

Proyecto Fin de Carrera
Ingeniería Aeronáutica

Generación automática de trayectorias libres de
obstáculos considerando procedimientos GNSS-
RNAV

Autor: Miguel Ángel Caracuel Jiménez

Tutor: Alfonso Valenzuela Romero

Dpto. de Ingeniería Aeroespacial y Mecánica de
Fluidos

Escuela Técnica Superior de Ingeniería

Sevilla. 2014



Ingeniería de Aeronáutica
Proyecto Fin de Carrera

Generación automática de trayectorias libres de obstáculos considerando procedimientos GNSS- RNAV

Autor:

Miguel Ángel Caracuel Jiménez

Tutor:

Alfonso Valenzuela Romero

Profesor ayudante doctor

Dpto. de Ingeniería Aeroespacial y Mecánica de Fluidos

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2014

Proyecto Fin de Carrera: Generación automática de trayectorias libres de obstáculos considerando procedimientos GNSS-RNAV

Autor: Miguel Ángel Caracuel Jiménez

Tutor: Alfonso Valenzuela Romero

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2014

El Secretario del Tribunal

A Miguel Ángel e Isabel

A mis maestros

Agradecimientos

Dicen que es de bien nacidos el ser agradecidos, y llegados a este punto de mi vida tengo que dar gracias a muchas personas por muchas cosas, por su paciencia, por su dedicación y entrega conmigo, por haberme enseñado muchas de las cosas, pero sobre todo por quererme tal y como soy, no siendo a menudo fácil, sé que se me va a olvidar gente, pero al igual que han hecho durante muchos años, espero que me perdonen.

En primer lugar agradecer a todos los profesores que he tenido a lo largo de mi vida, ellos me han enseñado mucho de lo que hoy se, desde atarme los cordones hasta el proyecto que se presenta a continuación, gracias al Colegio Ahlzahir y a todos sus profesores y en especial a don Manuel Mata y a Don Rafael Villalba, siempre los llevaré en el corazón. Gracias a la familia del CMU San Juan Bosco por haber sido mi casa durante los cinco últimos años y haber sido un complemento a mi formación universitaria. Gracias por formar parte de mi familia y haber cuidado de mí a lo largo de estos veintitrés años.

Agradecer a Alfonso Valenzuela Romero su dedicación, él es el principal artífice de que este proyecto vea la luz, sin su ayuda, conocimientos y sobretodo paciencia conmigo, no podría haber realizado el proyecto. Alfonso es un docente magnifico, me ha enseñado, muchos aspectos técnicos de la aeronáutica, pero con la principal lección que me quedo es con que la perfección no debe ser la meta, sino el hábito, debemos de trabajar cada día dando lo mejor de nosotros mismos y poniendo el alma en todo lo que hacemos.

Gracias a todos mis compañeros de la carrera, desde el primero al último, esta etapa llega a su fin, pero a lo largo de ella hemos compartido muchos momentos, algunos buenos (la mayoría) y otros no tan buenos, pero después de estos cinco años el sabor de boca que me llevo es muy positivo. En especial a Fernando, él ha sido mi principal apoyo desde el día que lo conocí, a día de hoy lo considero mi hermano, y estoy convencido que llegar hasta aquí sin su ayuda no habría sido tan “sencillo”. Gracias a Samu por haber sido el artífice de darle forma, tuya es una parte importante de este proyecto.

Gracias a Tiscar Dueñas, ella ha sido mi principal apoyo desde que empecé el proyecto hasta hoy, solo ella sabe los sacrificios que ha conllevado este proyecto y el trabajo que me ha costado concluirlo. Sin su aliento, ánimo y cariño en los momentos más complicados hubiese sido difícil poder hacer realidad el sueño de estar a punto de acabar la carrera. Han sido momentos duros los que hemos pasado juntos, pero con su amor ha conseguido hacerlo todo más fácil.

Gracias a mis padres, ellos son la razón de que hoy pueda estar aquí, de que haya llegado hasta donde he llegado, su sacrificio, esfuerzo y renuncia a muchos placeres y privilegios con tal de darnos a mi hermana y a mí, la posibilidad de elegir nuestro futuro, pero sobre todo gracias por habernos educado rectamente para ser personas honestas y honradas. Sois el espejo en el que me miro cada día para tratar de ser mejor persona y tratar de paliar mis defectos. Os admiro y quiero mucho.

Y por supuesto gracias a Dios por los dones que me ha dado.

Miguel Ángel Caracuel Jiménez

Sevilla 2014

Resumen

En este proyecto se presenta un algoritmo que permite, dado un punto inicial y otro de destino, así como los obstáculos existentes en la región bajo estudio, generar la trayectoria que necesita de una menor distancia para conectar dichos puntos.

El algoritmo se desarrolla en dos pasos, en primer lugar se implementa un método de generación de grafos, para conectar los posibles *waypoints* entre sí, y posteriormente otro de búsqueda en grafos para obtener la ruta óptima. Para la elección de estos algoritmos, se han estudiados los más comunes en programas similares al que vamos a desarrollar y se han seleccionado el grafo de visibilidad y el A* respectivamente.

Para tener en cuenta que la solución propuesta va a ser recorrida por aeronaves, el programa hace uso de la normativa recogida en el documento 8168 de la OACI para dotar a los *waypoints* de una distancia mínima de separación entre ellos y establecer una distancia de separación entre la trayectoria propuesta y los posibles obstáculos, de modo que nunca se produzca la colisión de la aeronave.

Para finalizar la memoria se describirán dos aplicaciones en las cuales se podría hacer uso del programa como son evitar una tormenta o recorrer un valle.

Abstract

In this project, it is presented an algorithm which allows to, given the initial and final points, as well as the existing obstacles in the area under study, generate the trajectory which needs a shortest distance to connect those points.

To obtain this trajectory, the algorithm is divided in two steps, one is a graphs generation, to connect the possible waypoints and the other is a searcher in graphs to obtain the optimal route. To choose these algorithms, it has been studied the most commons in similar programs and the visibility graphs and A* have been selected, respectively.

To take into account that the given solution is going to be followed by aircraft, the algorithm takes into account the normative collected in the OACI's 8168 document to outfit the waypoints with a minimum distance between them and establish also a separation distance between the given trajectory and the possible obstacles, avoiding a potential collision.

To finish this report, two applications, which would use the program, will be presented, the avoidance of a storm and tour a valley.

Agradecimientos	i
Resumen	iii
Abstract	v
Índice	vii
Índice de Tablas	ix
Índice de Figuras	xi
Notación	xv
1 INTRODUCCIÓN	19
1.1. Motivación	19
1.2. Hipótesis y objetivo	20
1.3. Estructura del documento	20
2 MÉTODOS DE GENERACIÓN DE GRAFOS	23
2.1. Diagrama de Voronoi	23
2.2. Descomposición en celdas. Mallado	24
2.3. Grafo de visibilidad	27
2.4. Elección de un algoritmo de generación de grafos	28
3 ALGORITMOS DE BÚSQUEDA EN GRAFOS	31
3.1. Algoritmo Dijkstra	31
3.2. Algoritmo Best First Search	32
3.3. Algoritmo A*	32
3.4. Elección de un algoritmo de búsqueda en grafos	33
4 IMPLEMENTACIÓN. GRAFO DE VISIBILIDAD + ALGORITMO A*	35
4.1. Resolución del Problema para un Único Obstáculo	36
4.2. Resolución del Problema Para un Número Cualquiera de Obstáculos	40
4.3. Cálculo de Soluciones Alternativas. Algoritmo A*	42
5 CONVERSIÓN DE OBSTÁCULOS CÓNCAVOS EN CONVEXOS. “EAR CLIPPING”	45
5.1. Algoritmo de ejecución del “Ear Clipping”	46
5.2. Resolución de un ejemplo mediante la utilización del programa implementado	49
6 IMPLEMENTACIÓN DE PROCEDIMIENTOS RNAV	53

6.1.	Longitud Mínima de Estabilización	53
6.1.1.	Normativa	53
6.1.2.	Algoritmo de Generación de Mínima Distancia de Tramo RNAV	57
6.2.	Protección Lateral de la Aeronave	60
6.2.1.	Normativa	60
6.2.2.	Algoritmo de Protección Lateral de la Aeronave	66
7	APLICACIONES	71
7.1.	Esquivar una tormenta	71
7.2.	Pasar por un valle	74
8	CONCLUSIONES	77
	REFERENCIAS	79
	APÉNDICE A. BATERÍA DE CASOS PARA UN OBSTÁCULO	81
	APÉNDICE B. BATERÍA DE CASOS PARA “N” OBSTÁCULOS	85
	APÉNDICE C. BATERÍA DE CASOS PARA EL “EAR CLIPPING”	91
	APÉNDICE D. BATERÍA DE CASOS COMPROBACIÓN DE LA MÍNIMA DISTANCIA DE TRAMO RNAV	95
	APÉNDICE E. BATERÍA DE CASOS PARA LA COMPROBACIÓN DE LA PROTECCIÓN LATERAL	105

Índice de Tablas

Tabla 1	Visibilidad para el ejemplo de un obstáculo. Ejemplo 1.	38
Tabla 2	Visibilidad para el ejemplo de un obstáculo. Ejemplo 2.	39
Tabla 3	Visibilidad de nodos ejemplo varios obstáculos.	41
Tabla 4	Soluciones Alternativas. Distancia y Ruta.	43
Tabla 5	Visibilidad para el ejemplo de un obstáculo cóncavo. Ejemplo 1.	51
Tabla 6	Distancia mínima de estabilización entre puntos de recorrido de paso (bajas velocidades).	56
Tabla 7	Distancia mínima de estabilización entre puntos de recorrido de paso (altas velocidades).	56
Tabla 8	Valores intermedios.	61
Tabla 9	Semianchura del área de franqueamiento de obstáculos.	61
Tabla 10	Coordenadas de los puntos de paso solución de nuestro problema, 1.	73
Tabla 11	Coordenadas de los puntos de paso solución de nuestro problema, 2.	75
Tabla 12	Caso 1 de la batería de pruebas para un obstáculo. Visibilidad.	81
Tabla 13	Caso 2 de la batería de pruebas para un obstáculo. Visibilidad.	82
Tabla 14	Caso 3 de la batería de pruebas para un obstáculo. Visibilidad.	83
Tabla 15	Caso 4 de la batería de pruebas para un obstáculo. Visibilidad.	83
Tabla 16	Caso 1 de la batería de pruebas para “n” obstáculo. Visibilidad.	86
Tabla 17	Caso 2 de la batería de pruebas para “n” obstáculo. Visibilidad.	86
Tabla 18	Caso 3 de la batería de pruebas para “n” obstáculo. Visibilidad.	87
Tabla 19	Caso 4 de la batería de pruebas para “n” obstáculo. Visibilidad.	88
Tabla 20	Caso 5 de la batería de pruebas para “n” obstáculo. Visibilidad.	88

Índice de Figuras

Figura 1 Ejemplo de un polígono simple y no simple.	20
Figura 2 Diagrama de Voronoi. Ejemplo.	23
Figura 3 Algoritmo de Fortune.	24
Figura 4 Descomposición Trapezoidal.	25
Figura 5 Eliminación de las rectas no necesarias.	25
Figura 6 Nodos posibles mediante descomposición trapezoidal.	26
Figura 7 Descomposición Homogénea. Uniforme y no uniforme respectivamente.	26
Figura 8 Descomposición heterogénea.	27
Figura 9 Errores a evitar a la hora de la realización del mallado.	27
Figura 10 Grafo de visibilidad.	28
Figura 11 Comparación entre los métodos trapezoidales y grafos de visibilidad.	29
Figura 12 Ejemplo variables de entrada.	35
Figura 13 Un obstáculo. Ejemplo 1.	37
Figura 14 Visibilidad para nodos contenidos en la misma recta.	38
Figura 15 Un obstáculo. Ejemplo 2.	39
Figura 16 Un obstáculo. Ejemplo 3.	40
Figura 17 Varios obstáculos. Ejemplo 1.	40
Figura 18 Varios obstáculos. Ejemplo 2.	41
Figura 19 Cálculo de soluciones alternativas, paso 1.	42
Figura 20 Cálculo de soluciones alternativas, paso 2.	43
Figura 21 Cálculo de soluciones alternativas, paso 3.	44
Figura 22 Ejemplo obstáculo cóncavo.	45
Figura 23 Ejemplo método "Ear Clipping", paso 1.	46
Figura 24 Ejemplo método "Ear Clipping", paso 2.	47
Figura 25 Ejemplo método "Ear Clipping", paso 3.	47
Figura 26 Ejemplo método "Ear Clipping", paso 4.	47
Figura 27 Ejemplo método "Ear Clipping", paso 5.	48
Figura 28 Ejemplo método "Ear Clipping", paso 6.	48
Figura 29 Ejemplo método "Ear Clipping", paso 7.	48
Figura 30 Decágono inicial y su correspondiente triangulación.	49
Figura 31 Obstáculo cóncavo. Problema inicial.	49
Figura 32 Obstáculo cóncavo. Trinagulación	50

Figura 33 Obstáculo cóncavo. Solución final.	50
Figura 34 Obstáculo cóncavo. Valle. Ejemplo 2.	51
Figura 35 Dos puntos de recorrido de paso.	54
Figura 36 Punto de recorrido de paso seguido de un punto de recorrido de sobrevuelo.	54
Figura 37 Dos punto de recorrido de sobrevuelo.	55
Figura 38 Punto de recorrido de sobrevuelo seguido de un punto de recorrido de paso.	55
Figura 39 Ejemplo del algoritmo de Distancia mínima de estabilización.	57
Figura 40 Ejemplo RNAV, 1.	58
Figura 41 Ejemplo RNAV, 2.	59
Figura 42 Ejemplo eliminación punto de paso innecesario, 1.	59
Figura 43 Ejemplo eliminación punto de paso innecesario, 2.	60
Figura 44 Ejemplo "Minkowski Sum".	61
Figura 45 Círculos limitadores. Límite exterior.	63
Figura 46 Diferencia entre la protección lateral disponible y necesaria, límite interior.	64
Figura 47 Puntos de viraje anterior y posterior.	65
Figura 48 Diferencia entre la protección lateral disponible y necesaria para el viraje, límite exterior.	66
Figura 49 Ejemplo de recorridos sin intersección de los obstáculos.	66
Figura 50 Punto de recorrido de paso. Ejemplo de trayectoria.	67
Figura 51 Ejemplo de obstáculo con tres vértices cóncavos y rectas paralelas entre sí.	67
Figura 52 Ejemplo de obstáculo con dos vértices cóncavos y rectas no paralelas entre sí.	68
Figura 53 Ejemplo de intersección de obstáculos, 1.	68
Figura 54 Ejemplo de intersección de obstáculos, 2.	69
Figura 55 Ejemplo de intersección de obstáculos, 3.	69
Figura 56 Ejemplo de aplicación para evitar una tormenta, 1.	71
Figura 57 Ejemplo de aplicación para evitar una tormenta, 2.	72
Figura 58 Ejemplo de aplicación para evitar una tormenta, 3.	72
Figura 59 Ejemplo de aplicación para evitar una tormenta, 4.	73
Figura 60 Ejemplo para recorrer un valle, 1.	74
Figura 61 Ejemplo para recorrer un valle, 2.	74
Figura 62 Ejemplo para recorrer un valle, 3.	75
Figura 63 Ejemplo para recorrer un valle, 4.	76
Figura 64 Ejemplo para recorrer un valle, 5.	76
Figura 65 Caso 1 de la batería de pruebas para un obstáculo.	81
Figura 66 Caso 2 de la batería de pruebas para un obstáculo.	82
Figura 67 Caso 3 de la batería de pruebas para un obstáculo.	83
Figura 68 Caso 4 de la batería de pruebas para un obstáculo.	84
Figura 69 Caso 1 de la batería de pruebas para "n" obstáculos.	85
Figura 70 Caso 2 de la batería de pruebas para "n" obstáculos.	86
Figura 71 Caso 3 de la batería de pruebas para "n" obstáculos.	87
Figura 72 Caso 4 de la batería de pruebas para "n" obstáculos.	88

