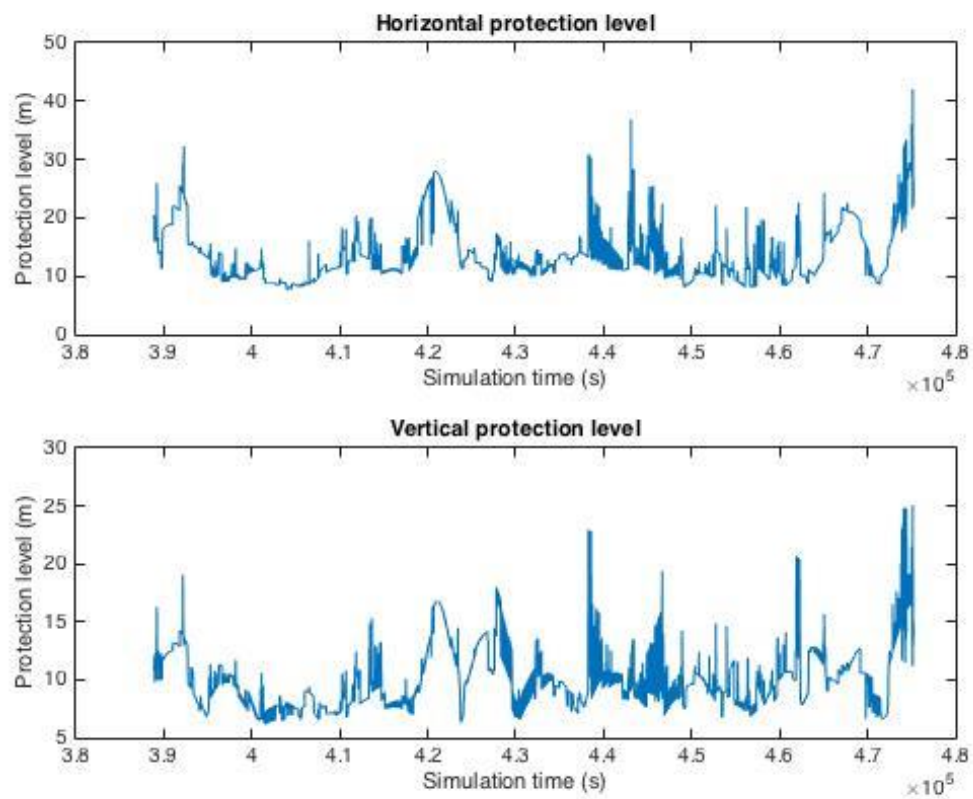


Figura 157: Niveles de protección para el RAIM convencional con un satélite con error constante para el caso de trayectoria bidimensional.

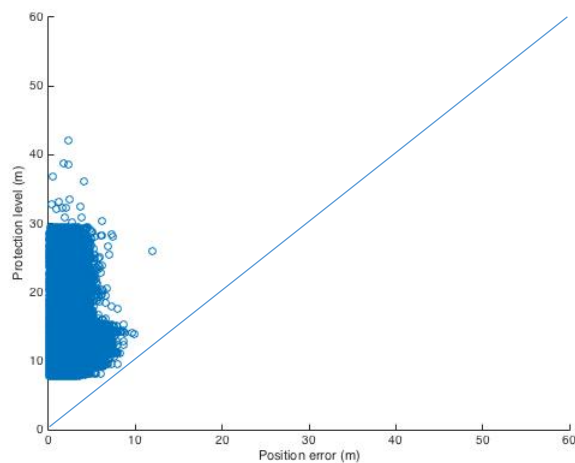
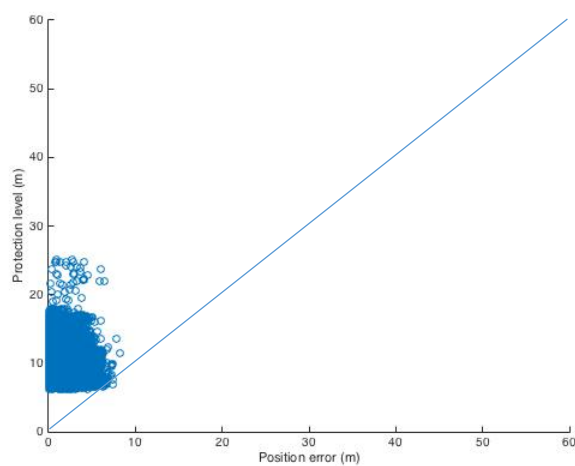


Figura 158: Diagrama de Stanford del componente horizontal para el RAIM convencional con un satélite con error constante para el caso de trayectoria bidimensional.

Figura 159: Diagrama de Stanford del componente horizontal para el RAIM convencional con un satélite



con error constante para el caso de trayectoria bidimensional.

8.2.2.1.2 RAIM basado en separación de soluciones

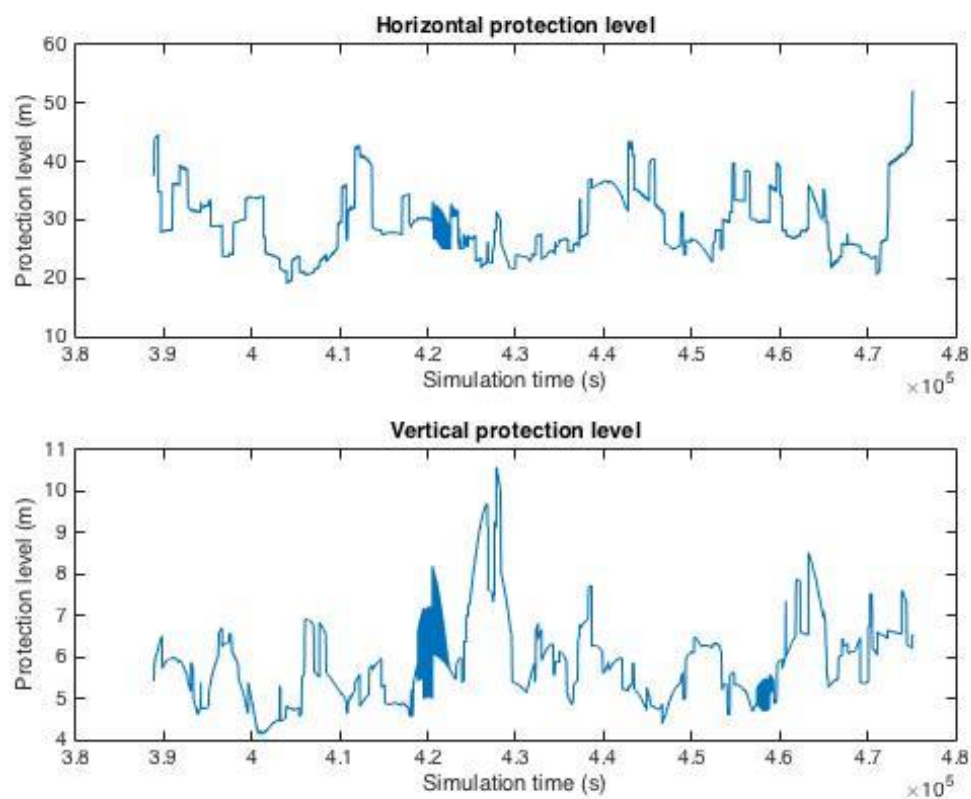


Figura 160: Niveles de protección del RAIM basado en separación de soluciones para el caso de trayectoria bidimensional con un satélite con error constante en el pseudorange.

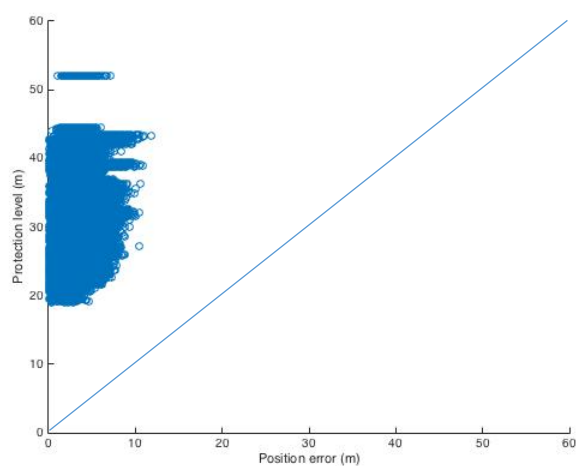


Figura 161: Diagrama de Stanford del componente horizontal para el RAIM basado en separación de soluciones con un satélite con error constante para el caso de trayectoria bidimensional.

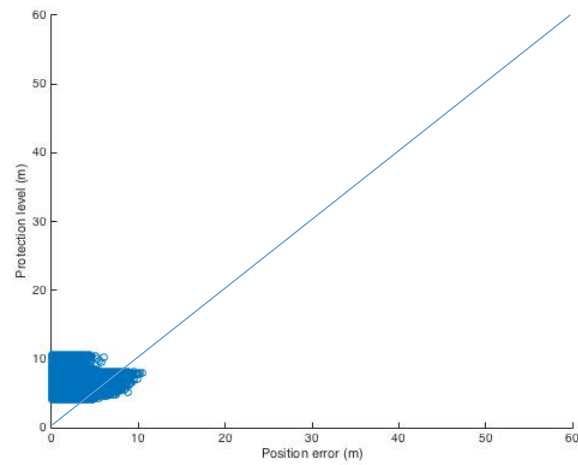


Figura 162: Diagrama de Stanford del componente vertical para el RAIM basado en separación de soluciones con un satélite con error constante para el caso de trayectoria bidimensional.

8.2.2.1.3 IBPL

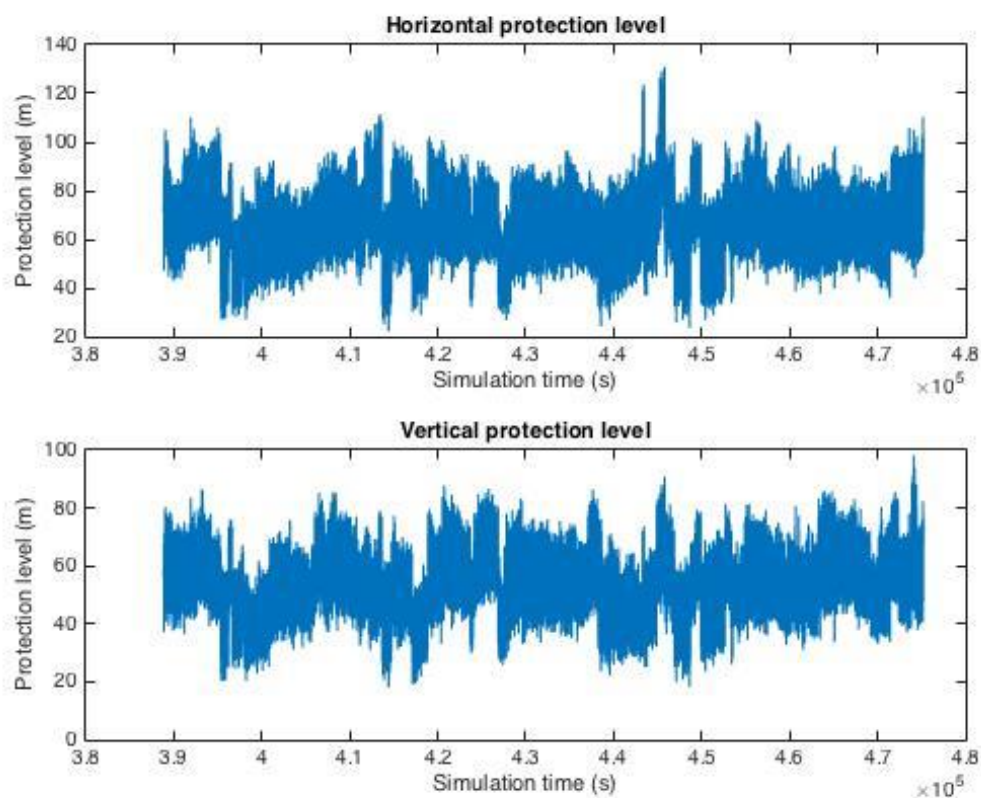


Figura 163: Niveles de protección del IBPL para el caso de trayectoria bidimensional con un satélite con error constante en el pseudorange.

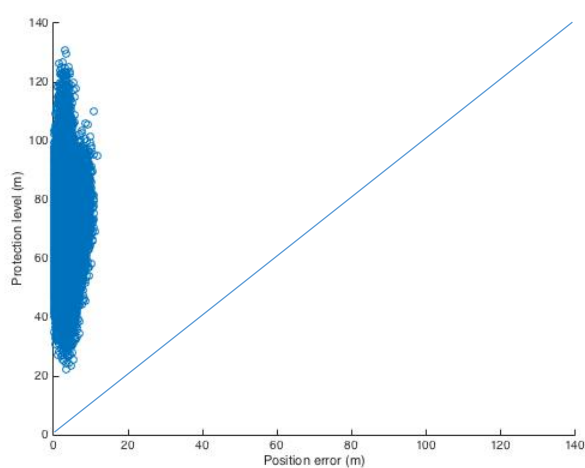


Figura 164: Diagrama de Stanford del componente horizontal para el IBPL con un satélite con error constante para el caso de trayectoria bidimensional.

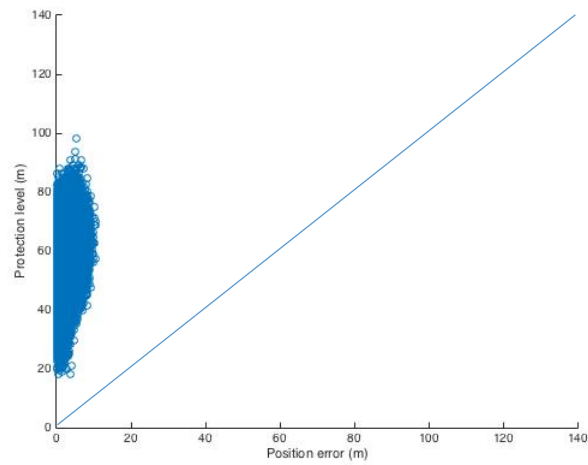


Figura 165: Diagrama de Stanford del componente vertical para el IBPL con un satélite con error constante para el caso de trayectoria bidimensional.

8.2.2.2 *Dos satélites con errores constantes en el pseudorango*

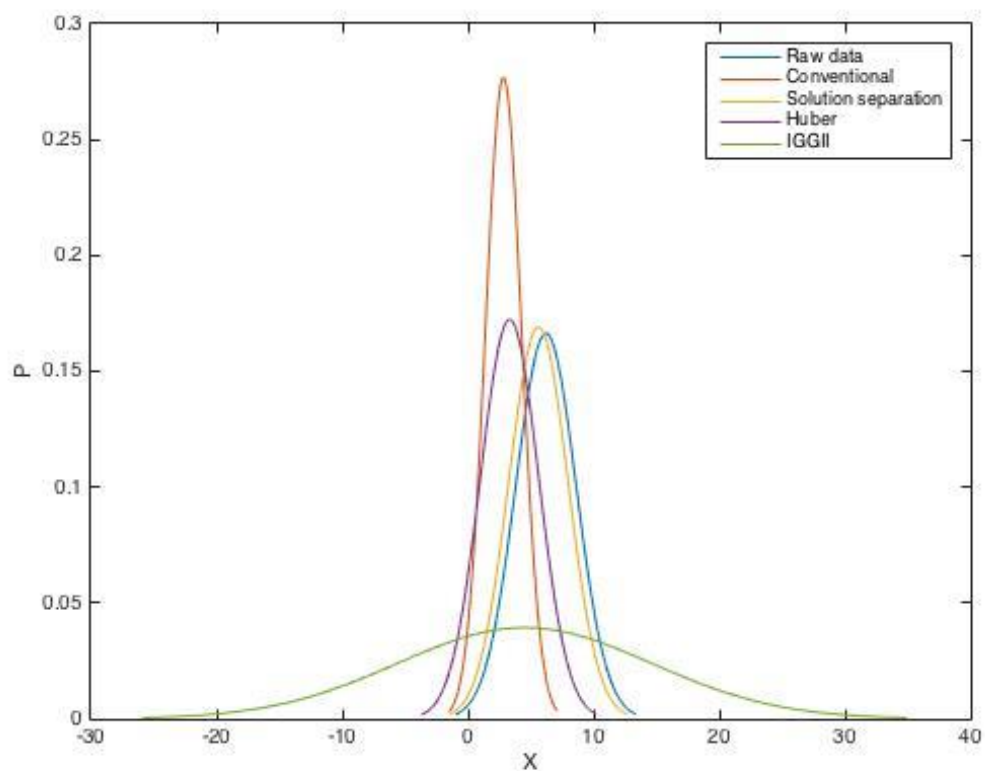


Figura 166: Funciones de densidad de probabilidad del error de posición con dos satélites con errores constantes para el caso de trayectoria bidimensional.

8.2.2.2.1 RAIM convencional

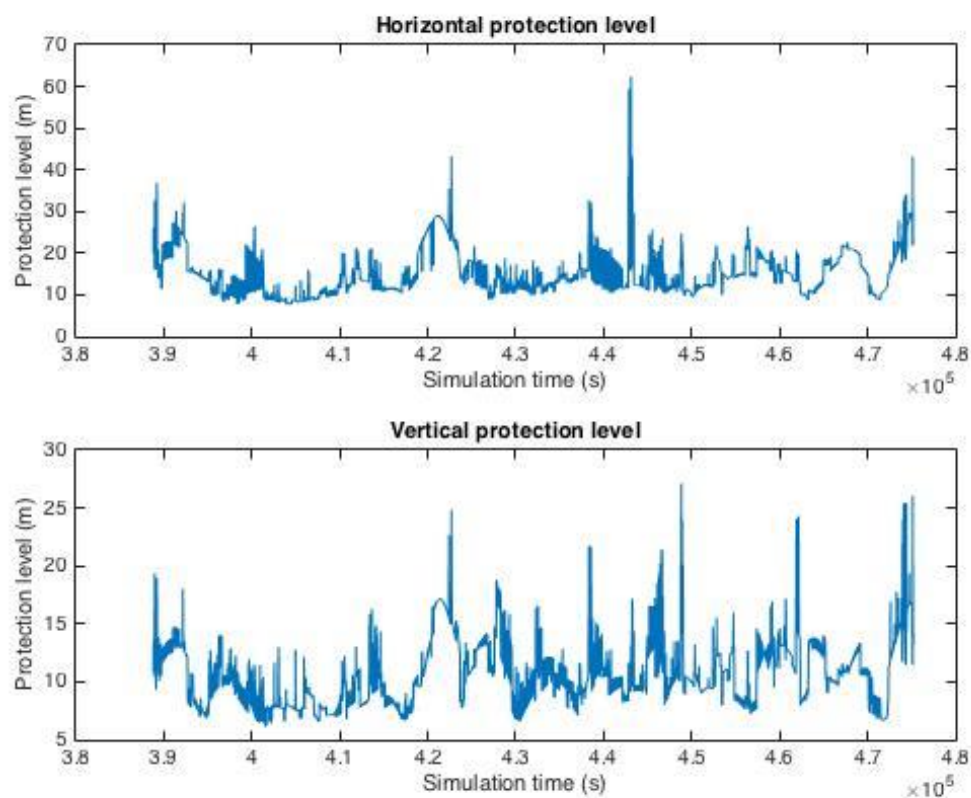


Figura 167: Niveles de protección para el RAIM convencional con dos satélites con errores constantes para el caso de trayectoria bidimensional.