Figura 73 Caso 5 de la batería de pruebas para “n” obstáculos.	89
Figura 74 Caso 1 de la batería de pruebas para “Ear Clipping”.	91
Figura 75 Caso 2 de la batería de pruebas para “Ear Clipping”.	92
Figura 76 Caso 3 de la batería de pruebas para “Ear Clipping”.	93
Figura 77 Caso 4 de la batería de pruebas para “Ear Clipping”.	93
Figura 78 Caso 5 de la batería de pruebas para “Ear Clipping”.	94
Figura 79 Caso 1 de la batería de pruebas para distancia de tramo RNAV.	95
Figura 80 Caso 2 de la batería de pruebas para distancia de tramo RNAV.	96
Figura 81 Caso 3 de la batería de pruebas para distancia de tramo RNAV.	96
Figura 82 Caso 4 de la batería de pruebas para distancia de tramo RNAV.	97
Figura 83 Caso 5 de la batería de pruebas para distancia de tramo RNAV.	97
Figura 84 Caso 6 de la batería de pruebas para distancia de tramo RNAV.	98
Figura 85 Caso 7 de la batería de pruebas para distancia de tramo RNAV.	98
Figura 86 Caso 8 de la batería de pruebas para distancia de tramo RNAV.	99
Figura 87 Caso 9 de la batería de pruebas para distancia de tramo RNAV.	99
Figura 88 Caso 10 de la batería de pruebas para distancia de tramo RNAV.	100
Figura 89 Caso 11 de la batería de pruebas para distancia de tramo RNAV.	100
Figura 90 Caso 12 de la batería de pruebas para distancia de tramo RNAV.	101
Figura 91 Caso 13 de la batería de pruebas para distancia de tramo RNAV.	101
Figura 92 Caso 14 de la batería de pruebas para distancia de tramo RNAV.	102
Figura 93 Caso 15 de la batería de pruebas para distancia de tramo RNAV.	102
Figura 94 Caso 16 de la batería de pruebas para distancia de tramo RNAV.	103
Figura 95 Caso 1 de la batería de pruebas para la protección lateral.	105
Figura 96 Caso 2 de la batería de pruebas para la protección lateral.	106

Notación

A	Pendiente de la recta
A*	Algoritmo A*
A1	Distancia mínima estabilización punto de recorrido de paso 1
A2	Distancia mínima estabilización punto de recorrido de paso 2
A/W	Anchura del área de franqueamiento de obstáculos
B	Término independiente ecuación de la recta
BV	Valores intermedios
B1	Distancia mínima estabilización punto de recorrido de sobrevuelo 1
B2	Distancia mínima estabilización punto de recorrido de sobrevuelo 2
C	Vértices convexos
D	Resistencia aerodinámica
D*	Algoritmo D*
E	Efecto del viento
ET	Vector Ear Tip
f	Función de evaluación
G	Grafo
g	Coste real del camino recorrido desde el nodo inicial
h	Altitud de vuelo
h1	Punto de paso 1 candidato a sustituir al punto de paso i
h2	Punto de paso 2 candidato a sustituir al punto de paso i
i	Nodo a evaluar
j	Valor heurístico de la distancia al nodo final
Li	Sustentación aerodinámica
L	Subconjunto de puntos de paso conectados entre sí
l	Línea de barrido
obs	Coordenadas de todos los vértices del problema
P	Punto
Pf	Punto final
Ps	Punto inicial
Q	Vector cola
N	Puntos de paso candidatos a ser solución
n	Número total de puntos
numpuntos	Número de vértices de cada obstáculo
R	Régimen de viraje
r	Radio de giro
T	Vértices que forman el triángulo
Ti	Empuje
V	Velocidad
W	Peso de la aeronave

w	Viento normalizado OACI
WP1	Punto de recorrido 1
WP2	Punto de recorrido 2
XTT	Tolerancia perpendicular a la derrota 2σ
x	Coordenada cartesiana
y	Coordenada cartesiana
θ	Rumbo
δ	Posición de la palanca de gases
μ	Ángulo de balance

1 INTRODUCCIÓN

Este proyecto nace de la necesidad de ser capaces de generar una ruta entre dos puntos, inicial y final, de forma automática y cumpliendo con todos los requisitos necesarios para asegurar la seguridad de la aeronave. A continuación se muestra la motivación que ha llevado al estudio y desarrollo de este trabajo.

1.1. Motivación

La motivación de este proyecto surge de una necesidad real de nuestra sociedad. Se puede observar cómo, en lo referido a trayectorias de aeronaves, la tendencia actual es el *free flight* tanto en aviación comercial como en las aeronaves no tripuladas.

Actualmente las aeronaves para llevar a cabo una ruta siguen las aerovías existentes en el cielo. El *free flight*, por su parte, lo que permite, como su nombre indica, es el vuelo libre. Es decir, cada aeronave tendría la capacidad de seguir la ruta más interesante. Para llevar a cabo el *free flight* se necesitaría disponer de algoritmos que permitan la generación de trayectorias que sean seguras, eficientes y volables.

La **seguridad** se consigue haciendo que la trayectoria evite el propio terreno y aquellas zonas del espacio que sean potencialmente peligrosas como por ejemplo otras aeronaves o determinados fenómenos meteorológicos, tales como tormentas o ráfagas. Otro problema que se tiene que evitar es el asociado a los sistemas de la aeronave, es decir, que aun existiendo errores de precisión de los sensores o de los sistemas de guiado, la aeronave deberá no verse comprometida de manera que se produzca un accidente.

La **eficiencia** se consigue escogiendo el camino que permita a la aeronave reducir los costes de operación, que en el caso de ausencia de viento se corresponde con el camino más corto entre el origen y el destino. Este hace referencia a aquel tal que la distancia necesaria para ir de un punto a otro sea la menor de todas. En primer lugar, es necesario indicar que para este trabajo se puede tomar cualquier tipo de distancia, Manhattan, Mahalanobis, Euclídea, ... En nuestro caso, se ha optado por tomar la euclídea ya que proporciona una solución más adecuada para el tipo de resultado que se pretende obtener para el problema, ya que el camino más corto entre dos puntos es la línea recta.

El que la trayectoria sea **volable** es importante, pues las aeronaves presentan ciertas limitaciones físicas a la hora de realizar maniobras. Para tener la certeza de que el camino generado por el algoritmo sea volable, se ha optado por cumplir la normativa recogida en el documento 8168 para procedimientos RNAV basados en sistemas GNSS.

El **sistema GNSS**, Global Navigation Satellite System, está formado por una constelación de satélites que transmite señales utilizadas para el posicionamiento y localización en cualquier parte del globo terrestre, ya sea en tierra, mar o aire.

El **RNAV**, Area Navigation, es un método de navegación aérea que permite la operación del avión en cualquier trayectoria de vuelo deseada, dentro de la cobertura de las ayudas para la navegación bien sean estaciones en tierra o satélites.

Teniendo en cuenta los requisitos descritos anteriormente se ha llevado a cabo la construcción de un programa el cual sea capaz de generar las trayectorias adecuadas para satisfacer las necesidades de cada aeronave.

1.2. Hipótesis y objetivo

El objetivo de este proyecto es llevar a cabo el desarrollo de un programa el cual sea capaz de proporcionar la ruta más corta, en el plano horizontal, entre dos puntos evitando todos los obstáculos y cumpliendo los requisitos expuestos por la OACI en el documento 8168.

Los obstáculos pueden ser como se ha mencionado anteriormente cualquier elemento que ponga en peligro la operación de la aeronave, fenómenos meteorológicos adversos, el terreno u otra aeronave. En este proyecto se considera que los obstáculos presentan las siguientes características:

- La ubicación y forma de los obstáculos que nuestra aeronave deberá evitar serán conocidas.
- Son poligonales y vienen descritos por las coordenadas de sus vértices.
- Los vértices se deben enumerar en sentido horario. Esta condición es muy importante, ya que si introducimos los vértices en sentido anti horario, aunque gráficamente no existiría ninguna diferencia, se consideraría como región prohibida la que en realidad es la zona volable.
- Para un obstáculo, cada vértice se enumerará una única vez.
- Los vértices se etiquetarán con los valores de 2 hasta $n+1$ siendo n el número total de vértices debido a todos los obstáculos.
- Los obstáculos han de ser polígonos simples¹.
- Se permitirá que la aeronave recorra las aristas del obstáculo, ya que realmente estas no pertenecerán al obstáculo, sino que será un recerido empleado de manera que la trayectoria generada por el programa sea segura.
- A la hora de aplicar la normativa para procedimientos RNAV, se tendrán en cuenta criterios en ruta.
- Se considera que la aeronave se desplaza a velocidad constante.

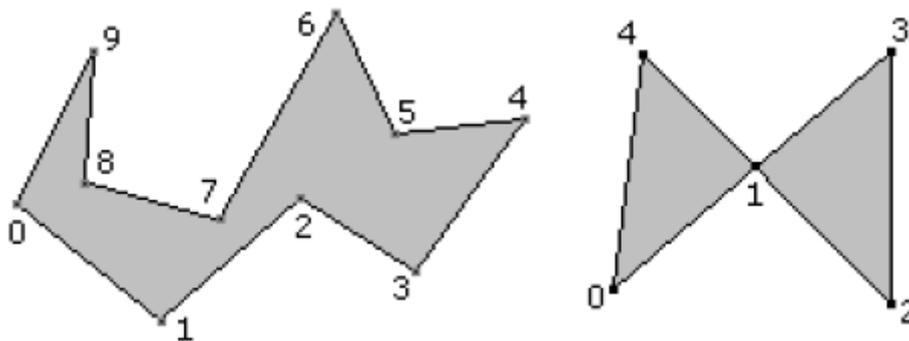


Figura 1 Ejemplo de un polígono simple y no simple.

1.3. Estructura del documento

En la memoria de este proyecto se desarrolla en los capítulos 2 a 5 los métodos de generación de trayectorias, en ellos se describirán los métodos existentes tanto para la generación de grafos como para la búsqueda de grafos, así como su implementación. El capítulo 5 es un caso particular, en el cual se desarrolla la aplicación del código para obstáculos cóncavos.

El capítulo 6 está dedicado a los procedimientos GNSS-RNAV, en este se le proporciona una serie de características a la ruta para que sea segura y volable. Las posibles aplicaciones para las que puede ser empleado el programa que se ha desarrollado vienen recogidas en el capítulo 7 de esta memoria, mientras que en el último capítulo se recogen las conclusiones de la memoria, así como proyectos que pueden surgir a partir de este trabajo.

¹ Un polígono simple es aquel cuyos lados no adyacentes no se intersectan.

Para finalizar se recogen cinco apéndices los cuales muestran una serie de casos propuestos para comprobar que el código implementado funciona correctamente y el resultado proporcionado es el requerido.

2 MÉTODOS DE GENERACIÓN DE GRAFOS

Un grafo $G = (N, L)$ se compone de dos subconjuntos. El subconjunto N son los puntos de paso candidatos a ser parte de la solución del problema, así como los nodos inicial y final. Estos puntos de paso vendrán definidos por sus coordenadas cartesianas, (x, y) . Mientras que el subconjunto L indica qué puntos de paso están conectados entre ellos. Es decir, un grafo representa un conjunto de nodos unidos en red [1, 9, 10 y 11].

Existen diversos métodos para la generación de grafos a partir de un mapa de la superficie o escenario a estudiar. En este capítulo se van a describir de forma somera los que se han considerado interesantes teniendo en cuenta las características de nuestro problema. Éstos son, los diagramas de Voronoi, descomposición en celdas y el grafo de visibilidad.

2.1 Diagrama de Voronoi

Sea P un conjunto de n puntos distintos en el plano, siendo $n > 1$. Se define el diagrama de Voronoi de P como la subdivisión del plano en n celdas, una para cada punto de P , con la propiedad de que dado un punto q del plano, éste pertenecerá a la celda del punto al que esté más cercano [3 y 8].

En la Figura 2 se representa un caso como éste. En dicha figura se observa cómo a cada punto de P , se le asigna una región del espacio, la cual comprende a todos aquellos puntos del espacio que estén más próximos al punto P antes que a cualquier otro.

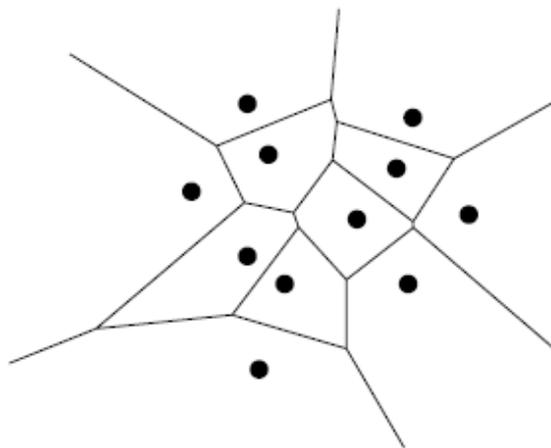


Figura 2 Diagrama de Voronoi. Ejemplo.

Dos de las propiedades más importantes de los diagramas de Voronoi son que son estables, es decir, que si un punto se desplaza un poco, las regiones se verán muy poco afectadas. La otra propiedad es que las celdas son siempre convexas.

Para la generación de las celdas, existen diferentes algoritmos, siendo uno de ellos el algoritmo de Fortune. Para la explicación de este algoritmo es necesario explicar dos conceptos previamente.

- Línea de barrido, "l". Esta línea avanza desde un extremo a otro de la región de interés, en vertical u horizontal. Para cada instante de tiempo se estudia a qué obstáculo está más próximo cada punto del espacio, tomando esta línea como si perteneciese al conjunto P . Únicamente se estudian los puntos por los que ya ha pasado esta línea.

- Línea de playa. Recibe este nombre por su parecido a las curvas que el oleaje deja en la arena cuando este se mueve en la orilla. Se genera para cada instante y determina el límite de los puntos que pertenecen a la línea de barrido o a los obstáculos esta línea tiene forma de parábola.

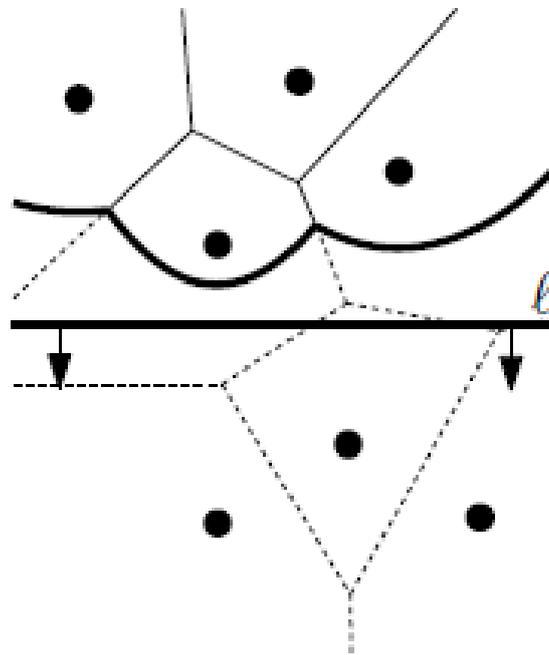


Figura 3 Algoritmo de Fortune.

En la Figura 3 se puede apreciar con mayor claridad estos conceptos. En esta figura se indica cómo la línea de barrido, “l”, se desplaza en dirección vertical y con sentido hacia abajo, por lo tanto los puntos ya estudiados son aquellos que están situados por encima de la línea de barrido. Se observa que hay zonas, las situadas más arriba, que ya tienen delimitada la frontera, pues el algoritmo ya las ha recorrido completamente, y otras sin embargo, en las que aparece la línea de playa, es decir, aún no está finalizado el cálculo de la frontera definitiva, pues existe una cierta proximidad a la línea de barrido por parte del obstáculo.

Este método se puede extender a obstáculos 2-D. Dado el conjunto P formado por líneas rectas, estas líneas pueden dividirse en tres partes, los vértices inicial y final y el resto de la recta. Una vez realizada esta división para cada una de las rectas, se procedería a obtener el diagrama de Voronoi mediante el mismo procedimiento de la línea de barrido explicado anteriormente. El hecho de llevar a cabo esta descomposición de las rectas permite una resolución del problema de una manera más sencilla, pues la problemática de los obstáculos en 2-D son los vértices de los mismos.

En este método el subconjunto L viene compuesto por todos los segmentos que delimitan las celdas de cada punto del conjunto P, mientras que el subconjunto N lo forman los puntos obtenidos de la intersección de las rectas que forman el subconjunto L.

Los posibles caminos que producen este método no serán, en general, los más cortos, sino que serán los más seguros, es decir, la posible trayectoria pasará lo más lejos posible del obstáculo.

2.2. Descomposición en celdas. Mallado

En el método de descomposición en celdas la región bajo estudio se divide en un determinado número de celdas las cuales no pueden solaparse. Para llevar a cabo la descomposición en celdas hay diversos métodos. En nuestro caso se van a describir dos de ellos.