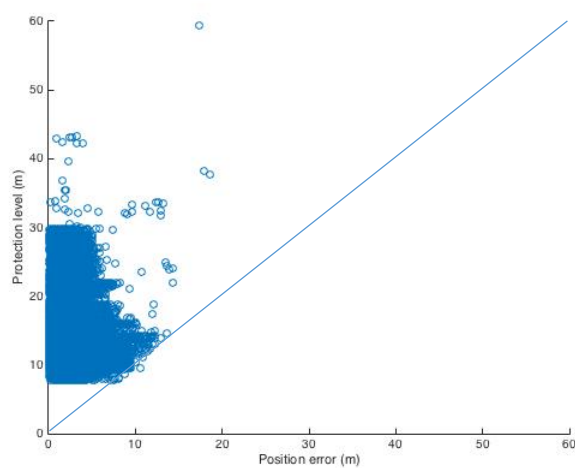


Figura 168: Diagrama de Stanford del componente horizontal para el RAIM convencional con dos satélites con errores constantes para el caso de trayectoria bidimensional.

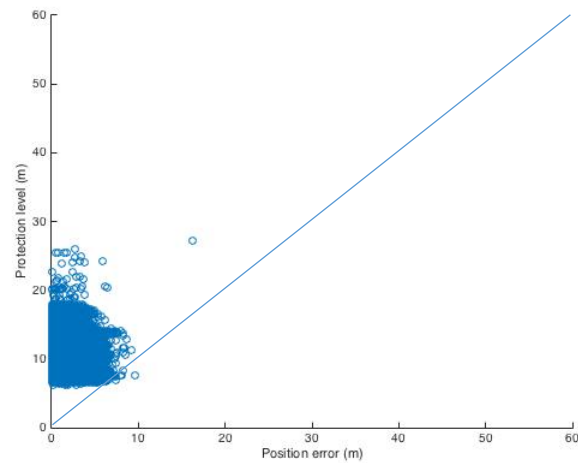


Figura 169: Diagrama de Stanford del componente horizontal para el RAIM convencional con dos satélites con errores constantes para el caso de trayectoria bidimensional.

8.2.2.2.2 RAIM basado en separación de soluciones

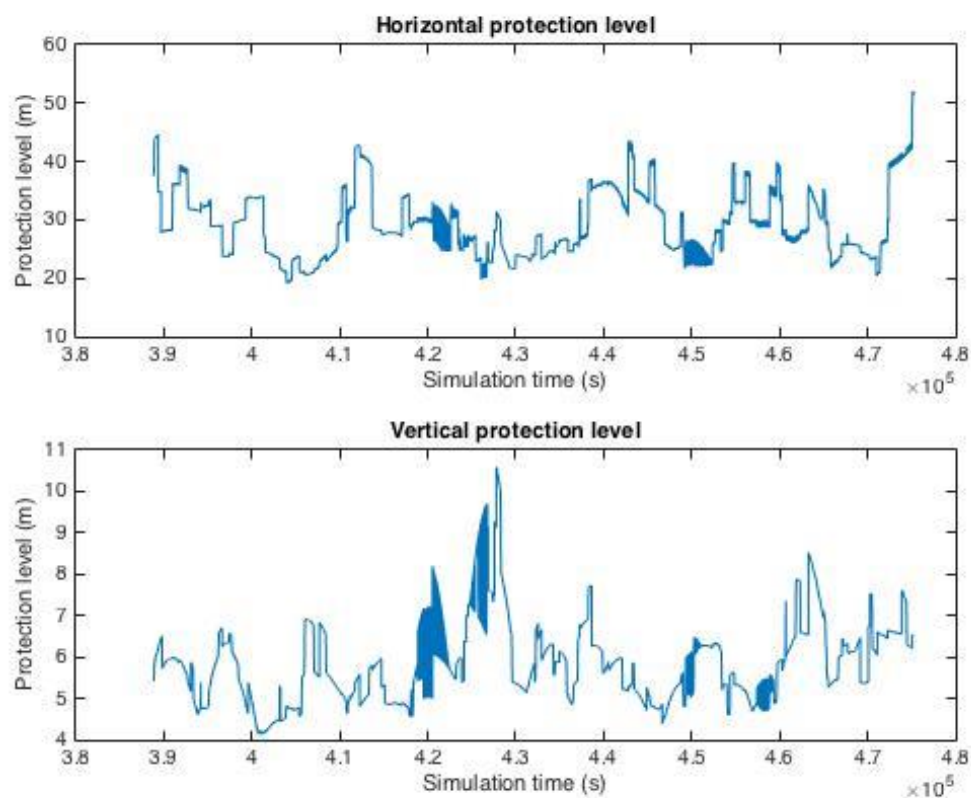


Figura 170: Niveles de protección del RAIM basado en separación de soluciones para el caso de trayectoria bidimensional con dos satélites con errores constantes en el pseudorange.

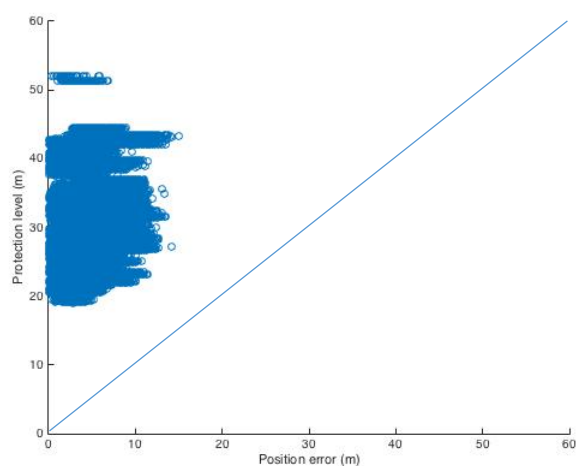


Figura 171: Diagrama de Stanford del componente horizontal para el RAIM basado en separación de soluciones con dos satélites con errores constantes para el caso de trayectoria bidimensional.

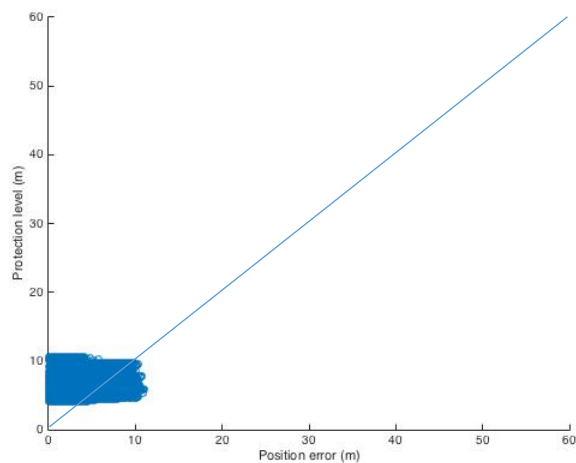


Figura 172: Diagrama de Stanford del componente vertical para el RAIM basado en separación de soluciones con dos satélites con errores constantes para el caso de trayectoria bidimensional.

8.2.2.2.3 IBPL

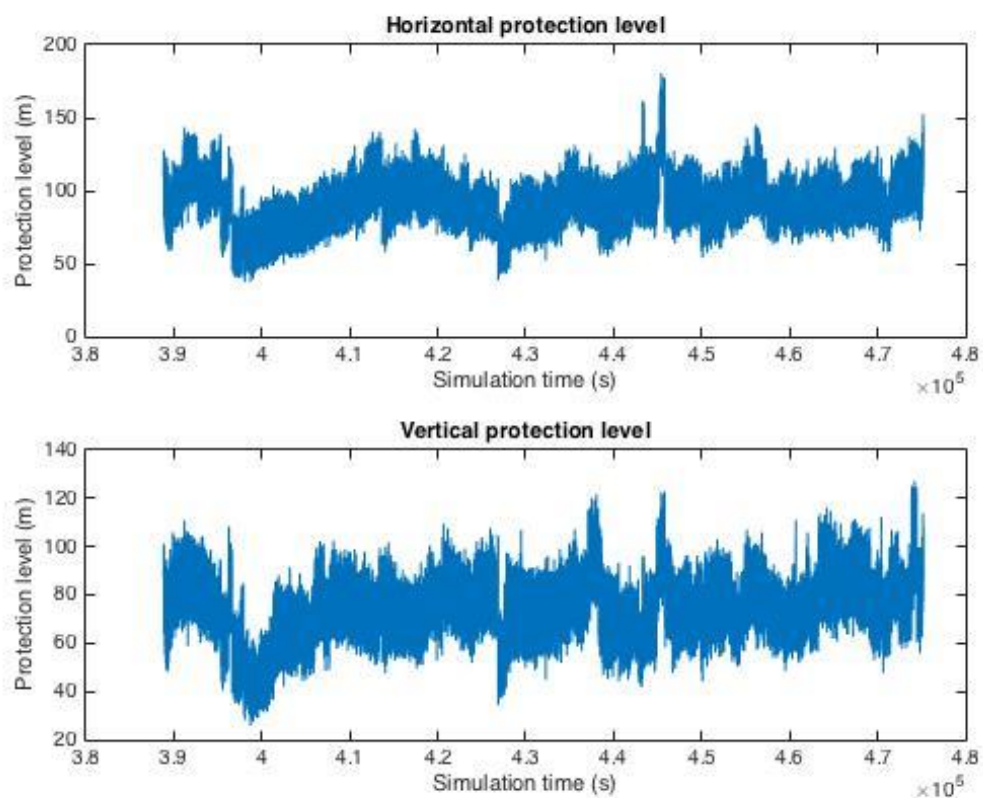


Figura 173: Niveles de protección del IBPL para el caso de trayectoria bidimensional con un satélite con error constante en el pseudorange.

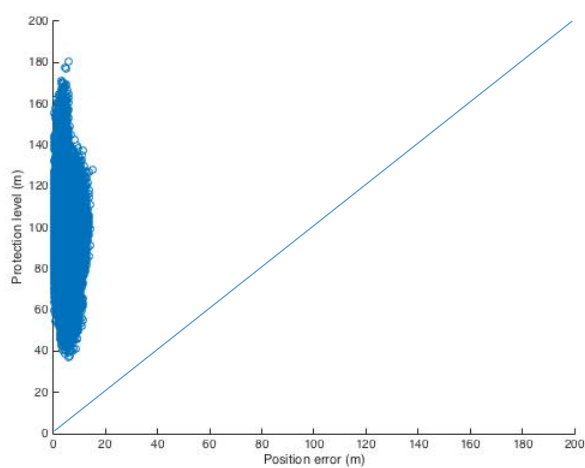


Figura 174: Diagrama de Stanford del componente horizontal para el IBPL con dos satélites con errores constantes para el caso de trayectoria bidimensional.

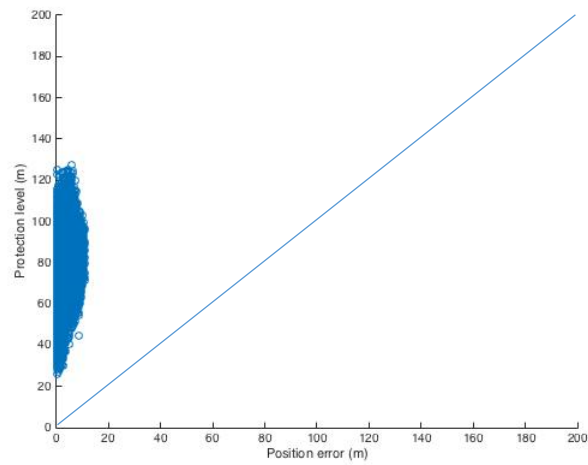


Figura 175: Diagrama de Stanford del componente vertical para el IBPL con dos satélites con errores constantes para el caso de trayectoria bidimensional.

8.2.2.3 Tres satélites con errores constantes en el pseudorange

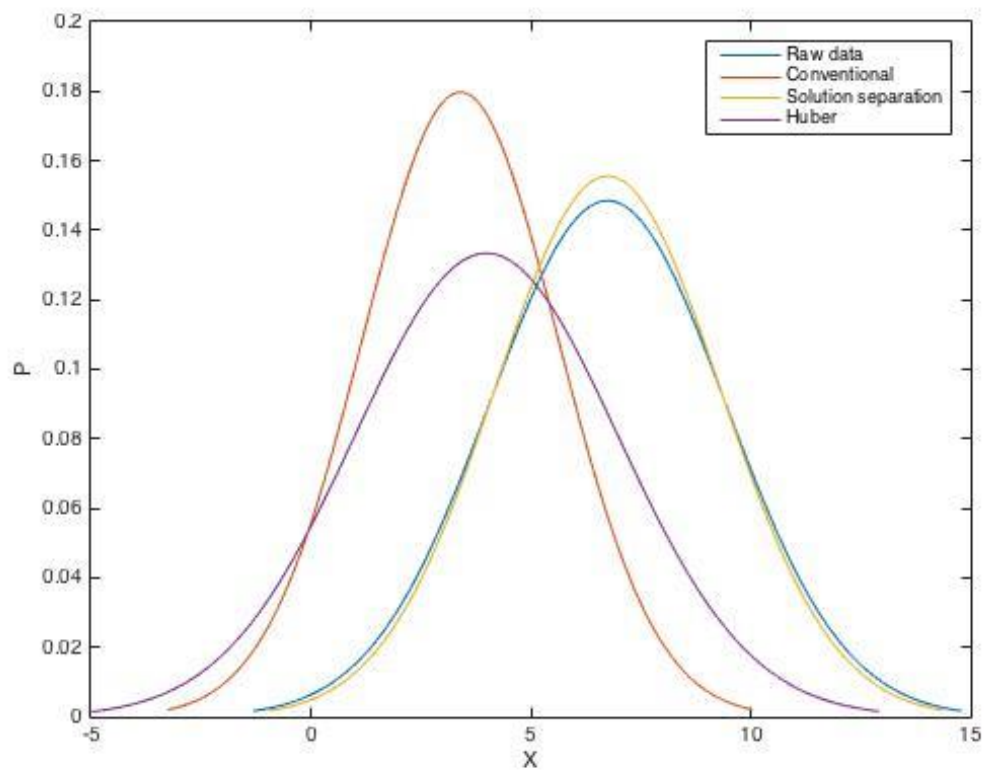


Figura 176: Funciones de densidad de probabilidad del error de posición con tres satélites con errores constantes para el caso de trayectoria bidimensional.

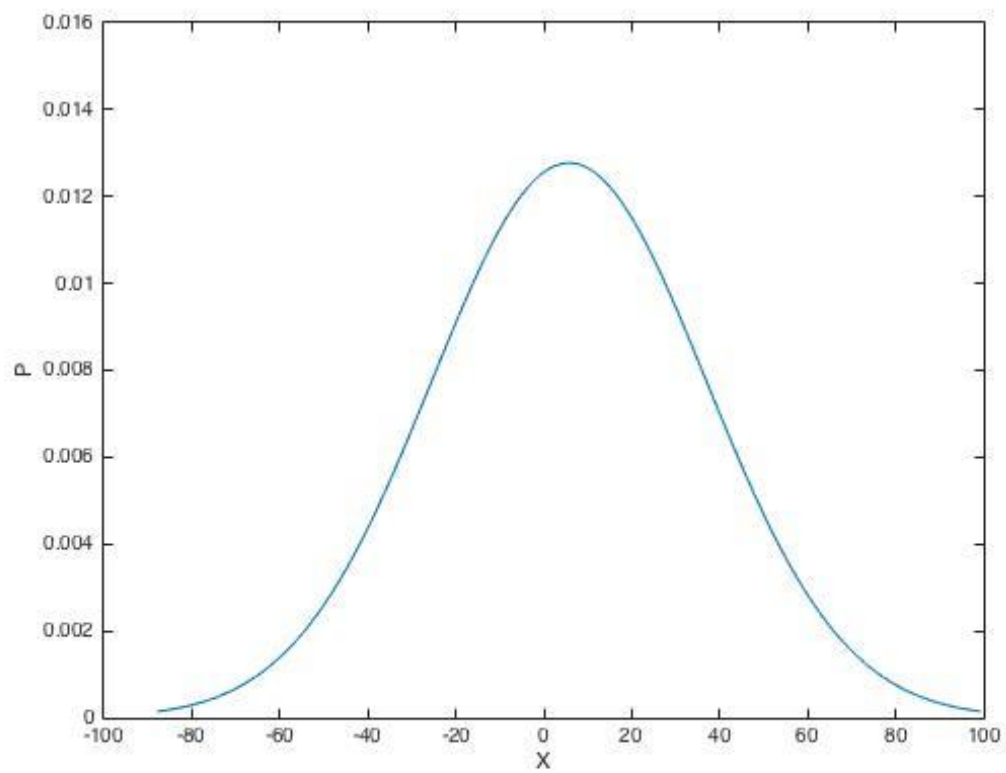


Figura 177: Función de densidad de probabilidad del error de posición para el estimador IGGII con tres satélites con errores constantes para el caso de trayectoria bidimensional.