- Descomposición trapezoidal. Una vez definida el área bajo estudio, habiendo delimitado las fronteras exteriores también, se da paso a la descomposición. De cada vértice se genera una línea vertical u horizontal, en el ejemplo de la Figura 4 es vertical, hasta que ésta intersecte con otro objeto o con una de las fronteras exteriores. Posteriormente los segmentos que se encuentren en el interior de los obstáculos se eliminan, éste hecho se recoge en la Figura 5. Una vez concluido este proceso, se calcula el punto medio de cada una de los segmentos verticales obtenidas así como de las regiones visitables, formando todos estos puntos el subconjunto N. por su parte el subconjunto L está formado por todas las rectas que unan nodos contiguos del subconjunto N. En la Figura 6 se observa el último paso del proceso.

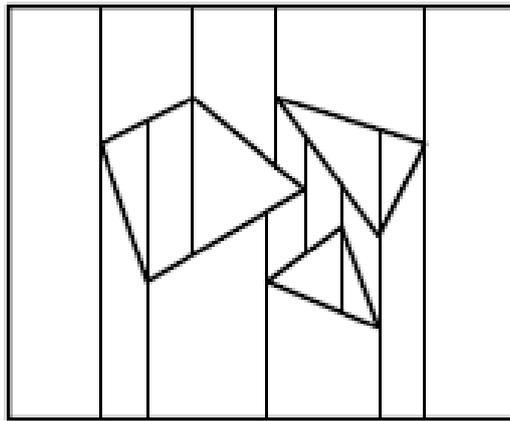


Figura 4 Descomposición Trapezoidal.

Una de las características de este método es que entre los nodos y los obstáculos puede existir una

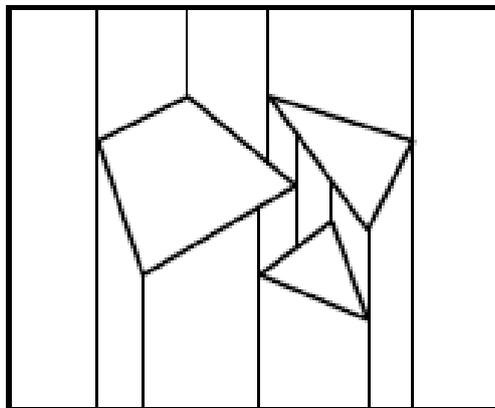


Figura 5 Eliminación de las rectas no necesarias.

distancia muy grande. Este hecho se produce cuando la distancia entre obstáculos o entre las fronteras exteriores y los mismos es considerable. Por tanto, los caminos que se obtendrán con este método son, en general, muy seguros, pero a costa de aumentar la distancia a recorrer.

- Quadtree. Divide el área bajo estudio en regiones de menor tamaño. Estas divisiones pueden ser homogéneas o heterogéneas, en la Figura 7 se muestra un ejemplo de cada uno de estos tipos. El subconjunto N viene determinado por los centros de cada uno de los cuadriláteros o el ortocentro en el caso de que la región sea dividida en triángulos.
 - Descomposición homogénea. El mallado puede estar formado por cuadrados, o bien dividir estos últimos en triángulos a través de su diagonal. Se tienen que tener en cuenta los siguientes requisitos.

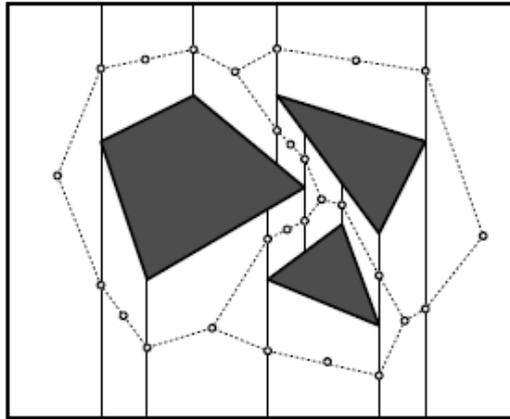


Figura 6 Nodos posibles mediante descomposición trapezoidal.

- ❖ El cuadrado o triángulo de menor unidad vendrá dado por el tamaño que se determine, pudiendo ser una longitud característica de la aeronave o un tamaño decidido por el usuario.
- ❖ La región bajo estudio debe tener las dimensiones adecuadas para ser dividida en $2n$ subpartes de lado la unidad mínima de cuadrado, donde n es la dimensión característica del cuadrado o triángulo de menor unidad.
- ❖ Subpartes contiguas no pueden diferir en más de una unidad, es decir si un cuadrado tiene de lado una determinada longitud, los cuadrados contiguos pueden tener una longitud del doble, igual o la mitad.
- ❖ Si uno de los cuadrados contiene parte de un obstáculo, ese cuadrado no podrá ser visitado.
- ❖ El cuadrado deberá disminuir sus dimensiones tanto como sea posible hasta que no tenga ningún obstáculo en su interior. Es decir, si una subparte no tienen ningún obstáculo en su interior, no es necesario disminuir su tamaño. Aunque podría llevarse a cabo para obtener un mallado uniforme. Ello conllevaría un aumento del coste computacional tanto para el mallado como para la obtención de la ruta óptima.

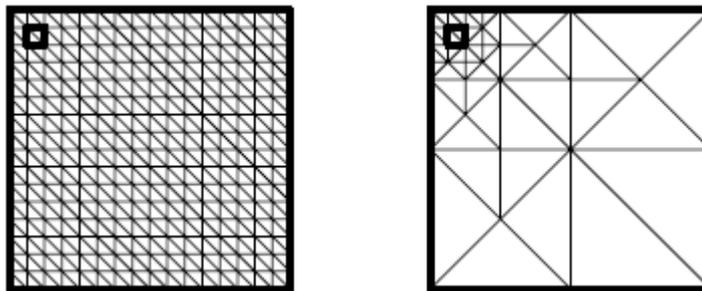


Figura 7 Descomposición Homogénea. Uniforme y no uniforme respectivamente.

- Descomposición heterogénea. Este tipo de mallado debe poseer las siguientes características.
 - ❖ Conformidad. Cada vértice debe unirse con otro vértice.
 - ❖ Buen diseño. Los ángulos deben no ser muy pequeños, mayores de 45 grados es un valor razonable.
 - ❖ Respetar los obstáculos. Si una de las divisiones tiene parte de su región dentro del obstáculo, entonces se deberá evitar que sea un punto a recorrer.

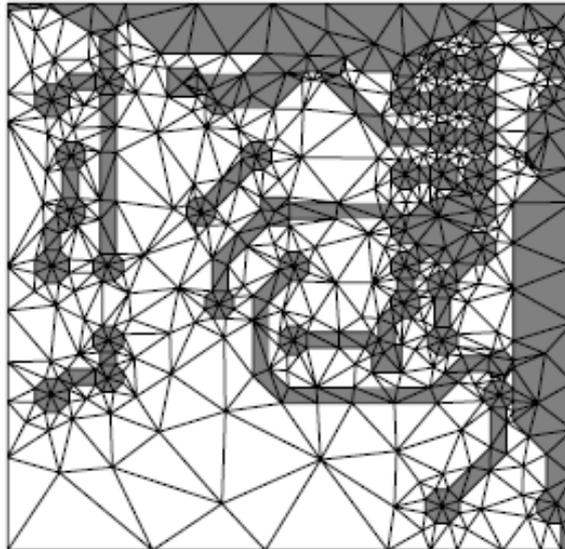


Figura 8 Descomposición heterogénea.

El proceso para llevar a cabo este mallado sería mediante iteraciones hasta obtener un mallado óptimo en toda la región a estudiar. Estas iteraciones se llevarán a cabo de forma que cuanto menor sea la distancia a un obstáculo, más fino deberá ser el mallado. Mientras menor tamaño tengan las celdas, mayor será la precisión del grafo, pero ello conlleva que el costo computacional sea muy elevado, es por ello que será necesario reconocer qué zonas necesitan un mallado más preciso y cuáles no, en la Figura 8 se observa un ejemplo de cómo sería la descomposición heterogénea.

En la Figura 9 se recoge todos los errores que se pueden cometer derivados de un mallado heterogéneo que no cumple las características de conformidad, buen diseño y respeto de los obstáculos.

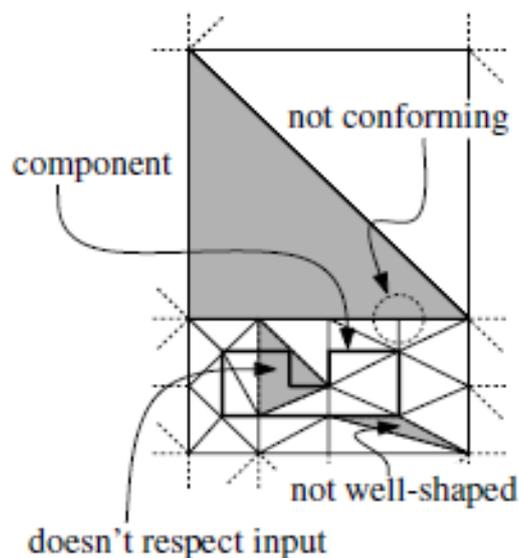


Figura 9 Errores a evitar a la hora de la realización del mallado.

Este método de generación de grafos proporciona como salida el centro de la figura, siendo en el caso del cuadrilátero la intersección de las diagonales y en el caso del triángulo el ortocentro. Todos estos puntos forman el subconjunto N. También nos proporciona qué nodos están conectados entre sí, es decir, el subconjunto L.

2.3. Grafo de visibilidad

Antes de pasar a la descripción del grafo de visibilidad es conveniente definir qué se entiende por visible en este proyecto. Un nodo es **visible** desde otro si la recta que une ambos nodos no intersecciona con ninguno de los obstáculos existentes.

El grafo de visibilidad ha sido uno de los métodos usados desde hace más tiempo a la hora de llevar a cabo el cálculo de trayectorias. Para este método se tiene que el subconjunto N está formado por todos los vértices de los obstáculos junto con los puntos inicial y final. Por su parte el subconjunto L estará formado por los segmentos que unen los vértices visibles entre sí. En la Figura 10 se recoge un ejemplo de grafo de visibilidad.

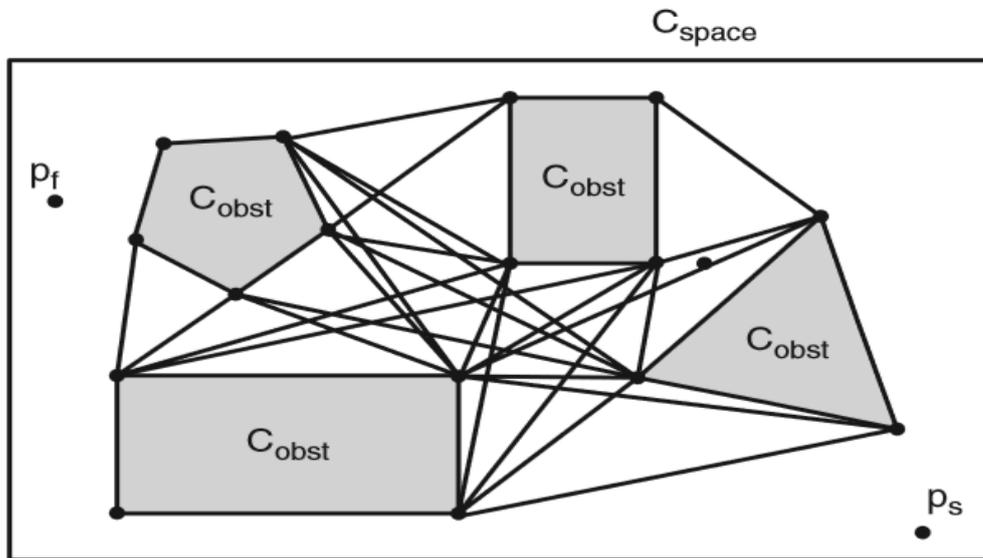


Figura 10 Grafo de visibilidad.

A mayor número de vértices se tiene que los subconjuntos N y L serán mayores, por lo tanto el coste computacional asociado a la resolución del problema será mayor. Es por ello que a la hora de introducir el escenario bajo estudio, ya sea de forma manual o como imagen, se debe tener en cuenta la precisión con la que se definen los obstáculos, por ejemplo, un obstáculo circular puede definirse con 50 o 350 nodos, influyendo esta decisión en el coste computacional.

Cabe destacar que este método no proporciona el camino óptimo, sino que los puntos de paso por ser los vértices de los obstáculos permiten que la solución sea óptima una vez empleado el algoritmo de búsqueda en grafos.

2.4. Elección de un algoritmo de generación de grafos

Una vez presentados todos los métodos estudiados se va a indicar cuál ha sido la opción a implementar para obtener la solución. Como se ha indicado con anterioridad, la intención de este proyecto es encontrar el camino más corto entre dos puntos, es decir, soluciones existirán múltiples pero solo estaremos interesados en una de ellas, a priori.

Anteriormente se han visto diferentes métodos los cuales nos proporcionaban puntos de paso entre los puntos inicial y final, pero no eran candidatos a ser caminos óptimos, es decir, había otros puntos de la región bajo estudio los cuales no eran candidatos, pero que proporcionaban soluciones de una menor distancia. En la Figura 11 se observa una comparación entre la descomposición trapezoidal y el grafo de visibilidad. En ella se observa que la descomposición trapezoidal proporciona un camino, pero éste dista de ser óptimo, y mientras más distancia haya entre los objetos, más distará la solución de ser la óptima. En cambio, se observa como el camino proporcionado por el grafo de visibilidad, línea discontinua, pasa por uno de los vértices, siendo más corto.

De los cuatro métodos planteados, sólo uno nos va a permitir la obtención de la ruta más corta, el del grafo de visibilidad. Además de esta cualidad que es la más importante tal y como hemos definido el problema y la solución deseada, presenta varias ventajas con respecto al resto de métodos. Por una parte, en general proporciona un menor número de nodos a estudiar con respecto al método de descomposición en celdas y por lo tanto, menor coste computacional. Por otra parte, es muy fácil obtener los nodos, los cuales son los vértices de nuestros obstáculos. También tiene algunas desventajas como puede ser el hecho de que la solución que proponga sea pasando muy cerca de los obstáculos. En realidad, no será una desventaja, pues proporcionándole la protección lateral adecuada, las aristas, en realidad, serán un recerido implementado con el fin de garantizar la seguridad de la aeronave.

Por lo tanto, como ya se ha comentado, se va a emplear el método del *grafo de visibilidad*.

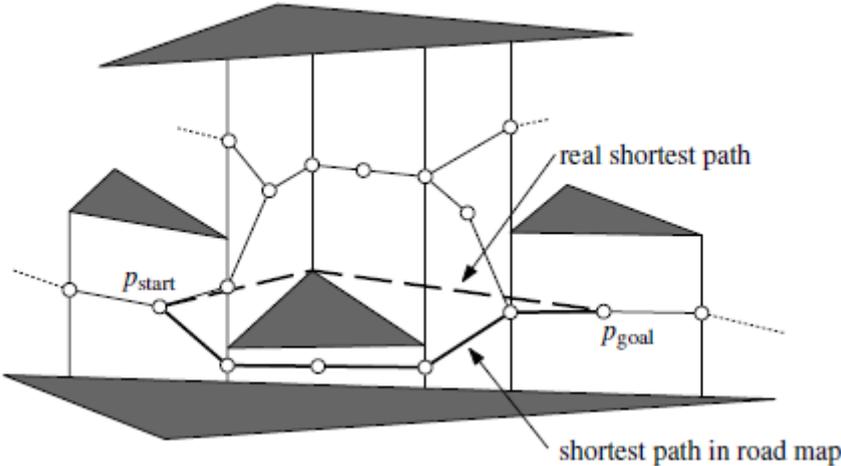


Figura 11 Comparación entre los métodos trapezoidales y grafos de visibilidad.

3 ALGORITMOS DE BÚSQUEDA EN GRAFOS

Los algoritmos de búsqueda en grafos nacen por la necesidad de crear un mecanismo de navegación autónoma, bien sea de robots, coches, o personajes en un videojuego. Algunos de los más conocidos son Dijkstra, A*, LPA*, Floyd-Warshall o D* [5], [6] y [7].

Si dos nodos están unidos, al viajar de uno a otro se considera sucesor al nodo al que nos movemos, y predecesor al nodo del que venimos. Además, normalmente existirá un coste vinculado al desplazamiento entre nodos. Un algoritmo de búsqueda en grafos tratará de encontrar un camino óptimo entre dos nodos como por ejemplo un camino que minimice el coste de desplazamiento, o el número de pasos a realizar.

La principal diferencia entre los algoritmos es la información que guardan acerca del grafo. Algunos de ellos no guardan información alguna, simplemente expanden la búsqueda desde el nodo inicial, hasta que se llega al nodo final, otros guardan el coste de viajar desde el origen hasta ese nodo, o incluso una estimación de lo prometedor que es un nodo para conducir el camino a su objetivo.

En esta memoria se van a describir los algoritmos Dijkstra, Best First Search y A*.

3.1. Algoritmo Dijkstra

El algoritmo de Dijkstra² [4], también llamado algoritmo de caminos mínimos, es un algoritmo para la determinación del camino más corto dados un punto inicial y otro final en un grafo con pesos en cada arista, siendo dichos pesos la distancia entre vértices. Su nombre se refiere a Edsger Dijkstra, quien lo describió por primera vez en 1959.

La idea subyacente en este algoritmo consiste en ir explorando los caminos que poseen una menor distancia desde el punto inicial hasta el resto de vértices; cuando se obtiene el camino más corto desde el vértice punto inicial al punto final el algoritmo se detiene. A continuación se muestra de forma somera como sería una iteración del método.

Sea un grafo de N nodos,

- 1) Se toma como nodo actual aquel nodo del vector cola, Q^3 , que tenga una menor distancia al nodo inicial.
- 2) Dicho nodo se elimina del vector cola, Q.
- 3) Se calcula la distancia a todos los nodos vecinos, los cuales no hayan sido visitados⁴.
- 4) Se comprueba si la distancia total al nodo inicial es menor que la actual. Si es así, se añade ese vecino a la cola, Q.
- 5) Marcamos el nodo actual como nodo visitado.
- 6) Volvemos al paso 1.

Este algoritmo se repetirá hasta que el nodo actual sea el nodo final.

Este algoritmo es el usado en algunas aerolíneas, por ejemplo Lufthansa, para elegir los puntos de paso entre dos aeropuertos.

Como principal inconveniente de este algoritmo tenemos que únicamente realiza búsquedas en anchura⁵. Por lo tanto puede requerir de más iteraciones para llevar a cabo la resolución del problema que otros métodos que

² http://es.wikipedia.org/wiki/Algoritmo_de_Dijkstra

³ El vector Q es un vector en el cual se encuentran los nodos que no han sido visitados y pueden ser visitados pues ya lo han sido alguno de sus vecinos. Inicialmente el vector Q está formado por el punto inicial.

⁴ Sea un nodo visitado aquel del cual se conoce su distancia al nodo inicial.

⁵ Búsqueda en anchura se refiere a que para calcular el camino óptimo el algoritmo no tiene en cuenta la distancia al nodo final, sino únicamente al nodo inicial.

además tengan búsquedas en profundidad como puede ser el A*, el cual se explica más adelante.

3.2. Algoritmo Best First Search

El algoritmo "Best First Search"⁶ elige el nodo no visitado con el menor valor heurístico para visitar. Heurístico se refiere a una regla de resolución de problemas en general o conjunto de normas que no garantizan la mejor solución pero sirve como guía útil para la resolución de problemas. Este algoritmo fue implementado por primera vez por Pearl en 1984. A continuación se va a mostrar una iteración del algoritmo:

Sea un grafo de N nodos,

- 1) Se toma como nodo actual aquel nodo del vector cola, Q^7 , que tenga una menor distancia al nodo final.
- 2) Dicho nodo se elimina del vector cola, Q .
- 3) Se calcula la distancia de todos los nodos vecinos, los cuales no hayan sido visitados, al nodo final.
- 4) Se comprueba si la distancia total al nodo final es menor que la actual. Si es así, se añade ese vecino a la cola, Q .
- 5) Marcamos el nodo actual como nodo visitado.
- 6) Volvemos al paso 1.

Este algoritmo se repetirá hasta que el nodo actual sea el nodo final. Como se ha comentado anteriormente, este algoritmo es funcional cuando se requiere de una primera aproximación a la solución óptima de un problema. Si bien el interés de este algoritmo reside en que combinado con el algoritmo de Dijkstra tiene una capacidad de resolución muy elevada.

Al igual que se ha comentado para el Dijkstra, las iteraciones para resolver el problema pueden ser más numerosas al realizar la búsqueda únicamente en profundidad.

3.3. Algoritmo A*

El algoritmo de búsqueda A*⁸ (pronunciado "A asterisco" o "A estrella") fue presentado por primera vez en 1968 por Peter E. Hart, Nils J. Nilsson y Bertram Raphael. El algoritmo A* encuentra, siempre y cuando se cumplan unas determinadas condiciones, el camino de menor coste entre un nodo origen y uno objetivo.

Para obtener la solución el algoritmo A* utiliza una función de evaluación $f(i) = g(i) + j(i)$, donde $j(i)$ representa el valor heurístico del nodo a evaluar desde el actual hasta el final, y $g(i)$, el coste real del camino recorrido para llegar a dicho nodo desde el nodo inicial. Siendo i el nodo bajo estudio.

El algoritmo A* mantiene dos estructuras de datos auxiliares. La primera de ellas la podemos denominar vector de cola de prioridad (ordenada por el valor $f(i)$ de cada nodo). La segunda de nodos cerrados, donde se guarda el nodo predecesor de los nodos que ya han sido visitados. En cada paso el algoritmo toma el nodo de la columna del vector de cola de menor valor, si éste no es el nodo final calcula la $f(i)$ de todos los nodos visitables desde él y los introduce en el vector cola, y pasando el nodo bajo estudio al vector de nodos cerrados.

El algoritmo es una combinación entre búsquedas del tipo anchura y profundidad: mientras que $j(i)$ es en profundidad, $g(i)$ es en anchura. De este modo, se cambia de camino de búsqueda cada vez que existen nodos más prometedores.

Las condiciones que se deben de dar para que el algoritmo proporcione el camino de menor distancia entre los nodos inicial y final es que la función $j(i)$ sea heurística admisible⁹. Para que se de esta situación se puede tomar por ejemplo como función heurística la distancia euclídea entre el nodo bajo estudio y el nodo final.

⁶ http://en.wikipedia.org/wiki/Best-first_search

⁷ El vector Q es un vector en el cual se encuentran los nodos que no han sido visitados y pueden ser visitados pues ya lo han sido alguno de sus vecinos. Inicialmente el vector Q está formado por el punto inicial.

⁸ es.wikipedia.org/wiki/Algoritmo_de_b%C3%BAsqueda_A*

⁹ Una función heurística se considera admisible si nunca se sobreestima el coste alcanzado.

3.4. Elección de un algoritmo de búsqueda en grafos

El problema de algunos algoritmos de búsqueda en grafos, como puede ser el algoritmo “Best-First-Search”, es que se guían en exclusiva por la función heurística, la cual puede no indicar el camino de coste más bajo, por lo que no nos interesa. Es por ello bastante intuitivo el hecho de que un buen algoritmo de búsqueda debería tener en cuenta ambos factores, el valor heurístico de los nodos y el coste real del recorrido.

Los algoritmos de búsqueda en grafos proporcionan únicamente una solución óptima, aunque existan dos posibles caminos los cuales necesiten de la misma distancia para llegar del nodo inicial al final. El algoritmo únicamente proporcionará el camino que calcule antes.

De los algoritmos estudiados para la búsqueda de grafos, se han implementado dos, el Dijkstra y el A*, aunque del que se hará uso para resolver el problema será el algoritmo A*, debido a que gracias a la búsqueda en profundidad, la cual no realiza el método de Dijkstra, obtiene de una manera más rápida la solución a nuestro problema. El tiempo empleado por ambos para un caso en el que se cuenta con 350 nodos es de 0.118 segundos para el algoritmo Dijkstra y de 0.045 segundos para el algoritmo A*.

Por lo tanto se va a emplear como ya se ha dicho el *algoritmo A**.

4 IMPLEMENTACIÓN. GRAFO DE VISIBILIDAD + ALGORITMO A*

Una vez planteado nuestro problema y elegido el método para resolverlo, se va a describir la implementación del código empleado para la resolución, así como la batería de casos, la cual muestra que se obtiene una solución correcta y el programa no contiene errores debidos a una desacertada programación. A continuación se muestra un ejemplo de llamada de la función VG, que es como se ha decidido llamar al programa:

```
>> VG (Ps, obs, numpuntos, Pf)
```

El programa posee cuatro variables de entrada.

- Ps: punto inicial.
- Pf: punto final.
- obs: en ella se encuentran todos los vértices de todos los obstáculos de nuestro problema.
- numpuntos: indica el número de vértices que posee cada obstáculo.

Un ejemplo de la sintaxis de las variables de entrada se expone a continuación.

- Ps=[0, 0];
- Pf=[10, 10];
- numpuntos= [3,3];
- obs= [1,1; 2,5; 5,0; 6,6; 8,3; 7,2];

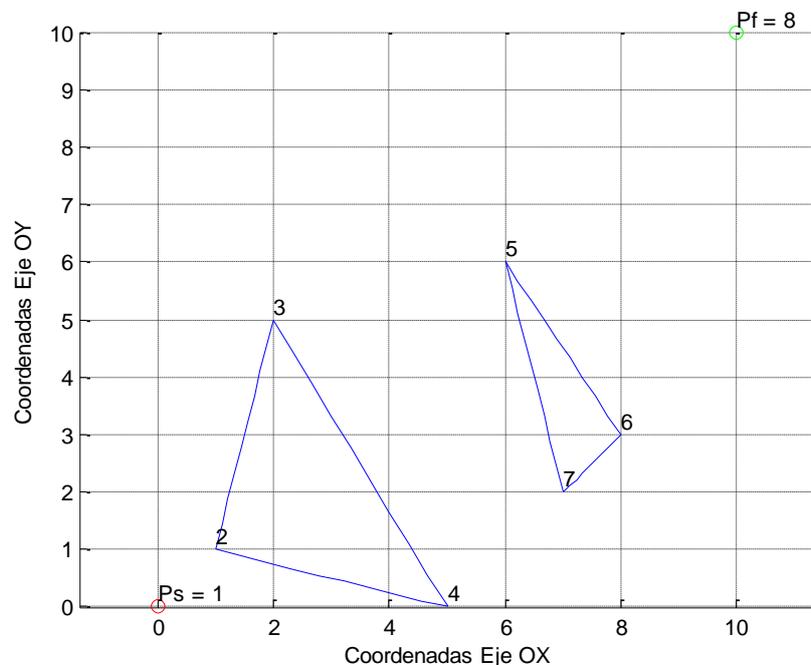


Figura 12 Ejemplo variables de entrada.

Con respecto a los argumentos de salida, el programa proporciona tres.

- El primero de ellos es una matriz que indica qué nodos son visibles desde cada uno de ellos. Esta matriz recibe el nombre de VISIBILIDAD y es equivalente al subconjunto L definido con anterioridad. En

cada columna j se tienen los índices de los nodos que son visibles desde el nodo j . Existen tantas columnas como nodos menos uno, pues no existe ningún interés en saber qué nodos son visibles desde el nodo final. La matriz tiene a lo sumo, $n-1$ filas, pues cada nodo puede ver como máximo a los otros $n-1$ nodos. Esta dimensión dependerá de cada escenario y vendrá dada por el nodo que tenga una mayor visibilidad.

- El segundo de los argumentos de salida es la ruta a seguir desde el nodo inicial al final. Esta ruta vendrá dada como un vector horizontal en el cual vendrán recogidos todos los puntos de paso, vértices, que habrá que recorrer.
- Por último la distancia que separa a ambos nodos, inicial y final, a través de la mencionada ruta.

En primer lugar se muestran los resultados para un único obstáculo en el área bajo estudio. Posteriormente un epígrafe con el método seguido para obtener la resolución del problema para varios obstáculos y por último, una modificación realizada en el código para ser capaces de obtener soluciones alternativas.

4.1. Resolución del Problema para un Único Obstáculo

La principal dificultad de la implementación del método de grafo de visibilidad reside en determinar qué nodos son visibles desde cada uno de ellos.

Para determinarlos se han diferenciado dos casos. Si el nodo bajo estudio es el punto inicial o no. Si se estudia un nodo perteneciente al obstáculo se tiene que sus vértices contiguos siempre van a ser visibles, por lo que solamente es necesario estudiar si el punto final es visible o no. Por su parte, si se está estudiando la visibilidad del punto inicial, habrá que estudiar si son visibles todos los vértices del obstáculo y el punto final.

Como se ha dicho antes, se ha desestimado estudiar los nodos “visibles” desde el nodo final, pues este es únicamente un nodo al que se debe llegar. Una vez que la aeronave ha llegado a él su misión ha concluido. Así mismo no se ha llevado a cabo el análisis de la visibilidad del nodo inicial desde otro nodo, pues no tiene sentido volver al nodo inicial en ningún caso.

A continuación se va a realizar una descripción de cómo se ha implementado el algoritmo para determinar la matriz de VISIBILIDAD.

- Para el nodo inicial (bajo estudio) se calculan los ángulos que forman el eje de coordenadas OX con el punto inicial y cada uno de los vértices. Se determina el ángulo mayor y menor, y para dichos vértices se calcula la recta que los une. En el caso de que el nodo bajo estudio sea perteneciente al obstáculo, los vértices con mayor y menor ángulo serían los contiguos.
- Una vez obtenida dicha recta se evaluarán todos los nodos para comprobar si son visibles o no. Si un nodo se encuentra entre la recta y el nodo que se está estudiando, entonces será siempre visible.
- Si por el contrario la recta analítica se encuentra entre el nodo bajo estudio y el nodo actual se debe estudiar el ángulo que forman. Si el ángulo del nodo bajo estudio es menor que los de los vértices con mayor y menor ángulo, entonces será visible, si es mayor que ambos también sería visible, si por el contrario su ángulo se encuentra comprendido entre ambos no sería visible por encontrarse por detrás del obstáculo.

Para los nodos contenidos en el mismo obstáculo, al considerarse los obstáculos convexos, se puede concluir que únicamente serán visibles los vértices contiguos.

Uno de los principales problemas que ha habido a la hora de resolver este problema ha sido el hecho de que a la hora de calcular los ángulos formados por los nodos mediante el software “Matlab”, este produce una singularidad en el punto $[-\pi, \pi]$, con lo cual hay que tener en cuenta la ubicación del obstáculo que se está estudiando y en qué cuadrantes se encuentra con respecto al nodo bajo estudio. Todo esto se va a explicar de una manera más clara y detallada con el ejemplo que se va a mostrar a continuación, aunque la solución sea obvia.

Conforme el software Matlab va llevando a cabo la resolución del problema, realiza una serie de operaciones, los resultados contienen un cierto error de precisión. Este error nos lleva a introducir una tolerancia en el código a la hora de calcular si un nodo está situado entre la recta y el nodo actual o no. Para la introducción de esta tolerancia se ha empleado la función “eps” existente en Matlab. Esta función tiene como entrada un valor y como salida proporciona la distancia existente al siguiente valor en Matlab. El argumento de entrada de “eps” será variable y su valor vendrá dado por la ecuación de la recta formada por los nodos cuyos ángulos sean el mayor

y menor.

- **Ejemplo 1**

En la Figura 13 se observa el camino para ir desde el nodo 1 hasta el nodo 6. Para obtener este camino se analiza la visibilidad desde cada uno de los nodos del problema.

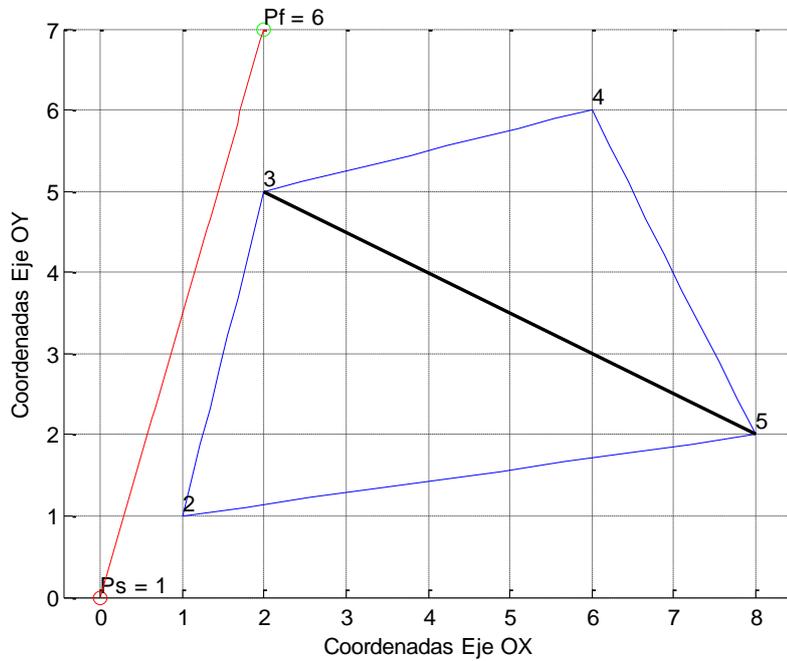


Figura 13 Un obstáculo. Ejemplo 1.