8.2.2.3.1 RAIM convencional

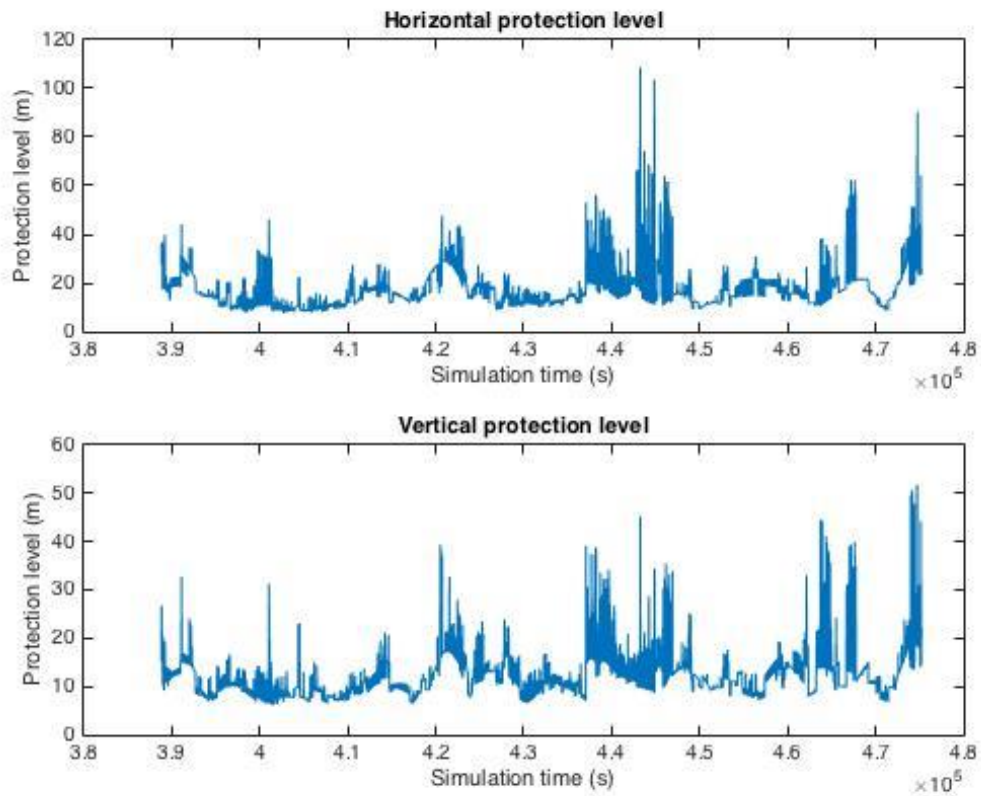


Figura 178: Niveles de protección para el RAIM convencional con tres satélites con errores constantes para el caso de trayectoria bidimensional.

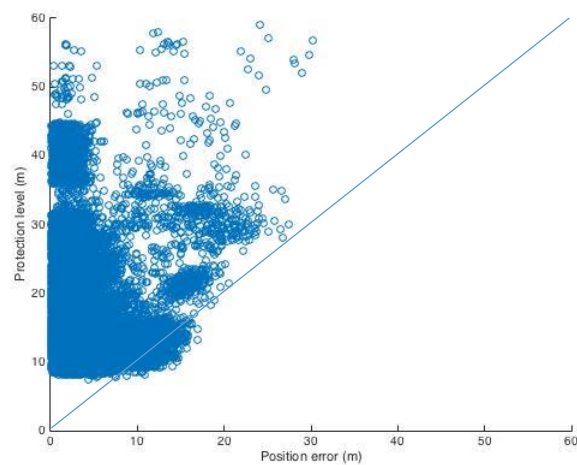


Figura 179: Diagrama de Stanford del componente horizontal para el RAIM convencional con tres satélites con errores constantes para el caso de trayectoria bidimensional.

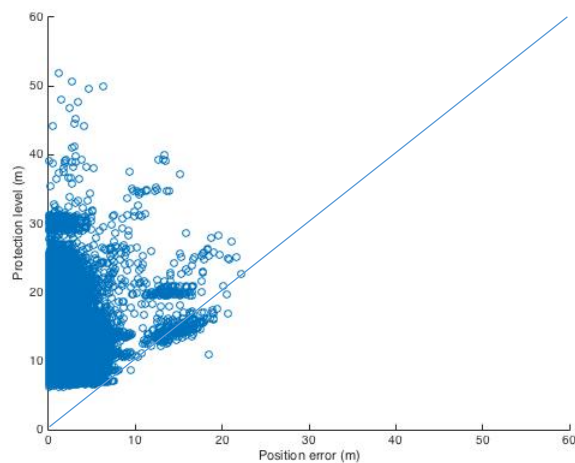


Figura 180: Diagrama de Stanford del componente vertical para el RAIM convencional con tres satélites con errores constantes para el caso de trayectoria bidimensional.

8.2.2.3.2 RAIM basado en separación de soluciones

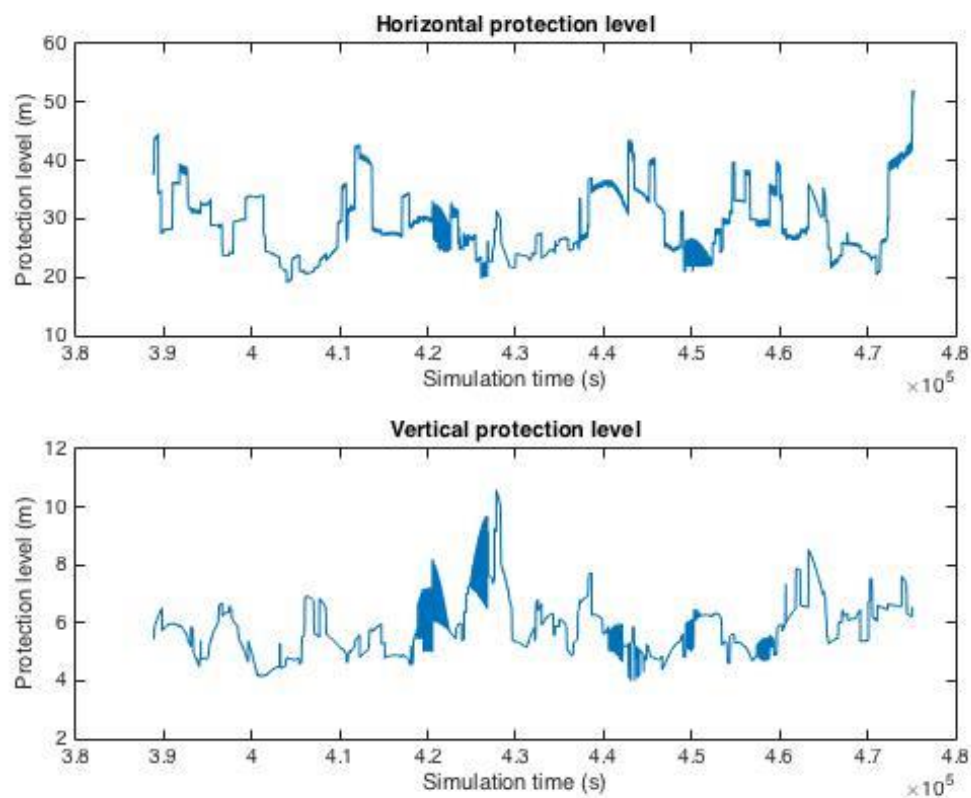


Figura 181: Niveles de protección para el RAIM basado en separación de soluciones con tres satélites con errores constantes para el caso de trayectoria bidimensional.

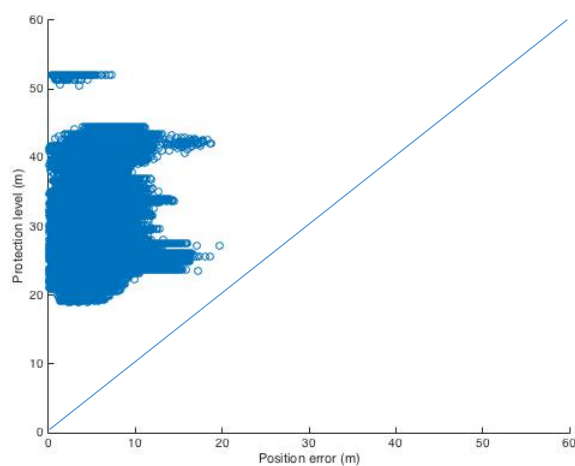


Figura 182: Diagrama de Stanford del componente horizontal para el RAIM basado en separación de soluciones con tres satélites con errores constantes para el caso de trayectoria bidimensional.

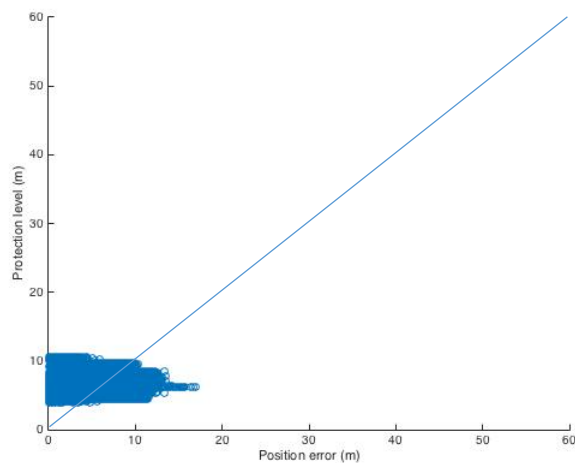


Figura 183: Diagrama de Stanford del componente vertical para el RAIM basado en separación de soluciones con tres satélites con errores constantes para el caso de trayectoria bidimensional.

8.2.2.3.3 IBPL

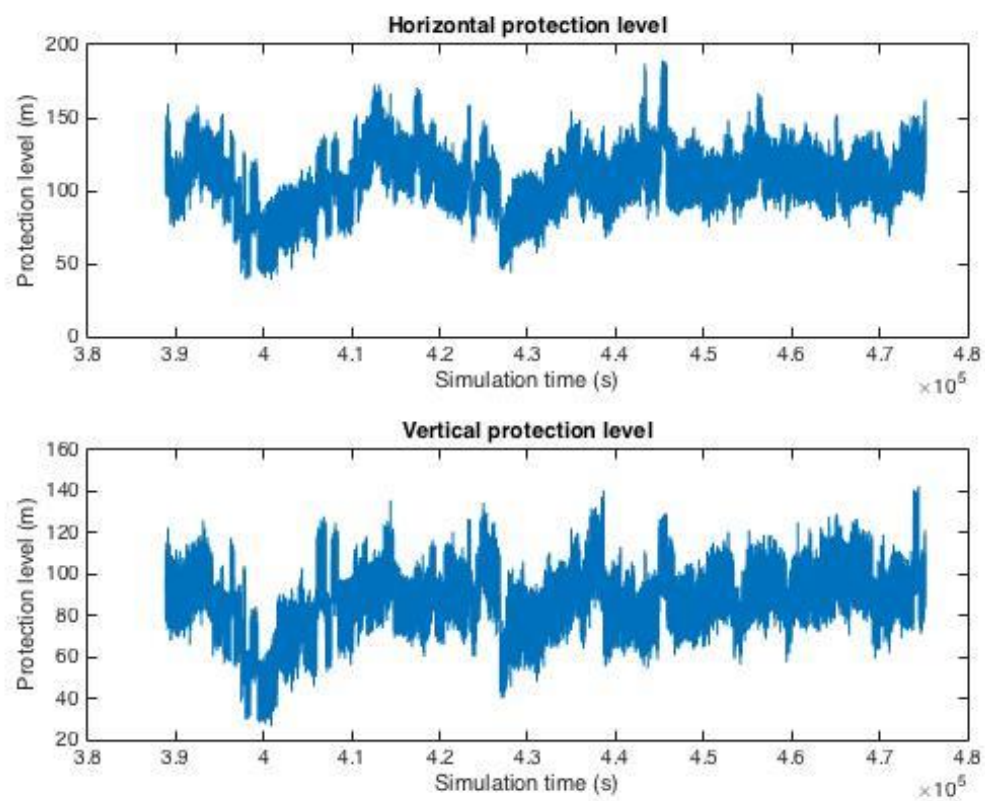


Figura 184: Niveles de protección para el IBPL con tres satélites con errores constantes para el caso de trayectoria bidimensional.

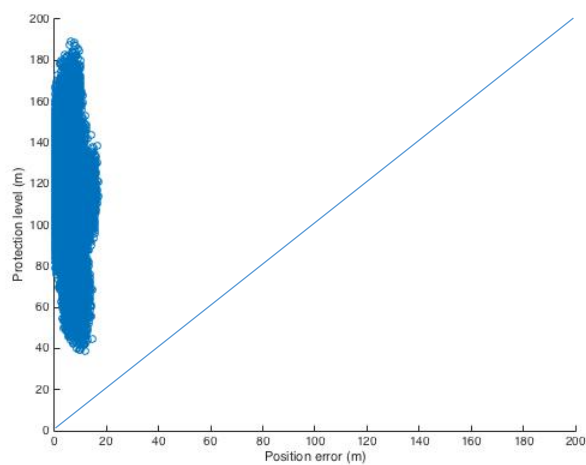


Figura 185: Diagrama de Stanford del componente horizontal para el IBPL con tres satélites con errores constantes para el caso de trayectoria bidimensional.

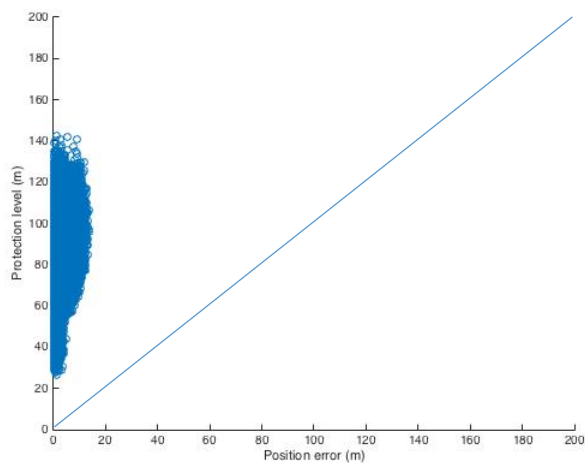


Figura 186: Diagrama de Stanford del componente vertical para el IBPL con tres satélites con errores constantes para el caso de trayectoria bidimensional.

8.3 Código fuente algoritmos detección saltos de ciclo

8.3.1 Single-frequency polynomial algorithm

```
%{
Matlab function for the single-frequency polynomial cycle slip detection
algorithm
%}

function [Tr, Fa] = functionSFPolynomial(NoiseVariable, CS)

clear
clc

% Import data from file
filename = 'GPSDataL1.txt';
filename2 = 'GPSDataL2.txt';
filename3 = 'GPSDataRange.txt';
A = importdata(filename);
B = importdata(filename2);
R = importdata(filename3);

% Physical constants
F1 = 1575.42E06;
C = 299792458;
Lambdal = C/F1;

N1 = R/Lambdal;

Threshold = 4*Lambdal; % Probamos asi y variamos si resulta necesario

% Set noise and cycle slip
NoiseVariable = 0.5;
CS = 5;

% Ionospheric effect

xIono=0:1:10800;
a = 1E-09;
```

```

h = 0;
k = 4;
yIono = a*(xIono-h).^2+k;
for i = 1:10800
    yIono(i) = yIono(i)*(1+randn*0.001);
end
yF1 = yIono;

% Observables
for n = 2:10800
    Phil(n) = R(n) - R(1) + N1(1)*Lambdal +
    (NoiseVariable*0.02*Lambdal*randn) - yF1(n);

    R1(n) = R(n) + (NoiseVariable*2*Lambdal*randn) + yF1(n);

    if (n == 5700) || (n == 5701) || (n == 5702) || (n == 5703)
        IonoScintillation = randn*0.6;
        Phil(n) = Phil(n) - IonoScintillation;
        R1(n) = R(n) + IonoScintillation;
    end
end

% Set window size and algorithm restablishment
WindowSize = 15;
AlgorithmRest = 5;

% Add noise
for n = 2:10800
    %dCS(n) = d(1) + (NoiseVariable*normrnd(0,0.2*Lambdal));
    dCS(n) = d(1) + (NoiseVariable*0.2*Lambdal*randn);
end

% Set cycle slips
CSPos = [500 ; 1000 ; 1500 ; 2000 ; 2500 ; 3000 ; 3500 ; 4000 ; 4500 ;
5000 ; 5500 ; 6000 ; 6500 ;...
7000 ; 7500 ; 8000 ; 8500 ; 9000 ; 9500 ; 10000 ; 10500 ; 0];
CSIndex = 1;

%Add cycle slip

```

```

for h = 1:10800
    if h == CSPos(CSIndex)
        for k = h:10800
            Phil(k) = Phil(k) + (CS*Lambdal);
        end
        CSIndex = CSIndex + 1;
    end
end

d = Phil - R1;

auxVector = zeros(WindowSize,1);
counterCS = 0;
lastCS = 0;

x = linspace(0,WindowSize,WindowSize);
y_plot = zeros(1,10800-WindowSize);

k = 1;
at_least_one = 0;

% CS detection algorithm
for i = (WindowSize+1):10800

    if (lastCS < i-WindowSize-AlgorithmRest) % If the last CS occurred
more than 5 samples away and the algorithm has enough historical data
        auxVector = d(i-WindowSize:i-1);

        p = polyfit(x,auxVector,2);

        x1 = linspace(0,WindowSize+1,WindowSize+1);

        y = polyval(p,x1);

        if (abs(d(i) - y(WindowSize+1)) > Threshold)
            %fprintf('%d - Cycle slip detected\n', i);
            counterCS = counterCS +1;
            lastCS = i;
            arrayCS(k) = i;

```

```

        k = k + 1;
        at_least_one = 1;
    else
        fprintf('%d - OK\n', i);
    end

    y_plot(i) = y(WindowSize+1);
end

end

% Output data
if at_least_one == 0
    OutputDetectedCS = 0;
    OutputFalseDetections = 0;
else
    OutputDetectedCS = 0;
    for i = 1:length(arrayCS)
        for j = 1:21
            if (arrayCS(i) == CSPos(j))
                OutputDetectedCS = OutputDetectedCS + 1;
            end
        end
    end
end

OutputFalseDetections = 0;
OutputFalseDetections = length(arrayCS) - OutputDetectedCS;

end

fprintf('Detected CS: %d\n', OutputDetectedCS);
fprintf('False detections: %d\n', OutputFalseDetections);

% Plot
x2 = linspace(WindowSize+1,10800,10800-WindowSize);

figure(1)
subplot(2,1,1);
scatter(x2,d(WindowSize+1:10800));