- En primer lugar se estudia el nodo inicial, 1.
 - Se calculan los vértices que forman un mayor y menor ángulo con respecto al eje OX, que son el nodo 3 y 5 respectivamente. Se puede determinar que estos vértices son visibles desde el nodo 1.
 - Para el nodo 2, se observa que se encuentra entre la recta formada por los nodos 3 y 5 (nodos con mayor y menor ángulo del obstáculo) y el nodo inicial, por lo tanto es visible.
 - Para el nodo 4 se tiene que la recta formada por los nodos 3 y 5 se encuentra entre el nodo inicial y él, por lo tanto se debe comprobar si el ángulo que forma con el eje OX está fuera del intervalo formado por los ángulos menor y mayor (5 y 3), al no estarlo se concluye que no es visible.
 - Por último, el nodo 6 se encuentra por detrás de la recta 3-5 por lo que al igual que el nodo 4 deberemos recurrir a comprobar si el ángulo formado con el eje OX se encuentra en el intervalo formado por los ángulos asociados a los nodos 3 y 5, al no ser así se puede determinar que es visible.
- Se pasa a estudiar el nodo 2. En primer lugar los nodos adyacentes son visibles, es decir el 3 y el 5. Con respecto al nodo 6 debemos comprobar qué posición relativa guarda con respecto a la recta formada por los nodos 3 y 5 (la recta está formada por los mismos vértices que para el nodo 1 debido a que son los adyacentes al nodo 2). La recta se encuentra entre ambos nodos, por lo tanto se debe estudiar el ángulo que forma el nodo 6 con respecto al eje OX. Se concluye que al estar fuera del intervalo formado por los valores de los ángulos mínimo y máximo, 3 y 5, el nodo 6 es visible.
- Con respecto al nodo 3, sus nodos adyacentes son el 2 y el 4, por lo que son visibles. En lo referido al nodo 6, la recta formada por los nodos 2 y 4 no se encuentra entre los nodos 3 y 6, por lo tanto el nodo 6 es visible.

- El nodo 4 tiene como nodos adyacentes a los vértices 3 y 5, por lo tanto son visibles. El nodo 6 se encuentra entre la recta formada por los vértices con un mayor y menor ángulo y el nodo 4, por lo tanto también es visible.
- Con respecto al nodo 5, tenemos que sus nodos adyacentes son los nodos 2 y 4, por lo tanto son visibles. En lo referido al nodo 6 se observa que éste se encuentra separado del nodo 5 por la recta formada por los nodos 2 y 4 lo que nos lleva a estudiar el ángulo que forma la recta 5-6 respecto al eje OX. Este ángulo está contenido en el intervalo formado por los ángulos asociados a los vértices contiguos, que en este caso son el 2 y el 4, por lo tanto el nodo 6 no es visible.

Se puede observar como la matriz de visibilidad tiene cinco columnas (n-1 nodos) y cuatro filas.

Nodos				
2	3	4	3	4
3	5	2	4	2
5	6	6	6	-
6	-	-	-	-

Tabla 1 Visibilidad para el ejemplo de un obstáculo. Ejemplo 1.

El programa nos indica que el camino más corto será pasando por los nodos uno y seis, y la distancia a recorrer será de 7.2801 unidades de longitud.

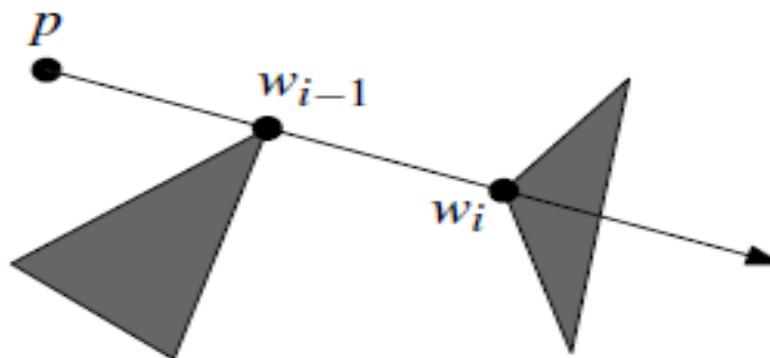


Figura 14 Visibilidad para nodos contenidos en la misma recta.

Otro hecho a resaltar es que si dos nodos están contenidos en la misma recta que el nodo bajo estudio, “borde de visibilidad”, el nodo más alejado no será visible. Esta decisión se ha llevado a cabo debido a que si ese camino realmente es el más corto, el pasar por el primero de los nodos no implica un aumento de la distancia a recorrer. Por ejemplo, en la Figura 14 el nodo w_i no es visible desde el nodo actual, p .

- Ejemplo 2

A continuación se va a presentar la resolución del problema para un obstáculo y puntos inicial y final cualesquiera, pero que tendrá especial interés para nuestro estudio por dos motivos. El primero es que la solución no va a ser una única recta que une el punto inicial y final, sino que la ruta pasará por el vértice del obstáculo. Para este ejemplo, se van a explicar con detenimiento el estudio de la visibilidad de los nodos 1 y 2, el resto se omitirá por ser prácticamente igual al caso del nodo 2.

- En este caso, a diferencia de lo que ocurría en el anterior, el ángulo formado por la recta que une el nodo 1 con el 6, con respecto al eje OX, está contenido en el intervalo formado por los ángulos máximo y mínimo (nodos 3 y 5), por lo tanto no será visible dicho nodo desde el nodo inicial. Con respecto al resto de nodos ocurre lo mismo que en casos anteriores, es decir, son visibles los nodos 2, 3 y 5.

- Con respecto al nodo 2, los nodos adyacentes, 3 y 5, son visibles. Con respecto al nodo 6, en primer lugar se observa que la recta que une a los vértices 3 y 5 se encuentra entre el nodo 2 y el 6. Por lo tanto, se calcula el ángulo formado por la recta 2-6 con respecto al eje OX y se observa que el valor del ángulo que forman está contenido entre los valores asociados a los nodos 3 y 5 concluyéndose que el nodo 6 no es visible.

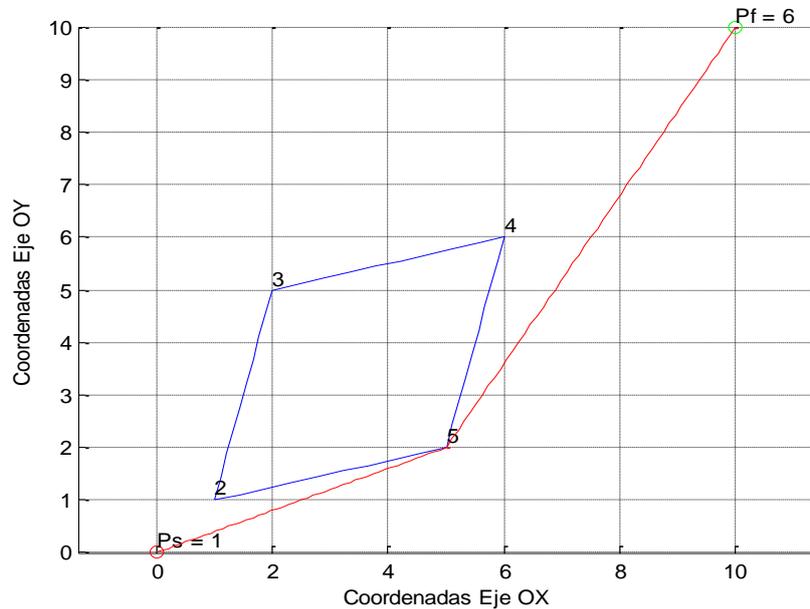


Figura 15 Un obstáculo. Ejemplo 2.

Nodos				
2	3	2	3	4
3	5	4	5	2
5	-	6	6	6

Tabla 2 Visibilidad para el ejemplo de un obstáculo. Ejemplo 2.

La ruta óptima proporcionada por el programa será aquella que recorra los puntos de paso 1, 5 y 6 siendo la distancia total empleada para dicho cometido de 14.8191 unidades de longitud.

- **Ejemplo 3**

Por último, en este apartado se presenta, en la Figura 16, una circunferencia. La razón de este caso reside en que debido al elevado número de vértices del obstáculo, puede plantear problemas desde el punto de vista del cálculo de ángulos y de singularidades. Se ha aproximado por un polígono de 50 vértices. Los nodos no se han etiquetado en esta ocasión debido a que por su proximidad, no se apreciaría la etiqueta.

Debido a la proximidad de los vértices, en muchos casos un nodo sólo ve a sus nodos adyacentes, es por ello que la ruta óptima va bordeando a la circunferencia hasta que llega a un vértice desde el cual el nodo final es visible. La solución suministrada por el programa indica que la distancia mínima a recorrer es de 10.2429 unidades de longitud y para ello se recorrerán los puntos de paso 1, 10, 9, 8, 7, 6, 52, siendo este último el nodo final.

Una vez planteado y resuelto el problema para un único obstáculo, falta llevar a cabo una batería de casos que muestre que el programa funciona correctamente. Para ello se ha analizado línea por línea el código, planteando

un caso para cada línea que pudiera generar problemas. Esta batería de casos se recoge en el Apéndice A.

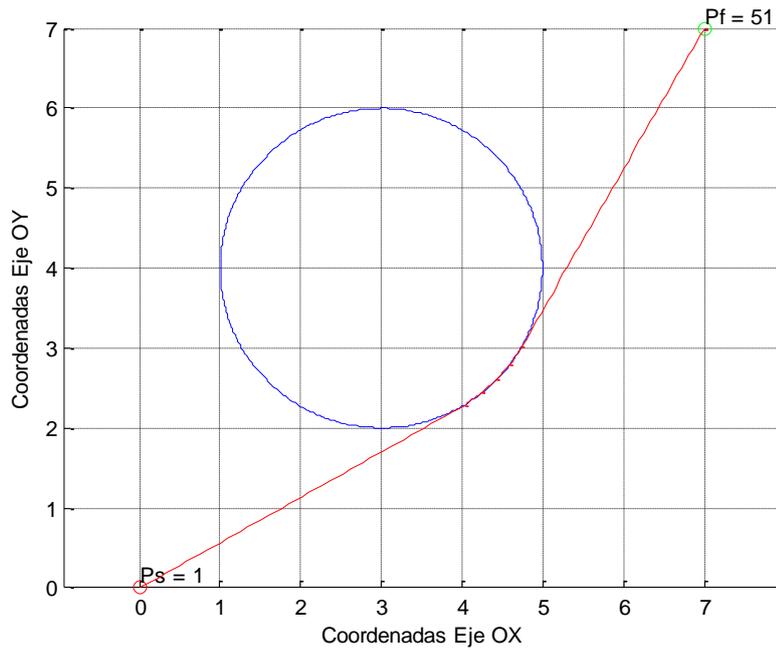


Figura 16 Un obstáculo. Ejemplo 3.

4.2. Resolución del Problema Para un Número Cualquiera de Obstáculos

Una vez explicados con detenimiento el caso de un único obstáculo, podemos pasar a la resolución del problema para un número cualquiera de obstáculos. Para esta sección el procedimiento para la resolución del problema es el mismo que en el caso de un obstáculo. Si hubiese alguna modificación se indicaría a lo largo de la sección.

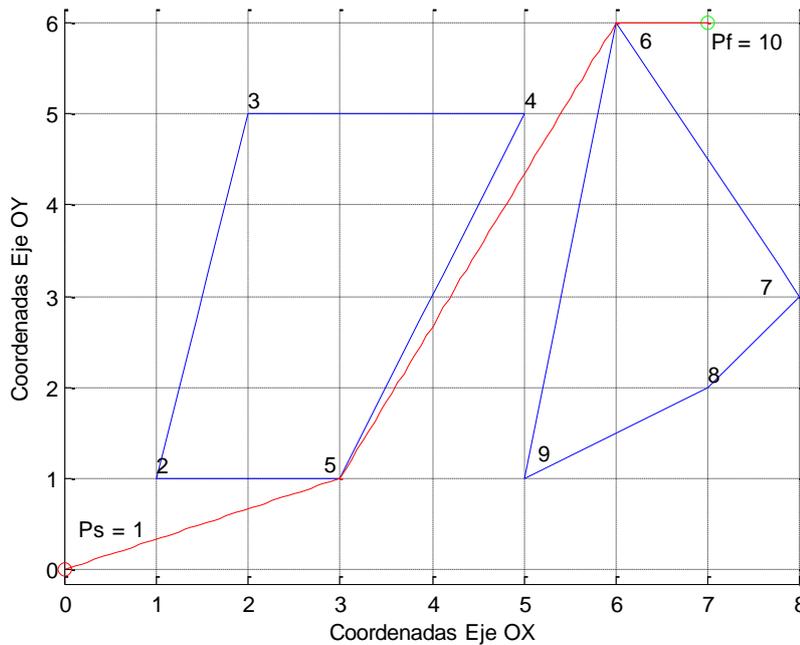


Figura 17 Varios obstáculos. Ejemplo 1.

La principal diferencia con respecto al caso de un único obstáculo reside en que anteriormente desde cada nodo perteneciente a un obstáculo solo se debía estudiar la visibilidad del punto final, mientras que ahora se debe estudiar la visibilidad de los nodos pertenecientes al resto de obstáculos. Esto conllevará un aumento del tiempo

computacional.

Para la resolución de este problema la mayor dificultad ha residido en estudiar la visibilidad desde cada nodo. Pues para cada uno de ellos se debe de estudiar los vértices de cada obstáculo que forman un mayor y menor ángulo con respecto a éste. Para el cálculo de estos ángulos había que tener en cuenta que algunos vértices podrían contener a la singularidad $[\pi, -\pi]$ y otros no. Este caso ha sido el más complejo de resolver, una vez resuelto el problema se ha abordado de una manera similar al caso de un obstáculo.

- **Ejemplo 4**

A continuación vamos a pasar a representar un posible problema. En nuestra zona bajo estudio existirán dos obstáculos, los cuales estarán formados por cuatro vértices cada uno de ellos.

En la Tabla 3 se recoge la matriz de VISIBILIDAD del problema.

Nodos								
2	3	2	3	2	7	6	7	6
3	5	4	5	4	9	8	9	8
5	-	6	6	6	3	10	-	2
9	-	-	9	9	4	-	-	4
-	-	-	-	-	5	-	-	5
-	-	-	-	-	10	-	-	-

Tabla 3 Visibilidad de nodos ejemplo varios obstáculos.

La solución propuesta por el programa recorre una distancia de 9.9932 unidades de longitud y para ello pasa por los puntos de paso 1, 5, 6, 10, siendo este último el nodo final.

Se ha expuesto un caso con dos obstáculos debido a la sencillez de obtener la ruta óptima de una manera visual y clara. Pero como se ha indicado, se puede llevar a cabo el problema con tantos obstáculos como se desee.

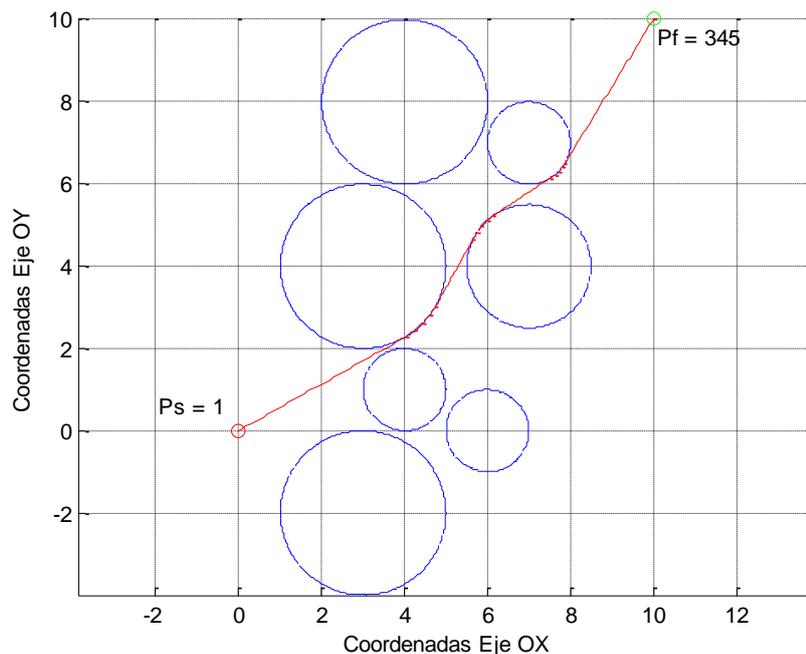


Figura 18 Varios obstáculos. Ejemplo 2.