```

```

title('Phi - R')

x3 = linspace(1, length(y_plot), length(y_plot));
subplot(2,1,2);
scatter(x3, y_plot);
title('Polynomial fit')

for k = 1:10800
    ThresholdArray(k) = Threshold;
end
figure(2)
scatter(x3, abs(d - y_plot));
hold on
plot(x3, ThresholdArray, 'LineWidth',5);
hold off
title('Polynomial fit - Single Frequency')
legend(' ', 'Umbral')

Tr = OutputDetectedCS;
Fa = OutputFalseDetections;

end

```

8.3.2 Single-frequency mean-sigma algorithm

```
%{
Matlab function for the single-frequency mean-sigma cycle slip detection
algorithm
%}

function [Tr, Fa] = functionSFMeanSigma(NoiseVariable, CS)

clear
clc

% Import data from file
filename = 'GPSDataL1.txt';
filename2 = 'GPSDataL2.txt';
filename3 = 'GPSDataRange.txt';
A = importdata(filename);
B = importdata(filename2);
R = importdata(filename3);

% Physical constants
F1 = 1575.42E06;
C = 299792458;
Lambda1 = C/F1;

N1 = R/Lambda1;

WindowSize = 15;
AlgorithmRestablishment = 5;

% Set noise and cycle slip
NoiseVariable = 0.5;
CS = 5;

% Ionospheric effect
xIono=0:1:10800;
a = 1E-09;
h = 0;
k = 4;
```



```

yIono = a*(xIono-h).^2+k;
for i = 1:10800
    yIono(i) = yIono(i)*(1+randn*0.001);
end
yF1 = yIono;

% Observables
for n = 2:10800
    Phil(n) = R(n) - R(1) + N1(1)*Lambdal +
    (NoiseVariable*0.02*Lambdal*randn) - yF1(n);

    R1(n) = R(n) + (NoiseVariable*2*Lambdal*randn) + yF1(n);

    if (n == 5700) || (n == 5701) || (n == 5702) || (n == 5703)
        IonoScintillation = randn*0.6;
        Phil(n) = Phil(n) - IonoScintillation;
        R1(n) = R(n) + IonoScintillation;
    end
end

% Add noise
for n = 2:10800
    %dCS(n) = d(1) + (NoiseVariable*normrnd(0,0.2*Lambdal));
    dCS(n) = d(1) + (NoiseVariable*0.2*Lambdal*randn);
end

% Set cycle slips
CSPos = [500 ; 1000 ; 1500 ; 2000 ; 2500 ; 3000 ; 3500 ; 4000 ; 4500 ;
5000 ; 5500 ; 6000 ; 6500 ; ...
7000 ; 7500 ; 8000 ; 8500 ; 9000 ; 9500 ; 10000 ; 10500 ; 0];
CSIndex = 1;

%Add cycle slip
for h = 1:10800
    if h == CSPos(CSIndex)
        for k = h:10800
            Phil(k) = Phil(k) + (CS*Lambdal);
        end
        CSIndex = CSIndex + 1;
    end
end

```

```

    end
end

d = Phil - R1;

counterCS = 0;
lastCS = -5;

Md = zeros(1, 10800);
Md2 = zeros(1, 10800);
Sd = zeros(1, 10800);

k = 1;

at_least_one = 0;

% Loop for all measures
for i = 1:10800

    if (i > WindowSize) && (i - lastCS > WindowSize +
AlgorithmRestablishment)
        Md(i) = mean(d(i-WindowSize:i-1));
    end

    if (i > WindowSize) && (i - lastCS > WindowSize +
AlgorithmRestablishment)
        Md2(i) = mean(d(i-WindowSize:i-1).^2);
    end

    if (i > WindowSize) && (i - lastCS > WindowSize +
AlgorithmRestablishment)
        Sd(i) = sqrt(abs(Md2(i) - (Md(i)^2)));
    end

    if (i > WindowSize) && (i - lastCS > WindowSize +
AlgorithmRestablishment)
        if (abs(d(i) - Md(i)) > 4*Sd(i))
            %fprintf('%d - Cycle slip detected\n', i);
            counterCS = counterCS + 1;
        end
    end
end

```

```

        lastCS = i;
        arrayCS(k) = i;
        k = k + 1;
        at_least_one = 1;
    end
end

end

% Output data
if at_least_one == 0
    OutputDetectedCS = 0;
    OutputFalseDetections = 0;
else
    OutputDetectedCS = 0;
    for i = 1:length(arrayCS)
        for j = 1:21
            if (arrayCS(i) == CSPos(j))
                OutputDetectedCS = OutputDetectedCS + 1;
            end
        end
    end

    OutputFalseDetections = 0;
    OutputFalseDetections = length(arrayCS) - OutputDetectedCS;
end

fprintf('Detected CS: %d\n', OutputDetectedCS);
fprintf('False detections: %d\n', OutputFalseDetections);

% Plot
figure(1)
subplot(4,1,1);
x = linspace(1,10800,10800);
scatter(x,d(1:10800));
title('Phi-R')

ThresholdArray = 4*Sd;
auxArray = abs(d - Md');

```

```

subplot(4,1,2);
scatter(x, auxArray(1:10800));
hold on
plot(x, ThresholdArray);
title('Mean and Sigma - Single Frequency')
legend(' ', 'Umbral')

```

```

subplot(4,1,3);
scatter(x, Md(1:10800));
title('Mean')

```

```

subplot(4,1,4);
scatter(x, Sd(1:10800));
title('Sd')

```

```

figure(2)
ThresholdArray = 4*Sd;
for i = 1:10800
    auxArray(i) = abs(d(i) - Md(i));
end
scatter(x, auxArray(1:10800));
hold on
plot(x, ThresholdArray, 'LineWidth',2);
title('Mean and Sigma - Single Frequency')
legend(' ', 'Umbral')

```

```

Tr = OutputDetectedCS;
Fa = OutputFalseDetections;

```

```

end

```

8.3.3 Multi-frequency geometry-free combination algorithm

```
%{
Matlab function for the multi-frequency geometry free cycle slip detection
algorithm
%}

function [Tr, Fa] = functionMFGeoFree(NoiseVariable, CS1, CS2)

clear
clc

filename = 'GPSDataL1.txt';
filename2 = 'GPSDataL2.txt';
filename3 = 'GPSDataRange.txt';
A = importdata(filename);
B = importdata(filename2);
R = importdata(filename3);

F1 = 1575.42E06;
F2 = 1227.60E06;
C = 299792458;
Lambda1 = C/F1;
Lambda2 = C/F2;

a0 = 1.5*(Lambda2 - Lambda1);
a1 = a0/2;
Threshold = a0 - (a1*exp(-1/10));
% Threshold = Threshold*2;

N1 = R/Lambda1;
N2 = R/Lambda2;

Set noise and cycle slips
NoiseVariable = 1;
CS1 = 2;
CS2 = 0;

% Set window size and algorithm reestablishment
```

```

WindowSize = 15;
AlgorithmRest = 5;

% Ionospheric effect

xIono=0:1:10800;
a = 1E-09;
h = 0;
k = 4;
yIono = a*(xIono-h).^2+k;
for i = 1:10800
    yIono(i) = yIono(i)*(1+randn*0.001);
end
yF1 = yIono;
yF2 = yIono*(F1/F2)^2;

% Add noise
for n = 1:10800
    phaseObsv1(n) = R(n) + (N1(1)*Lambda1) +
    (NoiseVariable*0.02*Lambda1*randn - yF1(n));
    phaseObsv2(n) = R(n) + (N2(1)*Lambda2) +
    (NoiseVariable*0.02*Lambda2*randn) - yF2(n);

    if (n == 5700) || (n == 5701) || (n == 5702) || (n == 5703)
        IonoScintillation = randn*0.6;
        phaseObsv1(n) = phaseObsv1(n) - IonoScintillation;
        phaseObsv2(n) = phaseObsv2(n) - (IonoScintillation*(F1/F2)^2);
    end
end

% Set cycle slips
CSPos = [500 ; 1000 ; 1500 ; 2000 ; 2500 ; 3000 ; 3500 ; 4000 ; 4500 ;
5000 ; 5500 ; 6000 ; 6500 ; ...
7000 ; 7500 ; 8000 ; 8500 ; 9000 ; 9500 ; 10000 ; 10500 ; 0];
CSIndex = 1;

%Add cycle slip
for h = 1:10800
    if h == CSPos(CSIndex)

```

```

        for k = h:10800
            phaseObsv1(k) = phaseObsv1(k) + (CS1*Lambda1);
            phaseObsv2(k) = phaseObsv2(k) + (CS2*Lambda2);
        end
        CSIndex = CSIndex + 1;
    end
end

geometryFreeCombination = phaseObsv1 - phaseObsv2;
% y(ti)
for n = 2:10800
    geometryFreeCombinationDif(n) = geometryFreeCombination(n) -
    geometryFreeCombination(1);
end

% % Differentiate
% for n = 2:10800
%     geometryFreeCombinationDif2(n) = geometryFreeCombinationDif(n) -
    geometryFreeCombinationDif(n-1);
% end
% % dif2_y
% for n = 2:10800
%     geometryFreeCombinationDif3(n) = geometryFreeCombinationDif2(n) -
    geometryFreeCombinationDif2(n-1);
% end
% % dif3_y
% for n = 2:10800
%     geometryFreeCombinationDif4(n) = geometryFreeCombinationDif3(n) -
    geometryFreeCombinationDif3(n-1);
% end
% % dif4_y
% for n = 2:10800
%     geometryFreeCombinationDif5(n) = geometryFreeCombinationDif4(n) -
    geometryFreeCombinationDif4(n-1);
% end

counterCS = 0;

auxVector = zeros(WindowSize,1);

```

```

x = linspace(0,WindowSize,WindowSize);
xP = linspace(0,WindowSize-1,WindowSize-1);

y_plot = zeros(1,10800-WindowSize);

lastCS = -5;

k = 1;

potentialCS = 0;

% Loop
for i = (WindowSize+1):10800

    if (lastCS < i-WindowSize-AlgorithmRest) % If the last CS occurred
more than 5 samples away and the algorithm has enough historical data

        % Ignore previous sample if it is a potential cycle slip
        if (potentialCS == 1)
            auxVector = geometryFreeCombinationDif(i-WindowSize:i-2);

            p = polyfit(xP,auxVector,2);

            x1 = linspace(0,WindowSize+1,WindowSize+1);

            y = polyval(p,x1);
        else
            auxVector = geometryFreeCombinationDif(i-WindowSize:i-1);

            p = polyfit(x,auxVector,2);

            x1 = linspace(0,WindowSize+1,WindowSize+1);

            y = polyval(p,x1);
        end

        if (abs(geometryFreeCombinationDif(i) - y(WindowSize+1)) >
Threshold)

```



```

        %fprintf('%d - Cycle slip detected\n', i);

        % If two CS are detected then declare CS
        if (potentialCS == 1)
            counterCS = counterCS +1;
            lastCS = i-1;
            arrayCS(k) = i-1;
            k = k + 1;
            potentialCS = 0;
        else
            potentialCS = 1;
        end

    else
        %fprintf('%d - OK\n', i);
        potentialCS = 0;
    end

    y_plot(i-WindowSize) = y(WindowSize+1);

end

end

% Threshold line for plotting
x2 = linspace(0,10800-WindowSize,10800-WindowSize);
auxArray = abs(geometryFreeCombinationDif(WindowSize+1:10800) -
y_plot);
ThresholdArray = zeros(1, 10800-WindowSize);
for k = 1:(10800-WindowSize)
    ThresholdArray(k) = Threshold;
end

% Output data
OutputDetectedCS = 0;
for i = 1:length(arrayCS)
    for j = 1:21
        if (arrayCS(i) == CSPos(j))
            OutputDetectedCS = OutputDetectedCS + 1;
        end
    end
end

```

```

        end
    end
end

OutputFalseDetections = 0;
OutputFalseDetections = length(arrayCS) - OutputDetectedCS;

fprintf('Detected CS: %d\n', OutputDetectedCS);
fprintf('False detections: %d\n', OutputFalseDetections);

% Plot
subplot(2,1,1);
scatter(x2,auxArray);
hold on
scatter(x2, ThresholdArray);
hold off
title('Geometry free combination - polynomial fit')
legend(' ','Threshold')

subplot(2,1,2);
scatter(x2, y_plot);
title('Polynomial fit')

Tr = OutputDetectedCS;
Fa = OutputFalseDetections;

end

```

8.3.4 Multi-frequency Melbourne-Wübbena combination algorithm

```
%{
Matlab function for the multi-frequency Melbourne-Wubben cycle slip
detection algorithm
%}

function [Tr, Fa] = functionMFMelbourneWubben(NoiseVariable, CS1, CS2)

clear
clc

filename = 'GPSDataL1.txt';
filename2 = 'GPSDataL2.txt';
filename3 = 'GPSDataRange.txt';
A = importdata(filename);
B = importdata(filename2);
R = importdata(filename3);

counterCS = 0;
lastCS = -5;

F1 = 1575.42E06;
F2 = 1227.60E06;
C = 299792458;
Lambda1 = C/F1;
Lambda2 = C/F2;

Kfactor = 4;

N1 = R/Lambda1;
N2 = R/Lambda2;

% Set noise and cycle slips
CS1 = 2;
CS2 = 0;
CS = (Lambda1*CS1) - (Lambda2*CS2);
NoiseVariable = 1;
```

```

% Ionospheric effect

xIono=0:1:10800;
a = 1E-09;
h = 0;
k = 4;
yIono = a*(xIono-h).^2+k;
for i = 1:10800
    yIono(i) = yIono(i)*(1+randn*0.001);
end
yF1 = yIono;
yF2 = yIono*(F1/F2)^2;

% Observables
for n = 2:10800
    Phi1(n) = R(n) - R(1) + N1(1)*Lambda1 +
    (NoiseVariable*0.02*Lambda1*randn) - yF1(n);
    Phi2(n) = R(n) - R(1) + N2(1)*Lambda2 +
    (NoiseVariable*0.02*Lambda2*randn) - yF2(n);

    R1(n) = R(n) + (NoiseVariable*2*Lambda1*randn) + yF1(n);
    R2(n) = R(n) + (NoiseVariable*2*Lambda2*randn) + yF2(n);

    if (n == 5700) || (n == 5701) || (n == 5702) || (n == 5703)
        IonoScintillation = randn*0.6;
        Phi1(n) = Phi1(n) - IonoScintillation;
        Phi2(n) = Phi2(n) - (IonoScintillation*(F1/F2)^2);
        R1(n) = R1(n) + IonoScintillation;
        R2(n) = R2(n) + (IonoScintillation*(F1/F2)^2);
    end
end

% Set algorithm restablishment
AlgorithmRest = 5;

% Set cycle slips
CSPos = [500 ; 1000 ; 1500 ; 2000 ; 2500 ; 3000 ; 3500 ; 4000 ; 4500 ;
5000 ; 5500 ; 6000 ; 6500 ;...
7000 ; 7500 ; 8000 ; 8500 ; 9000 ; 9500 ; 10000 ; 10500 ; 0];

```

```

CSIndex = 1;

%Add cycle slip
for h = 1:10800
    if h == CSPos(CSIndex)
        for k = h:10800
            Phi1(k) = Phi1(k) + CS1*Lambda1;
            Phi2(k) = Phi2(k) + CS2*Lambda2;
        end
        CSIndex = CSIndex + 1;
    end
end

% Melbourne-Wubben Combination
PhiW = ((F1*Phi1) - (F2*Phi2))/(F1 - F2);
Rn = ((F1*R1) + (F2*R2))/(F1 + F2);
Bw = PhiW - Rn;

% Compute combination wave length
LambdaW = C/(F1-F2);

Mbw = zeros(1,10800);
Sbw2 = zeros(1,10800);
Sbw2_0 = (LambdaW/2)^2;

k = 1;

at_least_one = 0;

for i = 1:10800
    if(i-lastCS > AlgorithmRest)
        if( ((i ~= 1) && (i - lastCS ~= AlgorithmRest + 1)) && (abs(Bw(i)
- Mbw(i-1)) > Kfactor*sqrt(Sbw2(i-1))))
            counterCS = counterCS + 1;
            lastCS = i;
            arrayCS(k) = i;
            k = k + 1;
            at_least_one = 1;
        end
    end
end

```

```

    if (i == 1) || (i-lastCS == AlgorithmRest + 1)
        Mbw(i) = Bw(i);
        Sbw2(i) = Sbw2_0;
    elseif (lastCS == -5)
        Mbw(i) = (((i-1)/i)*Mbw(i-1)) + (Bw(i)/i);
        Sbw2(i) = (((i-1)/i)*Sbw2(i-1)) + ((Bw(i)-Mbw(i-1))^2/i);
    else
        j = i - lastCS - AlgorithmRest;
        Mbw(i) = (((j-1)/j)*Mbw(i-1)) + (Bw(i)/j);
        Sbw2(i) = (((j-1)/j)*Sbw2(i-1)) + ((Bw(i)-Mbw(i-1))^2/j);
    end
end

end

end

for i = 1:10800
    if (i-lastCS > AlgorithmRest)
        if (i == 1) || (i-lastCS == AlgorithmRest + 1)
            Mbw(i) = Bw(i);
            Sbw2(i) = Sbw2_0;
        elseif (lastCS == -5)
            Mbw(i) = (((i-1)/i)*Mbw(i-1)) + (Bw(i)/i);
            Sbw2(i) = (((i-1)/i)*Sbw2(i-1)) + ((Bw(i)-Mbw(i-1))^2/i);
        else
            j = i - lastCS - AlgorithmRest;
            Mbw(i) = (((j-1)/j)*Mbw(i-1)) + (Bw(i)/j);
            Sbw2(i) = (((j-1)/j)*Sbw2(i-1)) + ((Bw(i)-Mbw(i-1))^2/j);
        end

        if (abs(Bw(i) - Mbw(i)) > Kfactor*sqrt(Sbw2(i)))
            fprintf('%d - Cycle slip detected\n', i);
            counterCS = counterCS + 1;
            lastCS = i;
            arrayCS(k) = i;
            k = k + 1;
            at_least_one = 1;
        else
            fprintf('%d - OK\n', i);

```

```

        end
    end
end

% Output data
if at_least_one == 0
    OutputDetectedCS = 0;
    OutputFalseDetections = 0;
else
    OutputDetectedCS = 0;
    for i = 1:length(arrayCS)
        for j = 1:(length(CSPos)-1)
            if (arrayCS(i) == CSPos(j))
                OutputDetectedCS = OutputDetectedCS + 1;
            end
        end
    end

    OutputFalseDetections = 0;
    OutputFalseDetections = length(arrayCS) - OutputDetectedCS;
end

fprintf('Detected CS: %d\n', OutputDetectedCS);
fprintf('False detections: %d\n', OutputFalseDetections);

% Plot
x = linspace(0,10800,10800);

subplot(3,1,1);
scatter(x, Bw);
title('Melbourne-Wubben combiantion')

% subplot(3,1,2);
auxArray = abs(Bw - Mbw);
scatter(x, auxArray);
title('(Melbourne-Wubben combiantion) - Mean')

% subplot(3,1,3);
scatter(x, Mbw);

```

```

title('Mean')

auxArray = abs(Bw - Mbw);
scatter(x,auxArray);
hold on
scatter(x, Kfactor*sqrt(abs(Sbw2)), 'LineWidth', 1);
hold off
title('Melbourne-Wubben combiantion - Multi frequency')
legend(' ','Umbra1')

Tr = OutputDetectedCS;
Fa = OutputFalseDetections;

end

```


8.4 Código fuente algoritmos selección de satélites

8.4.1 Optimal Satellite Selection Algorithm (OSSA)

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% O SSA Algorithm %%%%%%%%%%
% Calculate the PDOP of all possible satellite combinations and select
% the combination with the lower value

function [dX, MinPDOP_OSSA, GDOP, SelectedSatellitesArray] =
OSSA(DataMatrix, RangeEst, RecPosEst, NumSat)

% Variable declaration
MinPDOP_OSSA = 1000;

% Obtain number of satellites in the data matrix
[DataMatrixRows, DataMatrixColumns] = size(DataMatrix);

% Calculate number of possible combinations of 12 satellites
CombNum_OSSA =
factorial(DataMatrixRows)/(factorial(NumSat)*factorial(DataMatrixRows -
NumSat));

% Obtain a matrix with the different combinations of 12 satellites
for n = 1:DataMatrixRows
    SatellitesForCombination_OSSA(n) = n;
end

CombinationOfSatellites = nchoosek(SatellitesForCombination_OSSA,
NumSat);

% For each combination calculate PDOP

for n = 1:CombNum_OSSA
    % Build matrix of NumSat satellites containing x,y,z,GPS/Galileo
    MatrixForPDOP_OSSA = [DataMatrix(CombinationOfSatellites(n,1), 2)
DataMatrix(CombinationOfSatellites(n,1), 3)
DataMatrix(CombinationOfSatellites(n,1), 4)
DataMatrix(CombinationOfSatellites(n,1), 5)];
    for m = 2:NumSat

```

```

        MatrixForPDOP_OSSA = [MatrixForPDOP_OSSA;
DataMatrix(CombinationOfSatellites(n,m), 2)
DataMatrix(CombinationOfSatellites(n,m), 3)
DataMatrix(CombinationOfSatellites(n,m), 4)
DataMatrix(CombinationOfSatellites(n,m), 5)];
    end

    % RangeEst of combination
    RangeEst_OSSA = [RangeEst(CombinationOfSatellites(n,1))];
    for m = 2:NumSat
        RangeEst_OSSA = [RangeEst_OSSA;
RangeEst(CombinationOfSatellites(n,m))];
    end

    CalculatedPDOP_OSSA(n) = PDOPvar(MatrixForPDOP_OSSA, RecPosEst,
RangeEst_OSSA);

    if (CalculatedPDOP_OSSA(n) < MinPDOP_OSSA)
        MinPDOP_OSSA = CalculatedPDOP_OSSA(n);
        MinPDOPCombinationIndex_OSSA = n;
        MinPDOPRangeEst_OSSA = RangeEst_OSSA;
    end
end

% Compose matrix for least squares method
LastColumnSelectedSat_OSSA = [
DataMatrix(CombinationOfSatellites(MinPDOPCombinationIndex_OSSA,1),
1)];
for m = 2:NumSat
    LastColumnSelectedSat_OSSA = [LastColumnSelectedSat_OSSA;
DataMatrix(CombinationOfSatellites(MinPDOPCombinationIndex_OSSA,m),
1)];
end

LSMatAux =
[DataMatrix(CombinationOfSatellites(MinPDOPCombinationIndex_OSSA,1), 2)
DataMatrix(CombinationOfSatellites(MinPDOPCombinationIndex_OSSA,1), 3)

```

```

DataMatrix(CombinationOfSatellites(MinPDOPCombinationIndex_OSSA,1), 4)
DataMatrix(CombinationOfSatellites(MinPDOPCombinationIndex_OSSA,1),
5)];
for m = 2:NumSat
    LSMatAux = [LSMatAux;
DataMatrix(CombinationOfSatellites(MinPDOPCombinationIndex_OSSA,m), 2)
DataMatrix(CombinationOfSatellites(MinPDOPCombinationIndex_OSSA,m), 3)
DataMatrix(CombinationOfSatellites(MinPDOPCombinationIndex_OSSA,m), 4)
DataMatrix(CombinationOfSatellites(MinPDOPCombinationIndex_OSSA,m),
5)];
end

LSMat_OSSA = [LSMatAux LastColumnSelectedSat_OSSA];

% Call least squares method function
% RangeEst of combination
dX = LeastSquaresV2(LSMat_OSSA, MinPDOPRangeEst_OSSA, RecPosEst);

% GDOP
GDOP = GDOPvar(LSMatAux, RecPosEst, MinPDOPRangeEst_OSSA);

% Output selected satellites
SelectedSatellitesArray =
CombinationOfSatellites(MinPDOPCombinationIndex_OSSA, :);

end

```

8.4.2 Highest Elevation Algorithm (HEA)

```

##### HEA Algorithm #####
% Calculate the elevation for all visible satellite and select
% the 12 satellites with the higher elevation wrt the receiver

function [dX, MaxElevationArray, PDOP_HEA, GDOP] = HEA(DataMatrix,
RangeEst, RecPosEst, SatelliteElevation)

MaxElevationArray = zeros(2, 12);

% Elevation in rads
for n = 1:size(DataMatrix, 1)
    E2(n) = SatelliteElevation(n)*pi/180;
end

[MaxElevationArray(2,1), MaxElevationArray(1,1)] = max(E2);
E2(MaxElevationArray(1,1)) = 0;

[MaxElevationArray(2,2), MaxElevationArray(1,2)] = max(E2);
E2(MaxElevationArray(1,2)) = 0;

[MaxElevationArray(2,3), MaxElevationArray(1,3)] = max(E2);
E2(MaxElevationArray(1,3)) = 0;

[MaxElevationArray(2,4), MaxElevationArray(1,4)] = max(E2);
E2(MaxElevationArray(1,4)) = 0;

[MaxElevationArray(2,5), MaxElevationArray(1,5)] = max(E2);
E2(MaxElevationArray(1,5)) = 0;

[MaxElevationArray(2,6), MaxElevationArray(1,6)] = max(E2);
E2(MaxElevationArray(1,6)) = 0;

[MaxElevationArray(2,7), MaxElevationArray(1,7)] = max(E2);
E2(MaxElevationArray(1,7)) = 0;

[MaxElevationArray(2,8), MaxElevationArray(1,8)] = max(E2);
E2(MaxElevationArray(1,8)) = 0;

```

```
[MaxElevationArray(2,9), MaxElevationArray(1,9)] = max(E2);
E2(MaxElevationArray(1,9)) = 0;
```

```
[MaxElevationArray(2,10), MaxElevationArray(1,10)] = max(E2);
E2(MaxElevationArray(1,10)) = 0;
```

```
[MaxElevationArray(2,11), MaxElevationArray(1,11)] = max(E2);
E2(MaxElevationArray(1,11)) = 0;
```

```
[MaxElevationArray(2,12), MaxElevationArray(1,12)] = max(E2);
E2(MaxElevationArray(1,12)) = 0;
```

```
% Build LS matrix
```

```
LSMat_HEA = [DataMatrix(MaxElevationArray(1,1),2:5);
    DataMatrix(MaxElevationArray(1,2),2:5);
    DataMatrix(MaxElevationArray(1,3),2:5);
    DataMatrix(MaxElevationArray(1,4),2:5);
    DataMatrix(MaxElevationArray(1,5),2:5);
    DataMatrix(MaxElevationArray(1,6),2:5);
    DataMatrix(MaxElevationArray(1,7),2:5);
    DataMatrix(MaxElevationArray(1,8),2:5);
    DataMatrix(MaxElevationArray(1,9),2:5);
    DataMatrix(MaxElevationArray(1,10),2:5);
    DataMatrix(MaxElevationArray(1,11),2:5);
    DataMatrix(MaxElevationArray(1,12),2:5)];
```

```
LastColumnSelectedSat_HEA = [ DataMatrix(MaxElevationArray(1,1), 1); ...
    DataMatrix(MaxElevationArray(1,2), 1); ...
    DataMatrix(MaxElevationArray(1,3), 1); ...
    DataMatrix(MaxElevationArray(1,4), 1); ...
    DataMatrix(MaxElevationArray(1,5), 1); ...
    DataMatrix(MaxElevationArray(1,6), 1); ...
    DataMatrix(MaxElevationArray(1,7), 1); ...
    DataMatrix(MaxElevationArray(1,8), 1); ...
    DataMatrix(MaxElevationArray(1,9), 1); ...
    DataMatrix(MaxElevationArray(1,10), 1); ...
    DataMatrix(MaxElevationArray(1,11), 1); ...
```

```

    DataMatrix(MaxElevationArray(1,12), 1)];

RangeEst_HEA = [RangeEst(MaxElevationArray(1,1));
    RangeEst(MaxElevationArray(1,2));
    RangeEst(MaxElevationArray(1,3));
    RangeEst(MaxElevationArray(1,4));
    RangeEst(MaxElevationArray(1,5));
    RangeEst(MaxElevationArray(1,6));
    RangeEst(MaxElevationArray(1,7));
    RangeEst(MaxElevationArray(1,8));
    RangeEst(MaxElevationArray(1,9));
    RangeEst(MaxElevationArray(1,10));
    RangeEst(MaxElevationArray(1,11));
    RangeEst(MaxElevationArray(1,12))];

% Calculate PDOP
PDOP_HEA = PDOPvar(LSMat_HEA, RecPosEst, RangeEst_HEA);

% GDOP
GDOP = GDOPvar(LSMat_HEA, RecPosEst, RangeEst_HEA);

LSMat_HEA = [LSMat_HEA LastColumnSelectedSat_HEA];

%Call LS function
dX = LeastSquaresV2(LSMat_HEA, RangeEst_HEA, RecPosEst);

end

```

8.4.3 Maximum Volume Algorithm (MVA)

```

##### MVA Algorithm #####
% This algorithm follows the principle of the maximum volume tetrahedron
% Select the 4 satellites which give the tetrahedron of the greatest
volume
% Satellite 1: Higher elevation
% Satellite 2: Angle of 109.5° with the first satellite
% Satellite 3: Maximum volume of a tetrahedron with virtual line of
sight
% Satellite 4: Maximum volume of a tetrahedron

function [dX, LSMat, SelectedSatellite_MVA, PDOP_MVA, GDOP] =
MVA(DataMatrix, RangeEst, RecPosEst, MaxElevationArray)

AngleS2_MVA = 109.5;
AngleClosest_MVA = 1000;

% Step 1: Select satellite with largest elevation
S1_MVA = DataMatrix(MaxElevationArray(1,1),:);

% GPS or Galileo
if (S1_MVA(5) == 0)
    GalileoSelection_MVA = 0;
else
    GalileoSelection_MVA = 1;
end

% Step 2: Select satellite with angle close to 109.5°
for n = 1:size(DataMatrix, 1)
    if ((n ~= MaxElevationArray(1,1)) && (DataMatrix(n,5) ==
GalileoSelection_MVA))
        Angle_MVA = AngleBetweenVectors([S1_MVA(2), S1_MVA(3),
S1_MVA(4)], [DataMatrix(n, 2), DataMatrix(n, 3), DataMatrix(n, 4)]);

        if (abs(Angle_MVA*180/pi - AngleS2_MVA) <
abs(AngleClosest_MVA*180/pi - AngleS2_MVA))
            AngleClosest_MVA = Angle_MVA;
            AngleClosestIndex_MVA = n;
        end
    end
end

```

```

end

end

S2_MVA = DataMatrix(AngleClosestIndex_MVA,:);

% Step 3: Maximize volume
l2_MVA = cos(AngleClosest_MVA);
m2_MVA = sin(AngleClosest_MVA);
VolMax3_MVA = 0;
for n = 1:size(DataMatrix, 1)
    if ((n ~= MaxElevationArray(1,1)) && (n ~= AngleClosestIndex_MVA)
    && (DataMatrix(n,5) == GalileoSelection_MVA))
        l3_MVA = cos(AngleBetweenVectors([S1_MVA(2), S1_MVA(3),
S1_MVA(4)], [DataMatrix(n, 2), DataMatrix(n, 3), DataMatrix(n, 4)]));

        m3_MVA =
(cos(AngleBetweenVectors([DataMatrix(AngleClosestIndex_MVA, 2),
DataMatrix(AngleClosestIndex_MVA, 3), DataMatrix(AngleClosestIndex_MVA,
4)], [DataMatrix(n, 2), DataMatrix(n, 3), DataMatrix(n, 4)])) -
(l2_MVA*l3_MVA))/m2_MVA;

        n3_MVA = sqrt(1 - l3_MVA^2 - m3_MVA^2);

        Vol_MVA = (sqrt(2*(1 - l2_MVA)*(1 + l3_MVA)*(1 - l2_MVA*l3_MVA
- m2_MVA*m3_MVA)) + abs(m2_MVA*n3_MVA))*(1 - l3_MVA)/6;

        if (Vol_MVA > VolMax3_MVA)
            VolMax3_MVA = Vol_MVA;
            VolMaxIndex3_MVA = n;
        end
    end
end

end

S3_MVA = DataMatrix(VolMaxIndex3_MVA,:);

% Step 4: Maximize volume
VolMax4_MVA = 0;
for n = 1:size(DataMatrix, 1)

```



```

        if ((n ~= MaxElevationArray(1,1)) && (n ~= AngleClosestIndex_MVA)
&& (n ~= VolMaxIndex3_MVA) && (DataMatrix(n,5) == GalileoSelection_MVA))
            Vol_MVA = abs(det([S1_MVA(2)-DataMatrix(n,2), S1_MVA(3)-
DataMatrix(n,3), S1_MVA(4)-DataMatrix(n,4); ...
S2_MVA(2)-DataMatrix(n,2), S2_MVA(3)-DataMatrix(n,3),
S2_MVA(4)-DataMatrix(n,4); ...
S3_MVA(2)-DataMatrix(n,2), S3_MVA(3)-DataMatrix(n,3),
S3_MVA(4)-DataMatrix(n,4)]))/6;

            if (Vol_MVA > VolMax4_MVA)
                VolMax4_MVA = Vol_MVA;
                VolMaxIndex4_MVA = n;
            end
        end

end

S4_MVA = DataMatrix(VolMaxIndex4_MVA,:);

% Array of indexes
SelectedSatellite_MVA = [MaxElevationArray(1,1);
    AngleClosestIndex_MVA;
    VolMaxIndex3_MVA;
    VolMaxIndex4_MVA];

% Build LS matrix
LSMat = [S1_MVA;
    S2_MVA;
    S3_MVA;
    S4_MVA];

% Build RangeEst_MVA matrix
RangeEst_MVA = [RangeEst(MaxElevationArray(1,1));
    RangeEst(AngleClosestIndex_MVA);
    RangeEst(VolMaxIndex3_MVA);
    RangeEst(VolMaxIndex4_MVA)];

% Calculate PDOP
PDOP_MVA = 0;

```

```

PDOP_MVA = PDOPvar(LSMat(:, 2:5), RecPosEst, RangeEst_MVA);

% Calculate GDOP
GDOP = GDOPvar(LSMat(:, 2:5), RecPosEst, RangeEst_MVA);

%Call LS function
% RangeEst de los elegidos
LSMat = [LSMat(:,2:5) LSMat(:,1)];
dX = LeastSquaresV2(LSMat, RangeEst_MVA, RecPosEst);

end

```

8.4.4 Dartboard DOP algorithm

```

%%%%%%%%%% Dartboard Algorithm alternative %%%%%%%%%%
% Build a dartboard using the GNSS skyplot (azimuth and elevation)
% Divide the dartboard into 12 zones and select the satellite closest
to
% the centre of each zone

function [dX, selectedDart, PDOP_dartboard, GDOP] =
DartboardDOPalt1605(DataMatrix, RangeEst, RecPosEst,
SatelliteElevation, SatelliteAzimuth)

% Elevation and azimuth in rads
E = SatelliteElevation*pi/180;
Azimuth = SatelliteAzimuth*pi/180;

elevationStep = pi/6;
azimuthStep = pi/2;

selectedDartIndex = 1;
selectedDart = zeros(1,12);
ClosestDartAzimuth = [1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000
1000 1000 1000 1000];
ClosestDartElevation = [1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000
1000 1000 1000 1000];

counter = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1];
Zones = zeros(12, []);
for n = 1:size(DataMatrix,1)
    if E(n) < pi/6
        if (Azimuth(n) >= 0) && (Azimuth(n) < pi/2)
            Zones(1, counter(1)) = n;
            counter(1) = counter(1) + 1;
        elseif (Azimuth(n) >= pi/2) && (Azimuth(n) < pi)
            Zones(2, counter(2)) = n;
            counter(2) = counter(2) + 1;
        elseif (Azimuth(n) >= pi) && (Azimuth(n) < 3*pi/2)
            Zones(3, counter(3)) = n;
            counter(3) = counter(3) + 1;
        elseif (Azimuth(n) >= 3*pi/2)

```