- **Ejemplo 5**

Por proponer un caso más genérico, se va a hacer uso, al igual que en el epígrafe referido a un único obstáculo, de obstáculos circulares. En el problema planteado existen siete circunferencias en la zona bajo estudio que van a complicar el cálculo de la ruta óptima.

No se van a indicar las coordenadas de los obstáculos, así como la matriz de VISIBILIDAD debido a que estamos hablando de 350 nodos, por lo que tendríamos una matriz de tamaño considerable.

La solución propuesta por el programa recorre una distancia de 16.5351 unidades de longitud y para ello pasa por los puntos de paso 1, 226, 227, 228, 326, 327, 328, 58, 57, 56, 129, 130, 131, 9, 8, 7, 6, 345, siendo este último el nodo final.

Una vez planteado y resuelto el problema para varios obstáculos, falta llevar a cabo una batería de casos que muestre que el programa funciona correctamente. Para ello se ha analizado línea por línea el código, planteando un caso para cada línea que pudiera generar problemas. Esta batería de casos se recoge en el Apéndice B.

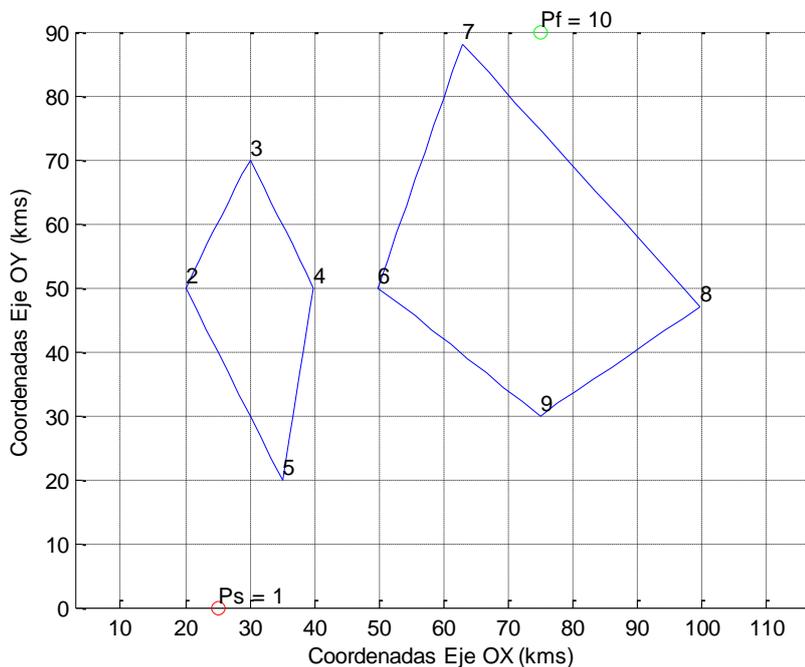


Figura 19 Cálculo de soluciones alternativas, paso 1.

4.3. Cálculo de Soluciones Alternativas. Algoritmo A*

Como se ha comentado anteriormente, se debe tener en cuenta que el algoritmo tal y como se ha descrito únicamente proporciona una solución. Sería interesante poder obtener varias soluciones por diversos motivos, como puede ser que se dé una circunstancia repentina que impida realizar dicha trayectoria, véase una tormenta u otra aeronave.

Para solventar el hecho de conseguir una única solución se ha pensado en que el avión no pudiese recorrer uno de los nodos por alguna circunstancia, por ejemplo las descritas en el párrafo anterior, entonces debería buscar otra ruta. Por ello lo que lleva a cabo el algoritmo es el cálculo de una nueva ruta eliminando la visibilidad de uno de los nodos por los que debía de pasar anteriormente. Es decir, la columna de la matriz VISIBILIDAD de dicho nodo se hace cero. En este caso el algoritmo A* llevaría a cabo el mismo proceso, pero no tendría uno de los nodos por los que pasaba antes, por lo que debe buscar una solución alternativa haciendo uso del resto de nodos.

Si la solución inicial está compuesta de x nodos, incluyendo el punto inicial y el final, esta modificación nos permite obtener $x-2$ soluciones alternativas.

Cabe destacar que las rutas obtenidas tendrán una distancia a recorrer igual o mayor que la calculada por el

algoritmo inicialmente. Por lo tanto éstas únicamente tendrán interés cuando la primera de ellas quede inoperativa.

A continuación vamos a estudiar un caso en el que sería necesario obtener una solución alternativa. En la Figura 19 se observa el problema a resolver. En ella se pueden visualizar dos obstáculos los cuales deben ser evitados.

Solución	Distancia	Ruta				
Original	108.0653	1	5	7	10	-
Alternativa 1	122.3655	1	2	3	7	10
Alternativa 2	138.2813	1	9	8	10	-

Tabla 4 Soluciones Alternativas. Distancia y Ruta.

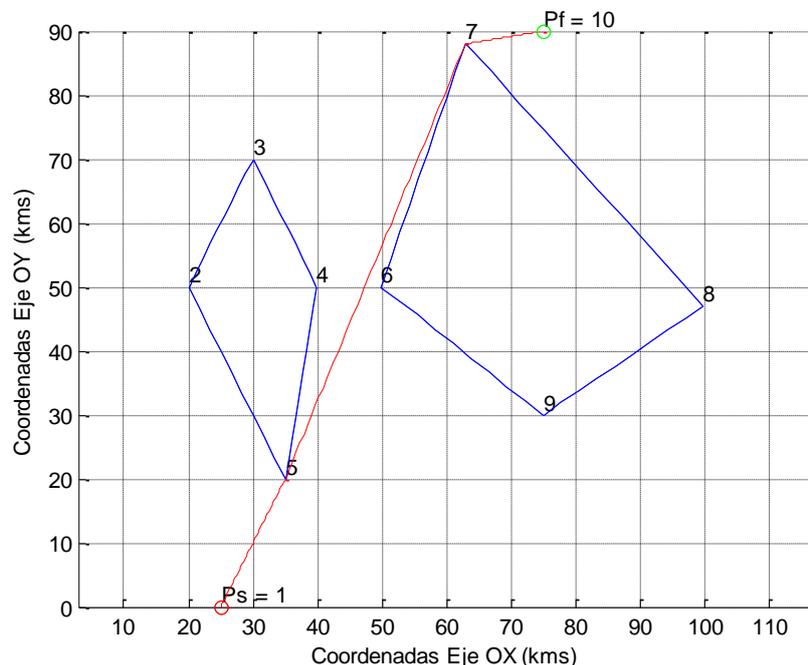


Figura 20 Cálculo de soluciones alternativas, paso 2.

La solución propuesta por el programa se recoge en la Figura 20 y en ella se puede observar que la trayectoria pasa por recorrer los nodos 1, 5, 7 y 9, lo cual conlleva pasar entre los dos obstáculos en una zona separada por diez kilómetros. La norma impediría dicha situación, por lo tanto sería conveniente generar una situación alternativa.

En la Tabla 4 se muestra tanto la solución inicial como dos posibles alternativas a dicha trayectoria inicial. La primera columna corresponde a la distancia para cada trayectoria y el resto a los nodos que se deben de correr para llegar a destino.

Por último, en la Figura 21 se muestra la solución alternativa 1 en la cual se puede observar como los nodos solución no pertenecen a la región situada entre ambos obstáculos.

Una vez terminado el estudio para obstáculos convexos, se va a pasar a abordar el problema para el caso de que los obstáculos tengan uno o varios ángulos cóncavos. Para ello se va llevar a cabo su descomposición en varios polígonos convexos pudiendo de este modo llevar a cabo la resolución del problema con todo el código desarrollado hasta el momento.

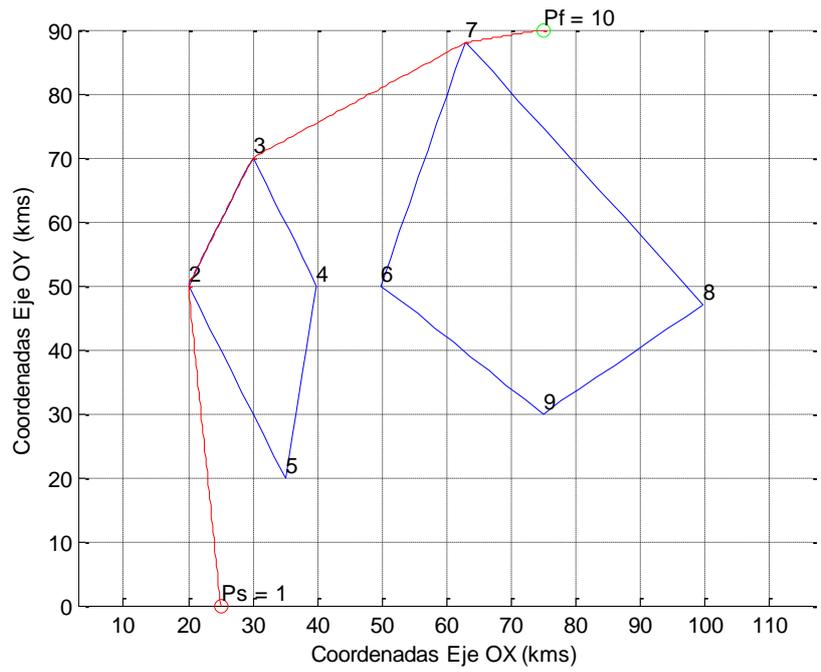


Figura 21 Cálculo de soluciones alternativas, paso 3.

5 CONVERSIÓN DE OBSTÁCULOS CÓNCAVOS EN CONVEXOS. “EAR CLIPPING”

Hasta este momento todos los obstáculos que se han presentado tenían una característica común, eran convexos. Nuestro procedimiento para determinar la visibilidad de un nodo se basa principalmente en determinar si el nodo bajo estudio se encuentra entre el nodo actual y la recta obtenida a partir de los vértices con mayor y menor ángulo del obstáculo con respecto al nodo bajo estudio. Si se cumplía esta condición, el nodo era visible. Para el caso de obstáculos cóncavos este hecho no es cierto. Un vértice puede encontrarse entre la recta calculada para un obstáculo y el nodo actual y no por ello ser visible. En la Figura 22 se puede ver con más claridad. Para el nodo cuatro, se observa cómo se encuentra situado entre el nodo inicial y la recta generada por los nodos de mayor y menor ángulo (nodos 3 y 5), y no por ello es visible.

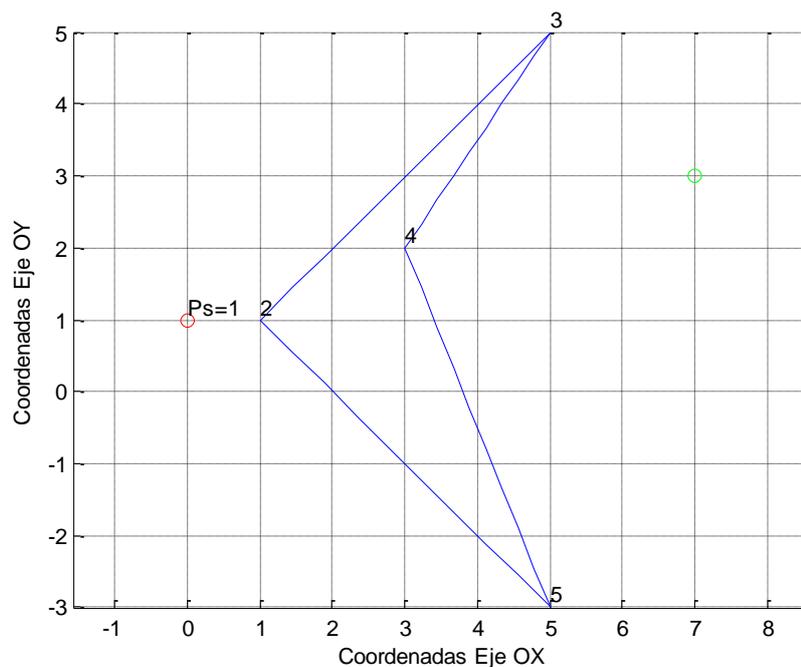


Figura 22 Ejemplo obstáculo cóncavo.

Por lo tanto, se debe buscar un método para ser capaz de convertir polígonos cóncavos en convexos. Se han encontrado diversos métodos para llevar a cabo esta operación.

1. Dividir el polígono desde el vértice cóncavo trazando su bisectriz.
2. Seccionar el polígono mediante la prolongación de una de las aristas del vértice cóncavo.
3. “Ear Clipping”.

En nuestro caso se ha decidido emplear el conocido como “Ear Clipping”¹⁰, el cual se basa en triangular el polígono. Esta decisión se basa principalmente en que es el método más genérico a priori.

Como se indicó en el epígrafe referido a las hipótesis, los obstáculos deben ser simples, es decir, si se recorre el perímetro del obstáculo en sentido horario, este siempre se encuentra a la derecha. En la Figura 1 se recoge un ejemplo de polígono simple y otro no simple. Esta hipótesis es necesaria para aplicar el método. Un polígono

¹⁰ Eberly, David, «Triangulation by Ear Clipping,» www.geometrictools.com, 2008.

simple siempre se podrá descomponer en $n-2$ triángulos, siendo n el número de aristas de dicho polígono.

5.1. Algoritmo de ejecución del “Ear Clipping”

Para explicar el funcionamiento del método es conveniente definir dos conceptos.

- “Ear of a polygon”. Se le da este nombre a un triángulo formado por tres vértices consecutivos en cuyo interior no se encuentra ningún otro vértice del polígono.
- “Ear tip”. Se le da este nombre al vértice del “Ear of polygon” que comparte aristas del polígono con los otros dos vértices.

En un polígono con más de cuatro vértices siempre existe al menos dos “ear of a polygon” que no se solapan. Este hecho nos lleva a plantear una triangulación mediante iteraciones. Una iteración del método sería como sigue.

- Se determina si los vértices son cóncavos o convexos. Se crea un vector C = [Los vértices cuyos ángulos sean convexos].
- De los ángulos convexos, se determinan cuales son “ear tip”. Se crea un segundo vector ET = [Los vértices que cumplan las características explicadas para que sean “Ear tip”]
- Se toma el primero de los vértices contenidos en el vector ET .
- Se genera el triángulo. Se elimina el vértice, Ear Tip, del obstáculo.
- Si el obstáculo sigue teniendo vértices cóncavos, o tiene $n > 3$, siendo n el número de aristas, se vuelve al paso 1, si por el contrario el obstáculo se ha convertido en un triángulo, el proceso de triangulación ha llegado a su fin (n se va actualizando en cada iteración, siendo el número de vértices del polígono una vez aplicado el “Ear Clipping”).

Un vértice convexo será convexo siempre, es decir, durante la triangulación no se convertirá en cóncavo en ningún momento, mientras que un vértice cóncavo, en un determinado momento de la triangulación se convertirá en convexo.

- Ejemplo 1

A continuación se va a presentar un ejemplo para clarificar todos los conceptos expuestos en los párrafos anteriores. Sea un polígono de diez vértices tal y como se recoge en la imagen de la izquierda de la Figura 23. Se observa que los vértices convexos son $C = [0\ 1\ 3\ 4\ 6\ 9]$, y de ellos los “ear tip” son, $ET = [3\ 4\ 6\ 9]$. Se elimina el vértice tres, con lo que el polígono se queda como la imagen de la derecha de la Figura 23. El triángulo formado será $T = [2\ 3\ 4]$.

Ahora se va a llevar a cabo una segunda iteración del método. En este momento nos encontramos con que el vector ET ha permanecido invariable, por lo tanto el siguiente vértice que se eliminará del obstáculo será el cuatro. En la Figura 24 se observa el obstáculo antes y después de eliminar dicho vértice. En este caso el triángulo formado será $T = [2\ 4\ 5]$.

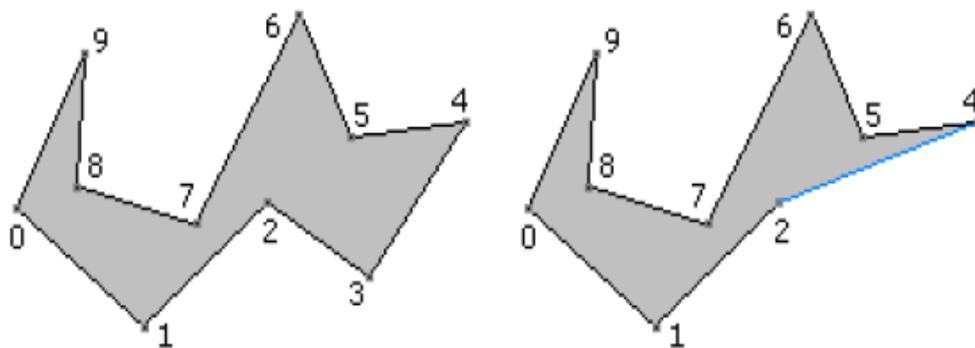


Figura 23 Ejemplo método “Ear Clipping”, paso 1.