```

        Zones(4, counter(4)) = n;
        counter(4) = counter(4) + 1;
    end
elseif (E(n) >= pi/6) && (E(n) < pi/3)
    if (Azimuth(n) >= pi/4) && (Azimuth(n) < 3*pi/4)
        Zones(5, counter(5)) = n;
        counter(5) = counter(5) + 1;
    elseif (Azimuth(n) >= 3*pi/4) && (Azimuth(n) < 5*pi/2)
        Zones(6, counter(6)) = n;
        counter(6) = counter(6) + 1;
    elseif (Azimuth(n) >= 5*pi/2) && (Azimuth(n) < 7*pi/4)
        Zones(7, counter(7)) = n;
        counter(7) = counter(7) + 1;
    elseif (Azimuth(n) >= 7*pi/4)
        Zones(8, counter(8)) = n;
        counter(8) = counter(8) + 1;
    end
elseif E(n) >= pi/3
    if (Azimuth(n) >= 0) && (Azimuth(n) < pi/2)
        Zones(9, counter(9)) = n;
        counter(9) = counter(9) + 1;
    elseif (Azimuth(n) >= pi/2) && (Azimuth(n) < pi)
        Zones(10, counter(10)) = n;
        counter(10) = counter(10) + 1;
    elseif (Azimuth(n) >= pi) && (Azimuth(n) < 3*pi/2)
        Zones(11, counter(11)) = n;
        counter(11) = counter(11) + 1;
    elseif (Azimuth(n) >= 3*pi/2)
        Zones(12, counter(12)) = n;
        counter(12) = counter(12) + 1;
    end
end
end

% Calculate center of zones
CenterDartAzimuth(1) = azimuthStep/2;
CenterDartAzimuth(5) = azimuthStep;
CenterDartAzimuth(9) = azimuthStep/2;

CenterDartAzimuth(2) = azimuthStep + azimuthStep/2;

```

```

CenterDartAzimuth(6) = 2*azimuthStep;
CenterDartAzimuth(10) = azimuthStep + azimuthStep/2;

CenterDartAzimuth(3) = 2*azimuthStep + azimuthStep/2;
CenterDartAzimuth(7) = 3*azimuthStep;
CenterDartAzimuth(11) = 2*azimuthStep + azimuthStep/2;

CenterDartAzimuth(4) = 3*azimuthStep + azimuthStep/2;
CenterDartAzimuth(8) = 0;
CenterDartAzimuth(12) = 3*azimuthStep + azimuthStep/2;

CenterDartElevation(1) = elevationStep/2;
CenterDartElevation(2) = elevationStep/2;
CenterDartElevation(3) = elevationStep/2;
CenterDartElevation(4) = elevationStep/2;

CenterDartElevation(5) = elevationStep + elevationStep/2;
CenterDartElevation(6) = elevationStep + elevationStep/2;
CenterDartElevation(7) = elevationStep + elevationStep/2;
CenterDartElevation(8) = elevationStep + elevationStep/2;

CenterDartElevation(9) = 2*elevationStep + elevationStep/2;
CenterDartElevation(10) = 2*elevationStep + elevationStep/2;
CenterDartElevation(11) = 2*elevationStep + elevationStep/2;
CenterDartElevation(12) = 2*elevationStep + elevationStep/2;

% Set satellites into zones

for n = 1:size(Zones,1)

    for m = 1:size(Zones,2)

        if (Zones(n,m) ~= 0)

            if (sqrt((ClosestDartAzimuth(n)-CenterDartAzimuth(n))^2 +
(ClosestDartElevation(n)-CenterDartElevation(n))^2) ...
                > sqrt((Azimuth(Zones(n,m))-
CenterDartAzimuth(n))^2 + (E(Zones(n,m))-CenterDartElevation(n))^2))

```



```

        end
    end
end

% Count number of selected satellites
selectedDartNum = [];
for n = 1:size(selectedDart,2)
    if (selectedDart(n) ~= 0)
        selectedDartNum = [selectedDartNum, n];
    end
end

LSMat_dartboard = [];
RangeEst_dartboard = [];
for n = selectedDartNum
    % Build LS function
    LSMat_dartboard = [LSMat_dartboard ; [DataMatrix(selectedDart(n),
2:5) DataMatrix(selectedDart(n), 1)]];

    % Build RangeEst_dartboard array
    RangeEst_dartboard = [RangeEst_dartboard ;
RangeEst(selectedDart(n))];
end

% Calculate PDOP
PDOP_dartboard = PDOPvar(LSMat_dartboard(:,1:4), RecPosEst,
RangeEst_dartboard);

% Calculate GDOP
GDOP = GDOPvar(LSMat_dartboard(:,1:4), RecPosEst, RangeEst_dartboard);

%Call LS function
% RangeEst de los elegidos
dX = LeastSquaresV2(LSMat_dartboard, RangeEst_dartboard, RecPosEst);

end

```

8.4.5 WPDOP algorithm

```

%%%%%%%%%%%% WPDOP Algorithm %%%%%%%%%%
% Algorithm that takes into account the weights of the different errors
in
% the pseudoranges
% Set weights and calculate contribution functions to exclude satellites
% from the list
% Iterate until threshold value sufficies

function [dX, PDOP_WPDOP, NumSatReq, GDOP, SeletedectedSatellites] =
WPDOP(DataMatrix, RangeEst, RecPosEst, GPSArrayN, GalileoArrayN,
W_WPDOP)

SatList = DataMatrix(:, 2:5);
PseudorangeEstList = DataMatrix(:, 1);
RangeEstList = RangeEst';

% Obtain number of satellites in the data matrix
[DataMatrixRows, DataMatrixColumns] = size(DataMatrix);
SatelliteAvailable = (1:DataMatrixRows)';

% Check for single member of constellation
if (size(GPSArrayN) < 2)
    ai_WPDOP = [DataMatrix(GPSArrayN(1),2) , DataMatrix(GPSArrayN(1),3)
, DataMatrix(GPSArrayN(1),4) , DataMatrix(GPSArrayN(1),5)];
    wi_WPDOP = W_WPDOP(GPSArrayN(1));

% Eliminate from satellite list
for n = 1:DataMatrixRows
    if (DataMatrix(n, 2) == ai_WPDOP(1))
        SatList = [SatList(1:(n-1),:); SatList((n+1):end,:)];
        RangeEstList = [RangeEstList(1:(n-1));
RangeEstList((n+1):end)];
        PseudorangeEstList = [PseudorangeEstList(1:(n-1));
PseudorangeEstList((n+1):end)];
        SatelliteAvailable = [SatelliteAvailable(1:(n-1));
SatelliteAvailable((n+1):end)];
    end
end

```



```

elseif (size(GalileoArrayN) < 2)
    ai_WPDOP = [DataMatrix(GalileoArrayN(1),2) ,
DataMatrix(GalileoArrayN(1),3) , DataMatrix(GalileoArrayN(1),4) ,
DataMatrix(GalileoArrayN(1),5)];
    wi_WPDOP = W_WPDOP(GalileoArrayN(1));

    % Eliminate from satellite list
    for n = 1:DataMatrixRows
        if (DataMatrix(n, 2) == ai_WPDOP(1))
            SatList = [SatList(1:(n-1),:); SatList((n+1):end,:)];
            RangeEstList = [RangeEstList(1:(n-1));
RangeEstList((n+1):end)];
            PseudorangeEstList = [PseudorangeEstList(1:(n-1));
PseudorangeEstList((n+1):end)];
            SatelliteAvailable = [SatelliteAvailable(1:(n-1));
SatelliteAvailable((n+1):end)];
        end
    end

end

% Calculate matrix M
A_WPDOP = [(RecPosEst(1) - SatList(1,1))/RangeEstList(1), (RecPosEst(2)
- SatList(1,2))/RangeEstList(1), (RecPosEst(3)
- SatList(1,3))/RangeEstList(1), 1, SatList(1, 4)];
for n = 2:DataMatrixRows
    A_WPDOP = [A_WPDOP; (RecPosEst(1)
SatList(n,1))/RangeEstList(n), (RecPosEst(2)
SatList(n,2))/RangeEstList(n), (RecPosEst(3)
SatList(n,3))/RangeEstList(n), 1, SatList(n, 4)];
end

M_WPDOP = A_WPDOP'*W_WPDOP*A_WPDOP;

% Obtain WPDOP value
diagonalDOP_WPDOP = diag(inv(M_WPDOP));
valueWPDOP = sqrt(diagonalDOP_WPDOP(1) + diagonalDOP_WPDOP(2) +
diagonalDOP_WPDOP(3));

```

```

Threshold = 12;
Jmin_WPDOP = 100000;
JminIndex_WPDOP = 0;

% Calculate all satellite's contribution
while (valueWPDOP <= Threshold)
    for n = 1:size(A_WPDOP,1)

        Aaux_WPDOP = [A_WPDOP(1:n-1,:); A_WPDOP(n+1:end,:)];
        Waux_WPDOP = [W_WPDOP(1:n-1,:); W_WPDOP(n+1:end,:)];
        Waux_WPDOP = [Waux_WPDOP(:,1:n-1), Waux_WPDOP(:,n+1:end)];

        aiJi_WPDOP = A_WPDOP(n, :);
        wiJi_WPDOP = W_WPDOP(n);
        Miaux_WPDOP = Aaux_WPDOP'*Waux_WPDOP*Aaux_WPDOP; % Mn-1, i
        Maux_WPDOP = Miaux_WPDOP +
        (wiJi_WPDOP*(aiJi_WPDOP')*aiJi_WPDOP); % Mn

        % Lambda i
        lambdai_WPDOP = wiJi_WPDOP +
        (wiJi_WPDOP^2)*aiJi_WPDOP*Miaux_WPDOP*(aiJi_WPDOP');

        % Contribution function value
        Ji_WPDOP =
        lambdai_WPDOP*aiJi_WPDOP*(inv(Maux_WPDOP)^2)*(aiJi_WPDOP');

        if (Jmin_WPDOP > Ji_WPDOP)
            Jmin_WPDOP = Ji_WPDOP;
            JminIndex_WPDOP = n;
        end

    end

end

% Remove satellite with smallest contribution
A_WPDOP = [A_WPDOP(1:JminIndex_WPDOP-1,:);
A_WPDOP(JminIndex_WPDOP+1:end,:)];

```

```

        W_WPDOP          =          [W_WPDOP(1:JminIndex_WPDOP-1,:)          ;
W_WPDOP(JminIndex_WPDOP+1:end,:)] ;
        W_WPDOP          =          [W_WPDOP(:,1:JminIndex_WPDOP-1)          ,
W_WPDOP(:,JminIndex_WPDOP+1:end)] ;

        aiJ_WPDOP = A_WPDOP(JminIndex_WPDOP,:);
        wiJ_WPDOP = W_WPDOP(JminIndex_WPDOP,:);

        RangeEstList          =          [RangeEstList(1:(JminIndex_WPDOP-1));
RangeEstList((JminIndex_WPDOP+1):end)] ;
        PseudorangeEstList    =    [PseudorangeEstList(1:(JminIndex_WPDOP-1));
PseudorangeEstList((JminIndex_WPDOP+1):end)] ;
        SatelliteAvailable    =    [SatelliteAvailable(1:(JminIndex_WPDOP-1));
SatelliteAvailable((JminIndex_WPDOP+1):end)] ;

% Recalculate matrix M
M_WPDOP = A_WPDOP'*W_WPDOP*A_WPDOP;

% Recalculate WPDOP value
diagonalDOP_WPDOP = diag(inv(M_WPDOP));
valueWPDOP = sqrt(diagonalDOP_WPDOP(1) + diagonalDOP_WPDOP(2) +
diagonalDOP_WPDOP(3));

% Count remaining satellites
iGPS_WPDOP = 1;
iGalileo_WPDOP = 1;

for n = 1:size(A_WPDOP,1)
    if (A_WPDOP(n, 5) == 0)
        GPSArrayN_WPDOP(iGPS_WPDOP) = n;
        GPSArray_WPDOP(iGPS_WPDOP,:) = A_WPDOP(n, 1:4);
        iGPS_WPDOP = iGPS_WPDOP + 1;
    else
        GalileoArrayN_WPDOP(iGalileo_WPDOP) = n;
        GalileoArray_WPDOP(iGalileo_WPDOP,:) = A_WPDOP(n, 1:4);
        iGalileo_WPDOP = iGalileo_WPDOP + 1;
    end
end
end

```

```

% Check for single member of constellation
if (size(GPSArrayN_WPDOP) < 2)
    ai_WPDOP      =      [A_WPDOP(GPSArrayN_WPDOP(1),2)      ,
A_WPDOP(GPSArrayN_WPDOP(1),3)      ,      A_WPDOP(GPSArrayN_WPDOP(1),4)      ,
A_WPDOP(GPSArrayN_WPDOP(1),5)];
    wi_WPDOP = W_WPDOP(GPSArrayN_WPDOP(1));

% Eliminate from satellite list
for n = 1:size(A_WPDOP,1)
    if (A_WPDOP(n, 1) == ai_WPDOP(1))
        A_WPDOP = [A_WPDOP(1:n-1,:) ; A_WPDOP(n+1:end,:)];
        W_WPDOP = [W_WPDOP(1:n-1,:) ; W_WPDOP(n+1:end,:)];
        RangeEstList      =      [RangeEstList(1:(n-1));
RangeEstList((n+1):end)];
        PseudorangeEstList      =      [PseudorangeEstList(1:(n-1));
PseudorangeEstList((n+1):end)];
        SatelliteAvailable      =      [SatelliteAvailable(1:(n-1));
SatelliteAvailable((n+1):end)];
    end
end

elseif (size(GalileoArrayN_WPDOP) < 2)
    ai_WPDOP      =      [A_WPDOP(GalileoArrayN_WPDOP(1),2)      ,
A_WPDOP(GalileoArrayN_WPDOP(1),3) , A_WPDOP(GalileoArrayN_WPDOP(1),4) ,
A_WPDOP(GalileoArrayN_WPDOP(1),5)];
    wi_WPDOP = W_WPDOP(GalileoArrayN_WPDOP(1));

% Eliminate from satellite list
for n = 1:size(A_WPDOP,1)
    if (A_WPDOP(n, 2) == ai_WPDOP(1))
        A_WPDOP = [A_WPDOP(1:n-1,:) ; A_WPDOP(n+1:end,:)];
        W_WPDOP = [W_WPDOP(1:n-1,:) ; W_WPDOP(n+1:end,:)];
        RangeEstList      =      [RangeEstList(1:(n-1));
RangeEstList((n+1):end)];
        PseudorangeEstList      =      [PseudorangeEstList(1:(n-1));
PseudorangeEstList((n+1):end)];
        SatelliteAvailable      =      [SatelliteAvailable(1:(n-1));
SatelliteAvailable((n+1):end)];
    end
end

```

```

end

% Recalculate matrix M
M_WPDOP = A_WPDOP'*W_WPDOP*A_WPDOP;

end

NumSatReq = size(A_WPDOP,1);

% Calculate PDOP
covOverUEREmat = A_WPDOP.'*A_WPDOP;
covOverUEREmat = pinv(covOverUEREmat);
PDOP_WPDOP = sqrt(covOverUEREmat(1,1) + covOverUEREmat(2,2) +
covOverUEREmat(3,3));
GDOP = sqrt(covOverUEREmat(1,1) + covOverUEREmat(2,2) +
covOverUEREmat(3,3) + covOverUEREmat(4,4) + covOverUEREmat(5,5));

% Calculate dX
b = PseudorangeEstList - RangeEstList;
dX = inv(A_WPDOP.'*A_WPDOP)*A_WPDOP.'*b;

SeletectedSatellites = SatelliteAvailable;

end

```

8.4.6 EPR algorithm

```

##### EPR Algorithm #####
% Combination of different algorithms to reduce computation but achieve
% quasi-optimal satellite selection
% 1: HEA
% 2: Dartboard
% 3: OSSA

function [dX, minimumPDOP, GDOP, SelectedSatellites] = EPR(DataMatrix,
RangeEst, RecPosEst, GPSArrayN, GalileoArrayN, SatelliteElevation,
SatelliteAzimuth)

NumberSatellites = size(DataMatrix, 1);
SatelliteAvailable = (1:NumberSatellites)';

% MaxElevation for GPS
[dX_HEA_GPS, MaxElevationArrayGPS] = HEA(DataMatrix(GPSArrayN, :),
RangeEst(GPSArrayN), RecPosEst, SatelliteElevation);

% MVA GPS
[dX_MVA_GPS, matMVA_GPS, SelectedSatellites_MVA_GPS] =
MVA(DataMatrix(GPSArrayN, :), RangeEst(GPSArrayN), RecPosEst,
MaxElevationArrayGPS);
SelectedSatellites_MVA_GPS = GPSArrayN(SelectedSatellites_MVA_GPS)';

% MaxElevation for Galileo
[dX_HEA_Galileo, MaxElevationArrayGalileo] =
HEA(DataMatrix(GalileoArrayN, :), RangeEst(GalileoArrayN), RecPosEst,
SatelliteElevation);

% MVA Galileo
[dX_MVA_Galileo, matMVA_Galileo, SelectedSatellites_MVA_Galileo] =
MVA(DataMatrix(GalileoArrayN, :), RangeEst(GalileoArrayN), RecPosEst,
MaxElevationArrayGalileo);
SelectedSatellites_MVA_Galileo =
GalileoArrayN(SelectedSatellites_MVA_Galileo)';

% Build new data matrix
DataMatrix2 = [];

```

```

RangeEst2 = [];
DataMatrix3 = [];
RangeEst3 = [];
SatelliteElevation2 = [];
SatelliteAzimuth2 = [];
SatelliteElevation3 = [];
SatelliteAzimuth3 = [];
SatellitesAvailable2 = [];
SatellitesAvailable3 = [];

for n = 1:size(DataMatrix, 1)
    if (n ~= SelectedSatellites_MVA_GPS(1) && n ~=
SelectedSatellites_MVA_GPS(2) ...
        && n ~= SelectedSatellites_MVA_GPS(3) && n ~=
SelectedSatellites_MVA_GPS(4) ...
        && n ~= SelectedSatellites_MVA_Galileo(1) && n ~=
SelectedSatellites_MVA_Galileo(2) ...
        && n ~= SelectedSatellites_MVA_Galileo(3) && n ~=
SelectedSatellites_MVA_Galileo(4))

        SatellitesAvailable2 = [SatellitesAvailable2; n];

    if (n == 1)
        DataMatrix2 = [DataMatrix(n, :)];
        RangeEst2 = [RangeEst(n)];
        SatelliteElevation2 = [SatelliteElevation(n)];
        SatelliteAzimuth2 = [SatelliteAzimuth(n)];
    else
        DataMatrix2 = [DataMatrix2; DataMatrix(n, :)];
        RangeEst2 = [RangeEst2; RangeEst(n)];
        SatelliteElevation2 = [SatelliteElevation2;
SatelliteElevation(n)];
        SatelliteAzimuth2 = [SatelliteAzimuth2;
SatelliteAzimuth(n)];
    end

end

end

% Max elevation

```

```

[dX_HEA, MaxElevationArray_HEA] = HEA(DataMatrix2, RangeEst2, RecPosEst,
SatelliteElevation2);