Se seguirán llevando a cabo iteraciones con el fin de que en lugar de un obstáculo cóncavo, tengamos varios

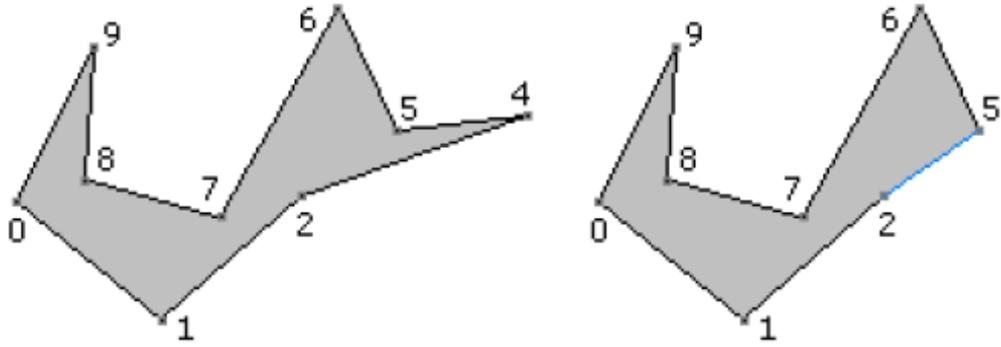


Figura 24 Ejemplo método "Ear Clipping", paso 2.

obstáculos convexos. En el tercer paso del ejemplo, Figura 25, se observa como el vértice 5, que antes era convexo, ahora es cóncavo. En este caso tenemos que el vector ET vendrá dado por los vértices [5, 6, 9] y el triángulo que se forma es $T = [2, 5, 6]$. En la Figura 25 se puede observar el estado del polígono antes y después de eliminar el mencionado triángulo.

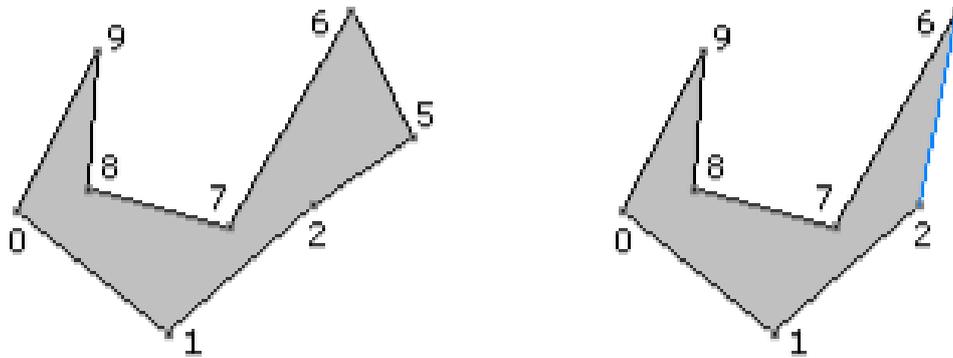


Figura 25 Ejemplo método "Ear Clipping", paso 3.

En la cuarta iteración se tiene como vector $ET = [6, 9]$, por lo tanto se toma el vértice 6 para llevar a cabo la triangulación, siendo de este modo el triángulo a eliminar del polígono $T = [2, 6, 7]$. El proceso mencionado se puede observar en la Figura 26.

En la Figura 27 de la izquierda se observa que únicamente quedan dos vértices cóncavos. El vector ET en este caso está formado por los vértices [2, 9]. Tomando el vértice 9 para realizar la triangulación tenemos que el triángulo obtenido es $T = [8, 9, 0]$. En la Figura 27 se puede observar el polígono tras la eliminación del triángulo indicado.

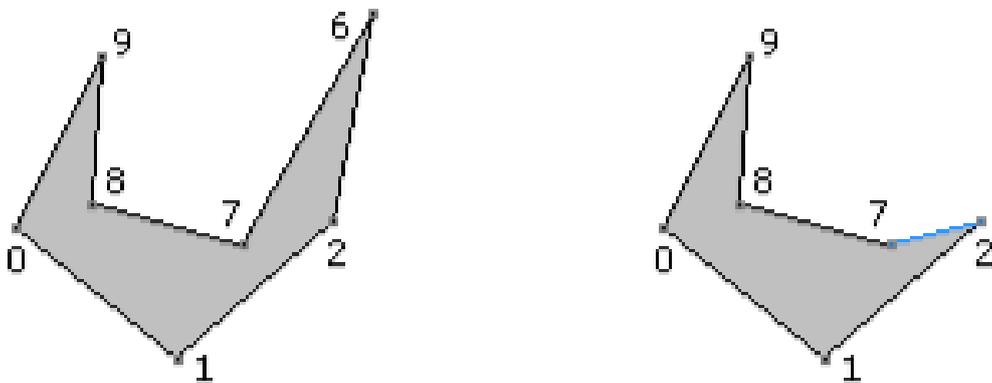


Figura 26 Ejemplo método "Ear Clipping", paso 4.

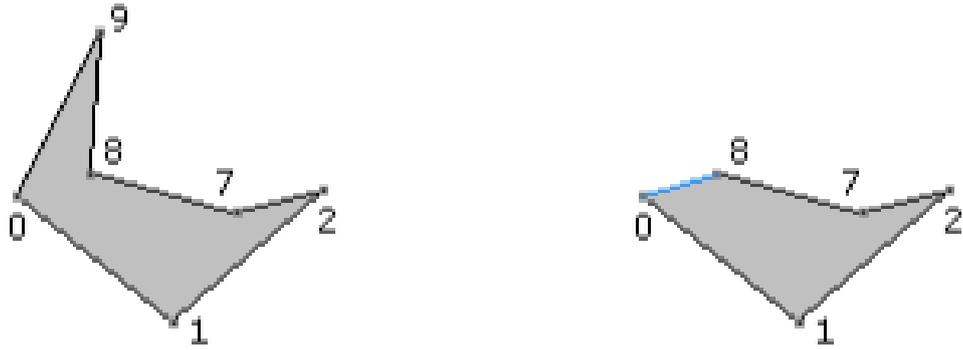


Figura 27 Ejemplo método "Ear Clipping", paso 5.

Llegados a este punto, el polígono que queda tras haber realizado las cinco iteraciones descritas anteriormente es un pentágono, por lo que aún se deberán de realizar dos iteraciones más. El vector "Ear Tipping" lo forman los vértices $ET = [0, 2, 8]$. En la Figura 28 se recoge el triángulo a eliminar de nuestro polígono que será el formado por los vértices $T = [8, 0, 1]$.



Figura 28 Ejemplo método "Ear Clipping", paso 6.

Nos queda una única iteración para completar el proceso de "Ear Clipping" y de este modo haber convertido el polígono cóncavo en ocho polígonos convexos (triángulos). En este caso el vector $ET = [2, 8]$ mientras que el triángulo a eliminar es $T = [1, 2, 7]$. En la Figura 29 se puede observar esta iteración. También se puede comprobar cómo el polígono inicial se ha visto reducido a un triángulo.



Figura 29 Ejemplo método "Ear Clipping", paso 7.

Para acabar con la explicación de este método, en la Figura 30 se observa el decágono inicial y los diferentes triángulos en los que se ha dividido.

En nuestro caso en vez de aplicarse el método hasta que únicamente se tengan triángulos, se ha decidido triangular hasta que el polígono sea convexo, no teniendo por ello que obtener un triángulo de la última partición. Por lo

que en algunos casos nos ahorraríamos las últimas iteraciones, siendo el costo computacional menor.

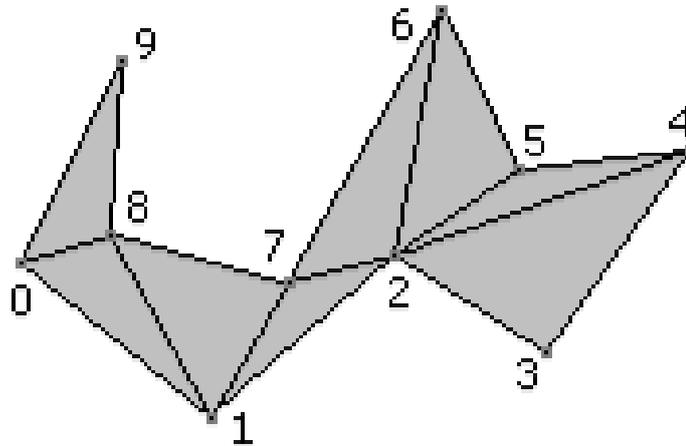


Figura 30 Decágono inicial y su correspondiente triangulación.

5.2. Resolución de un ejemplo mediante la utilización del programa implementado

- Ejemplo 2

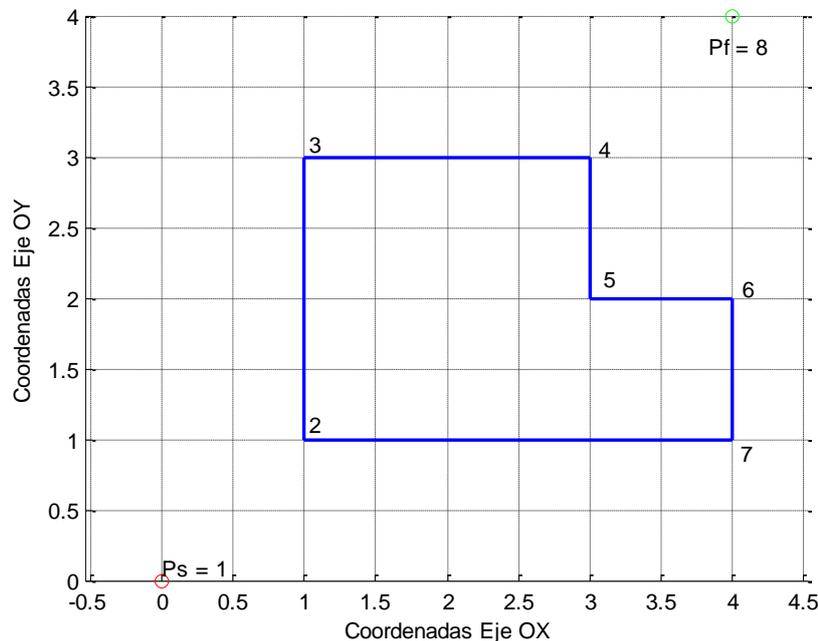


Figura 31 Obstáculo cóncavo. Problema inicial.

Ahora se va a exponer un ejemplo resuelto con el programa propuesto para tal fin. El obstáculo que se va a proponer tiene un vértice cóncavo, mientras que el resto de vértices serán convexos. En la Figura 31 se recoge dicho obstáculo mencionado, así como el punto inicial y el punto final. El obstáculo planteado es un hexágono en el que se puede ver como el vértice cóncavo es el designado con el número 5.

Una vez planteado el problema se observa en la Figura 32 cómo se ha llevado a cabo la triangulación el programa diseñado para tal fin. Se observa que de la forma que lo ha resuelto el programa se ha conseguido descomponer el hexágono en cuatro triángulos. En la Figura 32 también se puede ver cómo han sido re-etiquetados los nuevos

vértices del problema.

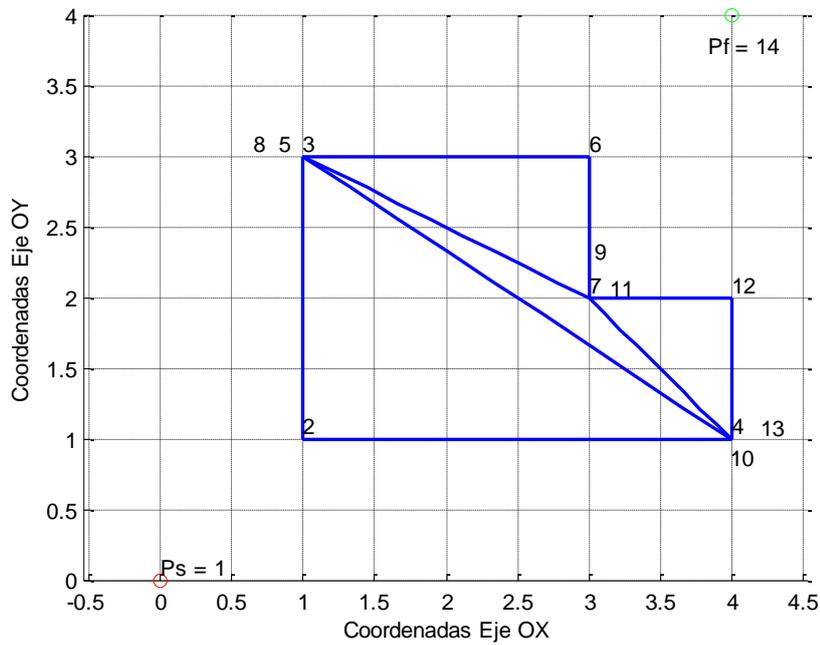


Figura 32 Obstáculo cóncavo. Trinagulación

Para finalizar con este ejemplo, se va a proceder a mostrar la Figura 33 en la que se recoge el obstáculo con la triangulación a la que se ha visto sometido, apareciendo los nuevos nodos etiquetados.

Para una mejor visualización del ejemplo y de que funciona como debe de hacerlo, se adjunta a continuación la Tabla 5 en la que se recoge los nodos visibles desde cada uno de los vértices reetiquetados.

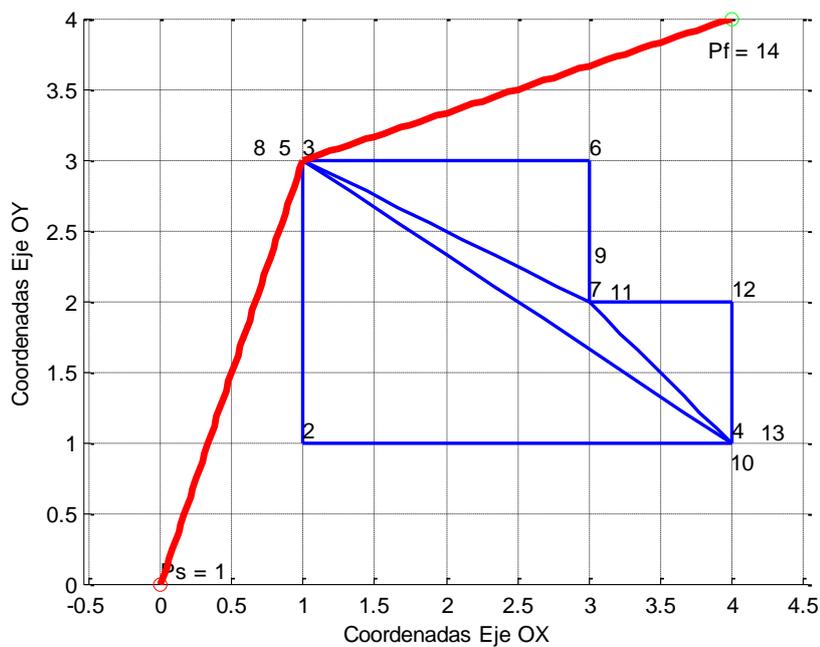


Figura 33 Obstáculo cóncavo. Solución final.

Nodos												
13	13	14	13	14	14	14	14	14	13	14	14	12
10	10	8	10	8	12	12	6	12	4	12	13	10
8	8	6	2	6	11	11	5	11	2	9	11	4
5	5	5	-	3	9	9	3	7	-	7	10	2
4	4	2	-	2	8	6	2	6	-	6	9	-
3	3	-	-	-	7	-	-	-	-	-	7	-
2	-	-	-	-	5	-	-	-	-	-	6	-
-	-	-	-	-	3	-	-	-	-	-	4	-

Tabla 5 Visibilidad para el ejemplo de un obstáculo cóncavo. Ejemplo 1.

La solución propuesta por el programa conlleva pasar por los vértices 1, 8 y 14, siendo la distancia necesitada para cubrir dicho camino de 6.3246 unidades de longitud.

La descomposición mediante el método “Ear Clipping” tiene varias soluciones, aunque ello no verá afectado al cálculo de la distancia a recorrer así como de los nodos por los que se habrá de pasar.

Cabe resaltar que la triangulación para llevar a cabo la descomposición de nuestro obstáculo, en caso de que sea cóncavo, puede producir diferentes soluciones. Ello dependerá de cómo estén enumerados los vértices inicialmente.