SelectedSatellite_HEA                                     =
SatellitesAvailable2(MaxElevationArray_HEA(1,1));
SatellitesAvailable3                                     =
[SatellitesAvailable2(1:MaxElevationArray_HEA(1,1)-1);
SatellitesAvailable2(MaxElevationArray_HEA(1,1)+1:end)];

% Build new data matrix
for n = 1:size(DataMatrix2, 1)
    if (n ~= MaxElevationArray_HEA(1,1))

        if (n == 1)
            DataMatrix3 = [DataMatrix2(n, :)];
            RangeEst3 = [RangeEst2(n)];
            SatelliteElevation3 = [SatelliteElevation2(n)];
            SatelliteAzimuth3 = [SatelliteAzimuth2(n)];
        else
            DataMatrix3 = [DataMatrix3; DataMatrix2(n, :)];
            RangeEst3 = [RangeEst3; RangeEst2(n)];
            SatelliteElevation3 = [SatelliteElevation3;
SatelliteElevation2(n)];
            SatelliteAzimuth3 = [SatelliteAzimuth3;
SatelliteAzimuth2(n)];
        end

    end
end

% Dartboard DOP
[dX_dartboard, selectedDart] = DartboardDOPalt1605(DataMatrix3,
RangeEst3, RecPosEst, SatelliteElevation3, SatelliteAzimuth3);

% Selected satellites from previous steps
SelectedSatellitesMat = [DataMatrix(SelectedSatellites_MVA_GPS, 2:5);
    DataMatrix(SelectedSatellites_MVA_Galileo, 2:5);
    DataMatrix2(MaxElevationArray_HEA(1,1), 2:5)];
RangeEstSelectedSatellites = [RangeEst(SelectedSatellites_MVA_GPS)'];

```



```

RangeEst(SelectedSatellites_MVA_Galileo)';
RangeEst2(MaxElevationArray_HEA(1,1));

SelectedSatellitesMat5 = [DataMatrix(SelectedSatellites_MVA_GPS, 2:5),
DataMatrix(SelectedSatellites_MVA_GPS, 1);
    DataMatrix(SelectedSatellites_MVA_Galileo, 2:5),
DataMatrix(SelectedSatellites_MVA_Galileo, 1);
    DataMatrix2(MaxElevationArray_HEA(1,1), 2:5),
DataMatrix2(MaxElevationArray_HEA(1,1), 1)];

if ((selectedDart(5) ~= 0) && (selectedDart(6) ~= 0) && (selectedDart(7)
~= 0))

    % OSSA of dartboard selection 1 (5,6,7)
    Dart1mat = [DataMatrix3(selectedDart(5), 2:5);
        DataMatrix3(selectedDart(6), 2:5);
        DataMatrix3(selectedDart(7), 2:5);];
    Dart1RangeEst = [RangeEst3(selectedDart(5));
        RangeEst3(selectedDart(6));
        RangeEst3(selectedDart(7))];

    Dart1mat5 = [[DataMatrix3(selectedDart(5), 2:5),
DataMatrix3(selectedDart(5), 1)];
        [DataMatrix3(selectedDart(6), 2:5),
DataMatrix3(selectedDart(6), 1)];
        [DataMatrix3(selectedDart(7), 2:5),
DataMatrix3(selectedDart(7), 1)]];

    PDOP1_OSSA = PDOPvar([SelectedSatellitesMat; Dart1mat], RecPosEst,
[RangeEstSelectedSatellites; Dart1RangeEst]);
    dX1_OSSA = LeastSquaresV2([SelectedSatellitesMat5; Dart1mat5],
[RangeEstSelectedSatellites; Dart1RangeEst], RecPosEst);

else

    PDOP1_OSSA = 100000;

end

```

```

if ((selectedDart(8) ~= 0) && (selectedDart(6) ~= 0) && (selectedDart(7)
~= 0))

    % OSSA of dartboard selection 2 (8,6,7)
    Dart2mat = [DataMatrix3(selectedDart(8), 2:5);
                DataMatrix3(selectedDart(6), 2:5);
                DataMatrix3(selectedDart(7), 2:5)];
    Dart2RangeEst = [RangeEst3(selectedDart(8));
                    RangeEst3(selectedDart(6));
                    RangeEst3(selectedDart(7))];

    Dart2mat5 = [DataMatrix3(selectedDart(8), 2:5),
DataMatrix3(selectedDart(8), 1);
                DataMatrix3(selectedDart(6), 2:5), DataMatrix3(selectedDart(6),
1);
                DataMatrix3(selectedDart(7), 2:5), DataMatrix3(selectedDart(7),
1)];

    PDOP2_OSSA = PDOPvar([SelectedSatellitesMat; Dart2mat], RecPosEst,
[RangeEstSelectedSatellites; Dart2RangeEst]);
    dx2_OSSA = LeastSquaresV2([SelectedSatellitesMat5; Dart2mat5],
[RangeEstSelectedSatellites; Dart2RangeEst], RecPosEst);

else

    PDOP2_OSSA = 100000;

end

if ((selectedDart(5) ~= 0) && (selectedDart(8) ~= 0) && (selectedDart(7)
~= 0))

    % OSSA of dartboard selection 3 (5,8,7)
    Dart3mat = [DataMatrix3(selectedDart(5), 2:5);
                DataMatrix3(selectedDart(8), 2:5);
                DataMatrix3(selectedDart(7), 2:5)];
    Dart3RangeEst = [RangeEst3(selectedDart(5));
                    RangeEst3(selectedDart(8));
                    RangeEst3(selectedDart(7))];

```

```

        Dart3mat5          =          [DataMatrix3(selectedDart(5),          2:5),
DataMatrix3(selectedDart(5), 1);
        DataMatrix3(selectedDart(8), 2:5), DataMatrix3(selectedDart(8),
1);
        DataMatrix3(selectedDart(7), 2:5), DataMatrix3(selectedDart(7),
1)];

        PDOP3_OSSA = PDOPvar([SelectedSatellitesMat; Dart3mat], RecPosEst,
[RangeEstSelectedSatellites; Dart3RangeEst]);
        dX3_OSSA  =  LeastSquaresV2([SelectedSatellitesMat5;  Dart3mat5],
[RangeEstSelectedSatellites; Dart3RangeEst], RecPosEst);

else

        PDOP3_OSSA = 100000;

end

if ((selectedDart(5) ~= 0) && (selectedDart(6) ~= 0) && (selectedDart(8)
~= 0))

    % OSSA of dartboard selection 4 (5,6,8)
    Dart4mat = [DataMatrix3(selectedDart(5), 2:5);
        DataMatrix3(selectedDart(6), 2:5);
        DataMatrix3(selectedDart(8), 2:5)];
    Dart4RangeEst = [RangeEst3(selectedDart(5));
        RangeEst3(selectedDart(6));
        RangeEst3(selectedDart(8))];

        Dart4mat5          =          [DataMatrix3(selectedDart(5),          2:5),
DataMatrix3(selectedDart(5), 1);
        DataMatrix3(selectedDart(6), 2:5), DataMatrix3(selectedDart(6),
1);
        DataMatrix3(selectedDart(8), 2:5), DataMatrix3(selectedDart(8),
1)];

        PDOP4_OSSA = PDOPvar([SelectedSatellitesMat; Dart4mat], RecPosEst,
[RangeEstSelectedSatellites; Dart4RangeEst]);

```

```

        dX4_OSSA = LeastSquaresV2([SelectedSatellitesMat5; Dart4mat5],
[RangeEstSelectedSatellites; Dart4RangeEst], RecPosEst);

else

    PDOP4_OSSA = 100000;

end

% Select minimum PDOP
minimumPDOP = min([PDOP1_OSSA, PDOP2_OSSA, PDOP3_OSSA, PDOP4_OSSA]);

if (minimumPDOP == PDOP1_OSSA)
    dX = dX1_OSSA;
    GDOP = GDOPvar([SelectedSatellitesMat; Dart1mat], RecPosEst,
[RangeEstSelectedSatellites; Dart1RangeEst]);
    SelectedSatellites_OSSA = [SatellitesAvailable3(selectedDart(5));
SatellitesAvailable3(selectedDart(6));
SatellitesAvailable3(selectedDart(7))];
    SelectedSatellites = [SelectedSatellites_MVA_GPS;
SelectedSatellites_MVA_Galileo; SelectedSatellite_HEA;
SelectedSatellites_OSSA];

elseif (minimumPDOP == PDOP2_OSSA)
    dX = dX2_OSSA;
    GDOP = GDOPvar([SelectedSatellitesMat; Dart2mat], RecPosEst,
[RangeEstSelectedSatellites; Dart2RangeEst]);
    SelectedSatellites_OSSA = [SatellitesAvailable3(selectedDart(8));
SatellitesAvailable3(selectedDart(6));
SatellitesAvailable3(selectedDart(7))];
    SelectedSatellites = [SelectedSatellites_MVA_GPS;
SelectedSatellites_MVA_Galileo; SelectedSatellite_HEA;
SelectedSatellites_OSSA];

elseif (minimumPDOP == PDOP3_OSSA)
    dX = dX3_OSSA;
    GDOP = GDOPvar([SelectedSatellitesMat; Dart3mat], RecPosEst,
[RangeEstSelectedSatellites; Dart3RangeEst]);

```

```

        SelectedSatellites_OSSA = [SatellitesAvailable3(selectedDart(5));
SatellitesAvailable3(selectedDart(8));
SatellitesAvailable3(selectedDart(7))];

        SelectedSatellites          =          [SelectedSatellites_MVA_GPS;
SelectedSatellites_MVA_Galileo;          SelectedSatellite_HEA;
SelectedSatellites_OSSA];

```

```

elseif (minimumPDOP == PDOP4_OSSA)
    dX = dX4_OSSA;
    GDOP = GDOPvar([SelectedSatellitesMat; Dart4mat], RecPosEst,
[RangeEstSelectedSatellites; Dart4RangeEst]);
    SelectedSatellites_OSSA = [SatellitesAvailable3(selectedDart(5));
SatellitesAvailable3(selectedDart(6));
SatellitesAvailable3(selectedDart(8))];

    SelectedSatellites          =          [SelectedSatellites_MVA_GPS;
SelectedSatellites_MVA_Galileo;          SelectedSatellite_HEA;
SelectedSatellites_OSSA];

```

```

end

```

```

end

```

8.5 Código fuente algoritmos detección de medidas erróneas

8.5.1 RAIM convencional

```
%{
Matlab function for the conventional RAIM technique
%}

function [xOUT, GT, LT, HPL, VPL] = RAIMconvencionalV3(Rho, RangeEst,
PosEst, SatelliteMatrix, WeightMatrix, Pnd, Per)

% Declare constants
SigLevel = 1.7*10^-5;

% Number of satellites
SatelliteNumber = size(SatelliteMatrix, 1);

LT = zeros(1, 24);

% Build matrix H
H = [(SatelliteMatrix(1, 1)-PosEst(1))/RangeEst(1) (SatelliteMatrix(1,
2)-PosEst(2))/RangeEst(1) (SatelliteMatrix(1, 3)-PosEst(3))/RangeEst(1)
1 SatelliteMatrix(1,4)];

for n = 2:SatelliteNumber
    H = [H; (SatelliteMatrix(n, 1)-PosEst(1))/RangeEst(n)
(SatelliteMatrix(n, 2)-PosEst(2))/RangeEst(n) (SatelliteMatrix(n, 3)-
PosEst(3))/RangeEst(n) 1 SatelliteMatrix(n,4)];
end

% Build measurement vector z
z = -(Rho - RangeEst);

% Solve for state vector x with WLS
x = pinv(H'*WeightMatrix*H)*H'*WeightMatrix*z;

% Calculate residuals r
r = z - H*x;
```

```

% Compute threshold Tg (Chi-Squared distribution with N-5 DoF)
%Tg = ChiSquared(SatelliteNumber - 5, 1 - SigLevel);
Tg = chi2inv(SigLevel, SatelliteNumber - 5);

D = 1000;

SatelliteExcluded = 0;

while (D > Tg)

    Dmin = inf;
    DminIndex = 0;

    % Solve for state vector x with WLS
    x = pinv(H'*WeightMatrix*H)*H'*WeightMatrix*z;

    % Calculate residuals r
    r = z - H*x;

    % Compute test statistic D
    D = r'*WeightMatrix*r;

    % Global test
    if D > Tg % Reject H0
        GT = 1;

        % Local test
        for n = 1:SatelliteNumber
            W_i = WeightMatrix;
            W_i(n,n) = 0;
            x_i = pinv(H'*W_i*H)*H'*W_i*z;
            r_i = z - H*x_i;
            D_i = r_i'*W_i*r_i;

            if D_i < Dmin
                Dmin = D_i;
                DminIndex = n;
            end
        end
    end
end

```

```

        end

    end

    for n = 1:SatelliteNumber
        if n == DminIndex
            LT(n) = 1;
        end
    end

    % Update
    if (WeightMatrix(DminIndex, DminIndex) ~= 0)
        WeightMatrix(DminIndex, DminIndex) = 0;
        D = Dmin;
        SatelliteExcluded = SatelliteExcluded + 1;
    else
        D = Dmin;
        break;
    end

else

    GT = 0;

end

if (SatelliteNumber - SatelliteExcluded - 6) == 1
    break;
end

end

H2 = H;
W2 = WeightMatrix;
for n = -SatelliteNumber:-1
    if (LT(-n) == 1)
        H2 = [H2(1:-n-1,:); H2(-n+1:end,:)];
        W2 = [W2(1:-n-1,:); W2(-n+1:end,:)];
        W2 = [W2(:,1:-n-1), W2(:, -n+1:end)];
    end
end

```



```

    end
end

% Solve for state vector x with WLS
x2 = pinv(H'*WeightMatrix*H)*H'*WeightMatrix*z;
xOUT = x2(1:3) + PosEst;

% Compute protection levels
[HPL, VPL] = ProtectionLevel(Pnd, Per, Tg, H2, W2, SatelliteNumber-
SatelliteExcluded);

end

```

8.5.2 Nivel de protección RAIM convencional

```
%{
Matlab function for the protection level computation
%}

function [HPL, VPL] = ProtectionLevel(Pnd, Per, T, H, W, N)

% Declare
HSLOPEmax = 0;
VSLOPEmax = 0;

% Compute standard deviations
A = pinv(H'*W*H);
Sigma = sqrt(max(eig(A(1:3,1:3)))));
SigmaH = sqrt(max(eig(A(1:2, 1:2)))));
SigmaV = sqrt(eig(A(3,3))));

% Compute pbias
pbias = T + Sigma*erfcinv(Pnd);

% Compute max slopes
S = eye(N) - H*A*H'*W;

A = pinv(H'*W*H)*H'*W;

for n = 1:N
    if abs(S(n,n)) > 0.01
        HSLOPE(n) = sqrt((A(1,n)^2 + A(1,n)^2)/S(n,n));
        if HSLOPE(n) > HSLOPEmax
            HSLOPEmax = HSLOPE(n);
        end

        VSLOPE(n) = abs(A(3, n))/sqrt(S(n,n));
        if VSLOPE(n) > VSLOPEmax
            VSLOPEmax = VSLOPE(n);
        end
    end
end

end
```

```
% Compute protection levels
HPL = HSLOPEmax*pbias + SigmaH*erfcinv(1-Per);
VPL = VSLOPEmax*pbias + SigmaV*erfcinv(1-Per);

end
```

8.5.3 RAIM basado en separación de soluciones

```
%{
Matlab function for the solution separation RAIM technique
%}

function [xOUT, yH, yV, HPL, VPL] = RAIMSolSepV2(Rho, RangeEst, PosEst,
SatelliteMatrix, WeightMatrix, Pnd, Per)

% Declare
funQ = @(x) exp(-(x^2)/2)/sqrt(2*pi);
Pfa = 0.05;
IR = Pnd*Per;
yH = zeros(1, 24);
yV = zeros(1, 24);

% Number of satellites
SatelliteNumber = size(SatelliteMatrix, 1);

% Build matrix H
H = [(SatelliteMatrix(1, 1)-PosEst(1))/RangeEst(1) (SatelliteMatrix(1,
2)-PosEst(2))/RangeEst(1) (SatelliteMatrix(1, 3)-PosEst(3))/RangeEst(1)
1 SatelliteMatrix(1,4)];

for n = 2:SatelliteNumber
    H = [H; (SatelliteMatrix(n, 1)-PosEst(1))/RangeEst(n)
(SatelliteMatrix(n, 2)-PosEst(2))/RangeEst(n) (SatelliteMatrix(n, 3)-
PosEst(3))/RangeEst(n) 1 SatelliteMatrix(n,4)];
end

% Build measurement vector z
z = -(Rho - RangeEst);

% Solve for state vector x0 (complete solution) with WLS
x = inv(H'*WeightMatrix*H)*H'*WeightMatrix*z;

% Solve for state vector xi (not including satellite i) with WLS
for i = 1:SatelliteNumber
    WeightMatrix_i = WeightMatrix;
    WeightMatrix_i(i,i) = 0;
    x_i(i,:) = inv(H'*WeightMatrix_i*H)*H'*WeightMatrix_i*z;
```

```

end

% Build matrices A0 and Ai
A0 = inv(H'*WeightMatrix*H)*H'*WeightMatrix;

for i = 1:SatelliteNumber
    WeightMatrix_i = WeightMatrix;
    WeightMatrix_i(i,i) = 0;
    Ai = inv(H'*WeightMatrix_i*H)*H'*WeightMatrix_i;

    % Calculate covariance matrices
    dPi = (Ai-A0)*inv(WeightMatrix)*(Ai-A0)';

    % Compute test statistics di
    di = (Ai - A0)*z;
    %sqrt(di(1)^2 + di(2)^2)
    %di(3)

    % Compute thresholds
    Lambda_i(i) = max(eig(dPi(1:2, 1:2)));
    Di(i) = sqrt(Lambda_i(i))*erfcinv(Pfa/(2*SatelliteNumber));

    SigmaVi(i) = dPi(3,3);
    Vi(i) = sqrt(SigmaVi(i))*erfcinv(Pfa/2/SatelliteNumber);

    % Compare test statistic with threshold
    if sqrt(di(1)^2 + di(2)^2) > Di(i)
        yH(i) = 1;
    else
        yH(i) = 0;
    end
    if di(3) > Vi(i)
        yV(i) = 1;
    else
        yV(i) = 0;
    end
end
end

```

```

SatelliteNumberPL = 0;
Lambda_iPL = [];
DiPL = [];
SigmaViPL = [];
ViPL = [];

for n = 1:SatelliteNumber
    if (yH(n)==0) && (yV(n)==0)
        SatelliteNumberPL = SatelliteNumberPL + 1;
        Lambda_iPL = [Lambda_iPL Lambda_i(n)];
        DiPL = [DiPL Di(n)];
        SigmaViPL = [SigmaViPL SigmaVi(n)];
        ViPL = [ViPL Vi(n)];
    end
end

if SatelliteNumberPL >= 6
    [HPL, VPL] = ProtectionLevelSS(IR, DiPL, ViPL, Lambda_iPL,
    SigmaViPL, SatelliteNumberPL);
else
    HPL = 0;
    VPL = 0;
end

% Compute solution
W_x = WeightMatrix;
for i = 1:SatelliteNumber
    if (yH(i) == 1) || (yV(i) == 1)
        W_x(i,i) = 0;
    end
end

x2 = pinv(H'*W_x*H)*H'*W_x*z;
xOUT = x2(1:3) + PosEst;
end

```

8.5.4 Nivel de protección RAIM basado en separación de soluciones

```
%{
Matlab function for the protection level for the solution separation
RAIM computation
%}
function [HPL, VPL] = ProtectionLevelSS(IR, D, V, Lambda, SigmaV, N)

% Declare
PLmaxH = 0;
PLmaxV = 0;

Delta = chi2inv(1 - IR, N - 5);

for i = 1:N

    Delta_i = sqrt(Lambda(i)*Delta);

    if Delta_i + D(i) > PLmaxH
        PLmaxH = Delta_i + D(i);
    end

end

for i = 1:N
    Gamma_i = SigmaV(i)*erfcinv(1 - IR/2);
    if Gamma_i + V(i) > PLmaxV
        PLmaxV = Gamma_i + V(i);
    end

end

HPL = PLmaxH;
VPL = PLmaxV;
end
```

8.5.5 Estimador de Huber

```
%{
Matlab function for the Huber estimator technique
%}

function [xOUT, y] = M_EstimatorHuber(Rho, RangeEst, PosEst,
SatelliteMatrix, WeightMatrix)

% Number of satellites
SatelliteNumber = size(SatelliteMatrix, 1);

% Build matrix H
H = [(SatelliteMatrix(1, 1)-PosEst(1))/RangeEst(1) (SatelliteMatrix(1,
2)-PosEst(2))/RangeEst(1) (SatelliteMatrix(1, 3)-PosEst(3))/RangeEst(1)
1 SatelliteMatrix(1,4)];

for n = 2:SatelliteNumber
    H = [H; (SatelliteMatrix(n, 1)-PosEst(1))/RangeEst(n)
(SatelliteMatrix(n, 2)-PosEst(2))/RangeEst(n) (SatelliteMatrix(n, 3)-
PosEst(3))/RangeEst(n) 1 SatelliteMatrix(n,4)];
end

% Build measurement vector z
z = -(Rho - RangeEst);

%%%%%%%%%% STEP 1 %%%%%%%%%%%
% Solve for state vector x with WLS
x = inv(H'*WeightMatrix*H)*H'*WeightMatrix*z;

%%%%%%%%%% STEP 2 %%%%%%%%%%%
% Calculate residuals r
r = z - H*x;
% Calculate weights using Huber's function
w = HuberFunction(r, SatelliteNumber);

% Update weights
for n = 1:SatelliteNumber
    WeightMatrix(n, n) = w(n)*WeightMatrix(n, n);
end
```



```

##### STEP 3 #####
ConvergenceEq = 0;
for n = 1:SatelliteNumber
    ConvergenceEq = ConvergenceEq + (WeightMatrix(n, n)*(z(n) - H(n,
:)*x)*H(n, :));
end

if (ConvergenceEq < 0.1) & (ConvergenceEq > -0.1)
    y = 0;
else
    y = 1;

    % Recalculate x
    x = inv(H'*WeightMatrix*H)*H'*WeightMatrix*z;
end

##### STEP 4 #####
if y == 1

    while(any(ConvergenceEq >= 0.1) | any(ConvergenceEq <= -0.1))

        % Iterate 2 and 3
        ##### STEP 2 #####
        % Calculate residuals r
        r = z - H*x;
        % Calculate weights using Huber's function
        w = HuberFunction(r, SatelliteNumber);

        % Update weights
        for n = 1:SatelliteNumber
            WeightMatrix(n, n) = w(n)*WeightMatrix(n, n);
        end

        ##### STEP 3 #####
        ConvergenceEq = 0;
        for n = 1:SatelliteNumber
            ConvergenceEq = ConvergenceEq + (WeightMatrix(n, n)*(z(n) -
H(n, :)*x)*H(n, :));

```

```

end

if (ConvergenceEq < 0.1) & (ConvergenceEq > -0.1)
    y = 0;
else
    y = 1;
    % recalculate x
    x = inv(H'*WeightMatrix*H)*H'*WeightMatrix*z;
end
end

end

% Compute position output
x2 = pinv(H'*WeightMatrix*H)*H'*WeightMatrix*z;
xOUT = x2(1:3) + PosEst;

end

%{
Matlab function for the Huber function
%}
function w = HuberFunction(r, N)

rMedian = median(r);
rStdDev = median(abs(r-rMedian))/0.6745;

for n = 1:N
    if abs(r(n)/rStdDev) <= 1.345
        w(n) = 1;
    else
        w(n) = 1.345/abs(r(n)/rStdDev);
    end
end
end
end

```

8.5.6 Estimador IGGII

```
%{
Matlab function for the IGGII estimator technique
%}

function [xOUT, y] = M_EstimatorIGGII(Rho, RangeEst, PosEst,
SatelliteMatrix, WeightMatrix)

% Number of satellites
SatelliteNumber = size(SatelliteMatrix, 1);

% Build matrix H
H = [(SatelliteMatrix(1, 1)-PosEst(1))/RangeEst(1) (SatelliteMatrix(1,
2)-PosEst(2))/RangeEst(1) (SatelliteMatrix(1, 3)-PosEst(3))/RangeEst(1)
1 SatelliteMatrix(1,4)];

for n = 2:SatelliteNumber
    H = [H; (SatelliteMatrix(n, 1)-PosEst(1))/RangeEst(n)
(SatelliteMatrix(n, 2)-PosEst(2))/RangeEst(n) (SatelliteMatrix(n, 3)-
PosEst(3))/RangeEst(n) 1 SatelliteMatrix(n,4)];
end

% Build measurement vector z
z = -(Rho - RangeEst);

%%%%%%%%%% STEP 1 %%%%%%%%%%%
% Solve for state vector x with WLS
x = inv(H'*WeightMatrix*H)*H'*WeightMatrix*z;

%%%%%%%%%% STEP 2 %%%%%%%%%%%
% Calculate residuals r
r = z - H*x;
% Calculate weights using Huber's function
w = IGGIIFunction(r, SatelliteNumber);

% Update weights
for n = 1:SatelliteNumber
    WeightMatrix(n, n) = w(n)*WeightMatrix(n, n);
end
```

```

##### STEP 3 #####
ConvergenceEq = 0;
for n = 1:SatelliteNumber
    ConvergenceEq = ConvergenceEq + (WeightMatrix(n, n)*(z(n) - H(n,
:)*x)*H(n, :));
end

if (ConvergenceEq < 0.1) & (ConvergenceEq > -0.1)
    y = 0;
else
    y = 1;

    % Recalculate x
    x = inv(H'*WeightMatrix*H)*H'*WeightMatrix*z;
end

##### STEP 4 #####
if y == 1

    while(any(ConvergenceEq >= 0.1) | any(ConvergenceEq <= -0.1))

        % Iterate 2 and 3
        ##### STEP 2 #####
        % Calculate residuals r
        r = z - H*x;
        % Calculate weights using Huber's function
        w = IGGIIFunction(r, SatelliteNumber);

        % Update weights
        for n = 1:SatelliteNumber
            WeightMatrix(n, n) = w(n)*WeightMatrix(n, n);
        end

        ##### STEP 3 #####
        ConvergenceEq = 0;
        for n = 1:SatelliteNumber
            ConvergenceEq = ConvergenceEq + (WeightMatrix(n, n)*(z(n) -
H(n, :)*x)*H(n, :));

```

```

end

if (ConvergenceEq < 0.1) & (ConvergenceEq > -0.1)
    y = 0;
else
    y = 1;
    % recalculate x
    x = inv(H'*WeightMatrix*H)*H'*WeightMatrix*z;
end
end

end

% Compute position output
x2 = pinv(H'*WeightMatrix*H)*H'*WeightMatrix*z;
xOUT = x2(1:3) + PosEst;

end
%{
Matlab function for the IGGII estimator function
%}
function w = IGGIIFunction(r, N)

for n = 1:N
    if abs(r(n)) <= 1.345
        w(n) = 1;
    elseif (abs(r(n)) > 1.345) && (abs(r(n)) <= 3)
        w(n) = (1.345/abs(r(n)))*((3-abs(r(n)))/(3-1.345))^2;
    else
        w(n) = 0;
    end
end
end

end

```

8.5.7 Integrity Based Protection Level (IBPL)

```
%{
Matlab function for the IBPL technique
%}

function [HPL, VPL] = IBPL(Rho, RangeEst, PosEst, SatelliteMatrix,
WeightMatrix)

% Declare
kMat = [14.9443 149.9944 1500 15000 1500000 15000000 150000000;
        4.2996 14.0890 44.7046 141.4162 447.2111 1414.2128 4472.1318;
        2.6690 6.1871 13.5204 29.2162 62.9849 135.7156 292.3994;
        2.0266 3.9971 7.3100 13.1100 23.3749 41.6019 73.9992;
        1.6781 3.0189 4.9868 8.0263 12.7972 20.3301 32.2512;
        1.4562 2.4696 3.8230 5.7398 8.5105 12.5495 18.4593;
        1.3007 2.1175 3.1349 4.4863 6.3249 8.8531 12.3476;
        1.1846 1.8716 2.6825 3.7076 5.0386 6.7886 9.1043;
        1.0940 1.6895 2.3627 3.1809 4.2044 5.5029 7.1629;
        1.0209 1.5485 2.1244 2.8021 3.6245 4.6378 5.8971];

IntegrityRisk = 7; %IR = 10^-IntegrityRisk

% Number of satellites
SatelliteNumber = size(SatelliteMatrix, 1);

% Build matrix H
H = [(SatelliteMatrix(1, 1)-PosEst(1))/RangeEst(1) (SatelliteMatrix(1,
2)-PosEst(2))/RangeEst(1) (SatelliteMatrix(1, 3)-PosEst(3))/RangeEst(1)
1 SatelliteMatrix(1,4)];

for n = 2:SatelliteNumber
    H = [H; (SatelliteMatrix(n, 1)-PosEst(1))/RangeEst(n)
(SatelliteMatrix(n, 2)-PosEst(2))/RangeEst(n) (SatelliteMatrix(n, 3)-
PosEst(3))/RangeEst(n) 1 SatelliteMatrix(n,4)];
end

% Build measurement vector z
z = Rho - RangeEst;

% Solve for state vector x with WLS
```

```

x = inv(H'*WeightMatrix*H)*H'*WeightMatrix*z;

% Calculate residuals r
r = z - H*x;

% Calculate HDOP
HDOPvar = HDOP(RangeEst, PosEst, SatelliteMatrix);

% Calculate VDOP
VDOPvar = VDOP(RangeEst, PosEst, SatelliteMatrix);

% Select k from matrix
if SatelliteNumber >= 14
    k = kMat(14-4, IntegrityRisk);
else
    k = kMat(SatelliteNumber-4, IntegrityRisk);
end

% Compute HPL
rMod = r(1)^2;
for i = 2:SatelliteNumber
    rMod = rMod + r(i)^2;
end
rMod = sqrt(rMod);
HPL = k*rMod*HDOPvar;
VPL = k*rMod*VDOPvar;

end

```

8.6 Arquitectura del simulador GPS-Galileo

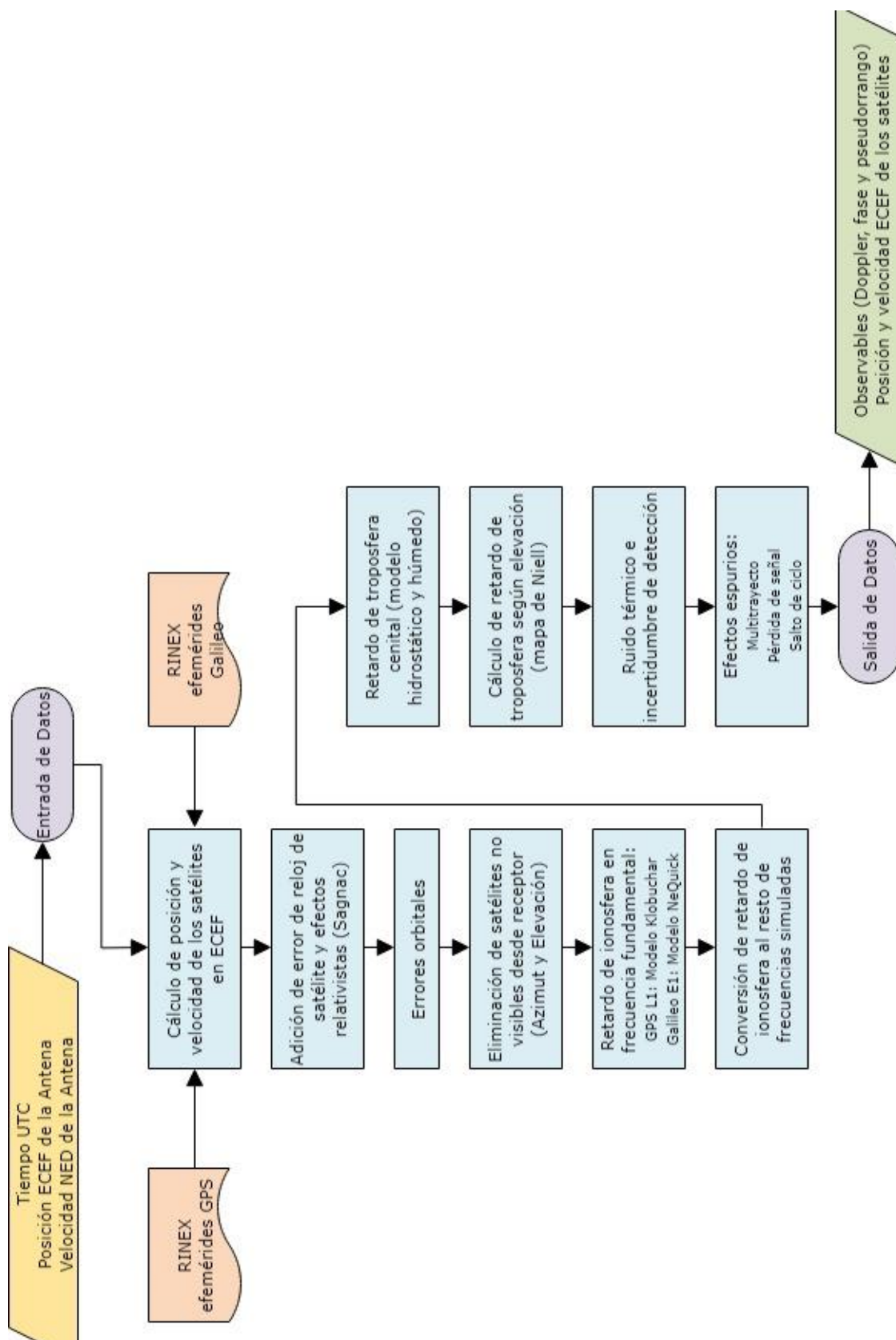


Tabla de contenido

CAPÍTULO 1:	Mediciones	1
1.1	Software	2
1.2	Mano de obra	3
CAPÍTULO 2:	Precios unitarios.....	5
2.1	Software	6
2.2	Mano de obra	7
CAPÍTULO 3:	Sumas parciales.....	9
3.1	Software	10
3.2	Mano de obra	11
CAPÍTULO 4:	Presupuesto general.....	13
4.1	Suma total	14

CAPÍTULO 1: Mediciones

1.1 Software

ELEMENTO	UNIDADES
MATLAB	1
MICROSOFT OFFICE	1

1.2 Mano de obra

ELEMENTO	HORAS
ESTUDIO Y REVISIÓN LITERATURA	30
DETECCIÓN SALTOS DE CICLO	80
SELECCIÓN DE SATÉLITES	90
DETECCIÓN DE MEDIDAS ERRÓNEAS	70
REDACCIÓN MEMORIA	50
PREPARACIÓN DEFENSA	10

CAPÍTULO 2: Precios unitarios

2.1 Software

ELEMENTO	€/UNIDAD
MATLAB	400
MICROSOFT OFFICE	300

2.2 Mano de obra

ELEMENTO	€/HORA
ESTUDIO Y REVISIÓN LITERATURA	9
DETECCIÓN SALTOS DE CICLO	12
SELECCIÓN DE SATÉLITES	12
DETECCIÓN DE MEDIDAS ERRÓNEAS	12
REDACCIÓN MEMORIA	9
PREPARACIÓN DEFENSA	7

CAPÍTULO 3: Sumas parciales

3.1 Software

ELEMENTO	UNIDADES	€/UNIDAD	COSTE(€)
MATLAB	1	400	400
MICROSOFT OFFICE	1	300	300
TOTAL			700

3.2 Mano de obra

ELEMENTO	HORAS	€/HORA	COSTE (€)
ESTUDIO Y REVISIÓN LITERATURA	30	9	270
DETECCIÓN SALTOS DE CICLO	80	12	960
SELECCIÓN DE SATÉLITES	90	12	1080
DETECCIÓN DE MEDIDAS ERRÓNEAS	70	12	840
REDACCIÓN MEMORIA	50	9	450
PREPARACIÓN DEFENSA	10	7	70
TOTAL			3670

CAPÍTULO 4:

Presupuesto general

4.1 Suma total

APARTADO	COSTE (€)
SOFTWARE	700
MANO DE OBRA	3670
PEM	4370
GASTOS GENERALES (10%)	437
IVA (21%)	917.7
PEC	5724.7