- **Ejemplo 3**

Una vez acabado con un ejemplo con un único obstáculo vamos a plantear un ejemplo en el que se simulará que nuestro vehículo deberá de recorrer un valle. Para llevar a cabo dicha simulación será necesario hacer uso de dos obstáculos. En la Figura 34 se muestran los dos obstáculos que nos servirán para simular el mencionado valle.

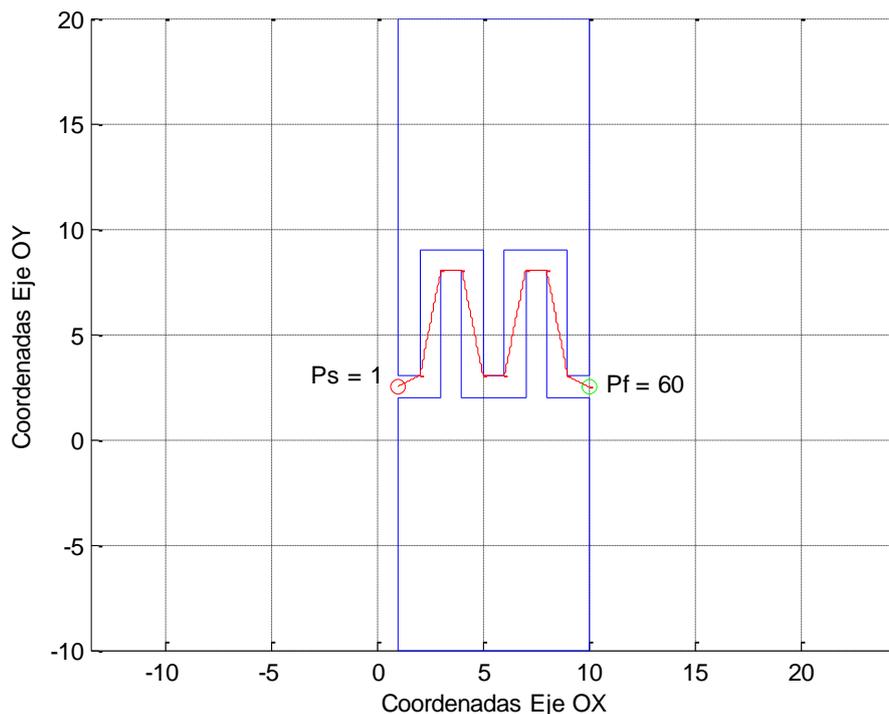


Figura 34 Obstáculo cóncavo. Valle. Ejemplo 2.

A diferencia de en el caso anterior, debido a la cantidad de nuevos vértices generados para convertir el obstáculo cóncavo en convexo, se ha considerado no indicarlos y de este modo apreciar la solución de una mejor manera gráficamente. En el caso de representar las etiquetas, sería imposible ver la solución con claridad.

Como se ha indicado existe un elevado número de vértices, motivo por el cual tampoco se va a mostrar la tabla con la visibilidad de cada nodo. La solución dada por el problema nos indica que la distancia a recorrer será de 25.6321 unidades de longitud y para ello habrá que sobrevolar los vértices 1, 25, 5, 6, 22, 21, 9, 10, 18 y 26, que es el nodo final.

Al igual que para el resto de casos será necesario llevar a cabo una comprobación de que el funcionamiento del programa es el correcto, para ello se ha revisado todo el código y se ha confeccionado una batería de casos, la cual se recoge en el Apéndice C.

6 IMPLEMENTACIÓN DE PROCEDIMIENTOS RNAV

En el capítulo 4, se ha llevado a cabo la implementación de un algoritmo el cual es capaz de generar el camino más corto entre dos puntos. Para ello se hacía uso del método “Grafo de Visibilidad” y el algoritmo A*, y en caso de que alguno de los obstáculos fuese cóncavo se llevaba a cabo una triangulación del obstáculo hasta reducir el mismo en varios polígonos convexos mediante el método “Ear Clipping”, explicado en el capítulo 5.

Una vez obtenido el camino más corto, se debe procurar que el mismo sea volable, este concepto se desarrolló en la Introducción y hacía mención a que la aeronave tiene que ser capaz de recorrerlo. Para ello se deben tener en cuenta dos aspectos:

- La longitud mínima de estabilización: es la mínima distancia que puede separar a dos “waypoints” de forma que la aeronave pueda recorrerlos.
- La protección lateral de la aeronave: es la distancia que debe existir entre la trayectoria de la aeronave y los obstáculos para evitar una colisión.

Para nuestro trabajo se ha empleado el documento 8168 de la OACI. Toda esta normativa está desarrollada para aviones tripulados.

6.1. Longitud Mínima de Estabilización

Esta distancia tendrá en cuenta la mínima separación entre puntos de recorrido sucesivos con el fin de evitar que éstos estén tan próximos que impidan la estabilización de la aeronave [2].

6.1.1. Normativa

Se consideran dos tipos de puntos de recorrido.

- Puntos de recorrido de paso.
- Puntos de recorrido de sobrevuelo.

La normativa tenida en cuenta para esta sección, recogida en el documento 8168 de la OACI, se basa en estudios teóricos combinados con resultados de simulaciones. El objetivo de esta sección no es determinar un área de protección sino determinar una distancia mínima entre dos puntos de recorrido en una trayectoria nominal. Por este motivo, en los cálculos teóricos no se tienen en cuenta el efecto del viento ni las tolerancias de los puntos de recorrido. Si fuese necesario podrían seleccionarse valores mayores.

Para cada punto de recorrido se determina una distancia mínima de estabilización. Se trata de la distancia entre el punto de recorrido y el punto en que la trayectoria se une tangencialmente a la derrota nominal. Para puntos de recorrido sucesivos, la distancia mínima entre ellos es la suma de ambas distancias de estabilización.

Estas distancias mínimas de estabilización dependen de cuatro características.

- Tipo de punto de paso.
- Ángulo de cambio de rumbo.
- Velocidad verdadera.
- Ángulo de balance.

A continuación se van a presentar una serie de ejemplos de cuál sería la distancia mínima de tramo en función del tipo de punto de recorrido que lo contenga. Se denotará el primer punto de recorrido como WP1 y el segundo

como WP2 para todos los casos que se explican.

- Dos puntos de recorrido de paso. Conocida la distancia mínima de estabilización para el punto 1, $A1$, y el punto 2, $A2$, se tiene que la distancia mínima de tramo es $A1+A2$. Este ejemplo se puede apreciar con mayor claridad en la Figura 35.

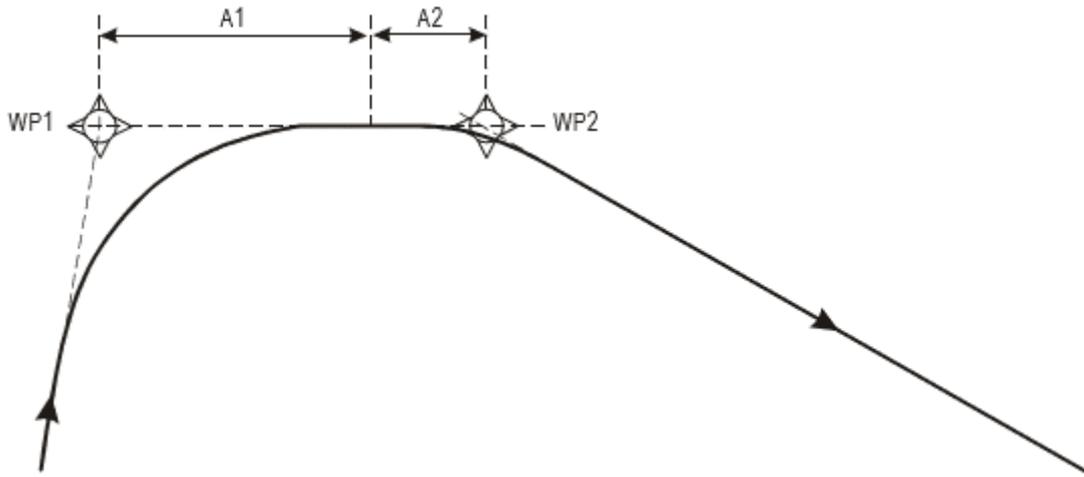


Figura 35 Dos puntos de recorrido de paso.

- Punto de recorrido de paso seguido de un punto de recorrido de sobrevuelo. En este caso, puesto que el segundo punto es de sobrevuelo, la distancia mínima de tramo será $A1+0=A1$.

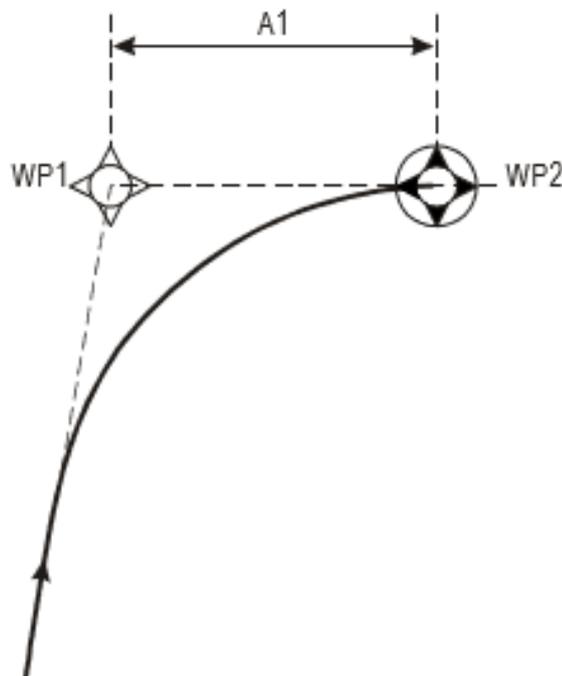


Figura 36 Punto de recorrido de paso seguido de un punto de recorrido de sobrevuelo.

- Dos puntos de recorrido de sobrevuelo. En este caso al igual que en el anterior al ser el segundo punto de recorrido de sobrevuelo, se tiene que la distancia mínima será $B1+0=B1$.

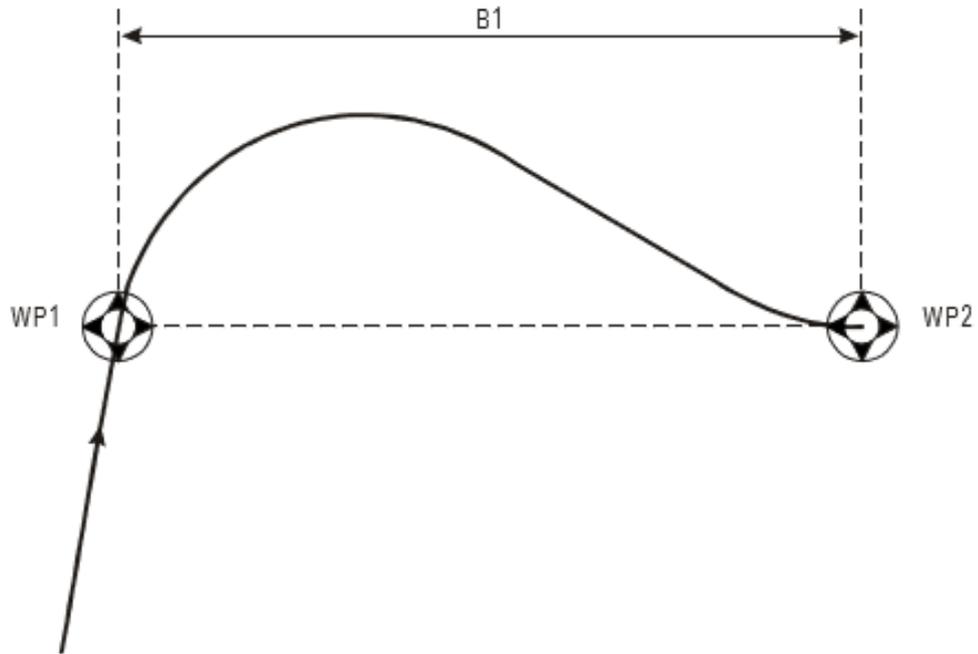


Figura 37 Dos punto de recorrido de sobrevuelo.

- Punto de recorrido de sobrevuelo seguido de un punto de recorrido de paso. Se determina la mínima distancia de estabilización tanto para el primer punto como para el segundo, siendo la distancia mínima de tramo $B1+A2$.

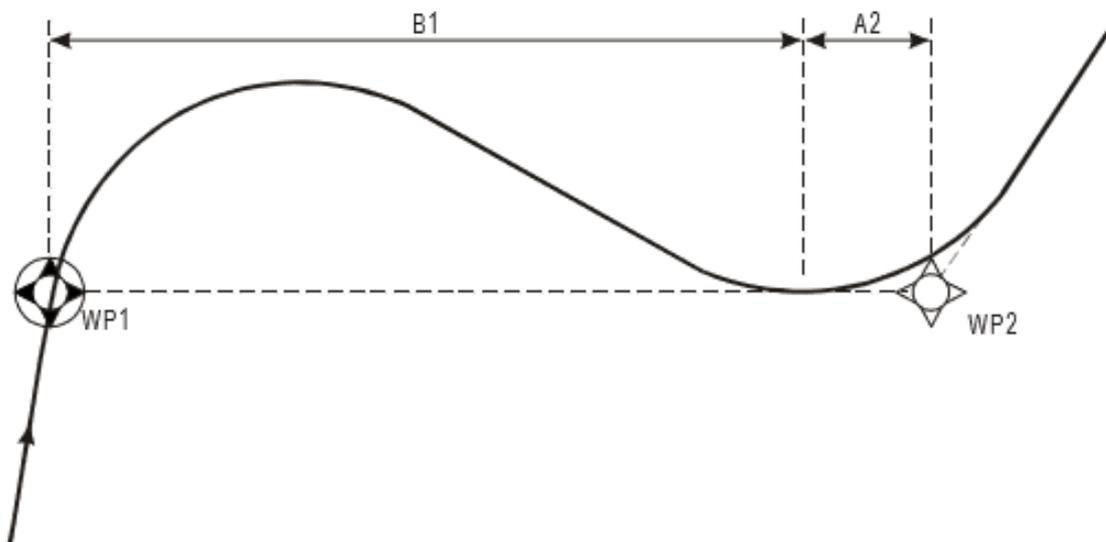


Figura 38 Punto de recorrido de sobrevuelo seguido de un punto de recorrido de paso.

En nuestro problema a resolver, únicamente se van a emplear criterios en ruta, este hecho nos permite simplificar la resolución del mismo, haciendo uso de los procedimientos recogidos en la Parte III- Sección 3, Capítulo 8. En dicho capítulo se indica a su vez que en nuestro caso los puntos de recorrido de paso serán los únicos permitidos en operaciones RNAV.

Por último, también se indica que para los virajes se use el método de las espirales de viento o los círculos limitadores, los cuales se detallarán en el apartado siguiente. Se menciona en este apartado ya que para criterios

en ruta, en la Parte I-Sección 2, Capítulo 3 se indica que el ángulo de balance que debe tenerse en cuenta será de 15°.

<i>Cambio de rumbo** (grados)</i>	<i>Velocidad verdadera (km/h)</i>						
	≤ 130	150	170	190	210	230	240
30	0,29	0,34	0,38	0,44	0,52	0,61	0,65
35	0,33	0,38	0,43	0,49	0,58	0,68	0,73
40	0,36	0,41	0,47	0,54	0,65	0,76	0,82
45	0,39	0,45	0,52	0,60	0,71	0,84	0,90
50	0,43	0,50	0,56	0,65	0,78	0,92	0,99
55	0,47	0,54	0,61	0,71	0,85	1,00	1,08
60	0,51	0,58	0,66	0,77	0,92	1,09	1,18
65	0,55	0,63	0,72	0,83	1,00	1,18	1,28
70	0,59	0,68	0,77	0,90	1,08	1,28	1,38
75	0,64	0,74	0,83	0,97	1,17	1,38	1,50
80	0,69	0,79	0,90	1,05	1,26	1,50	1,62
85	0,74	0,85	0,97	1,13	1,36	1,62	1,75
90	0,80	0,92	1,04	1,22	1,47	1,74	1,89
95	0,86	0,99	1,13	1,32	1,59	1,89	2,05
100	0,93	1,07	1,22	1,42	1,72	2,04	2,22
105	1,01	1,16	1,32	1,54	1,86	2,22	2,40
110	1,09	1,26	1,43	1,67	2,02	2,41	2,62
115	1,19	1,37	1,56	1,82	2,21	2,63	2,85
120	1,30	1,50	1,70	1,99	2,42	2,88	3,13

* 15° o 3°/seg

** Utilícese el valor de 30° para cambios de rumbo inferiores a 30°.

Tabla 6 Distancia mínima de estabilización entre puntos de recorrido de paso (bajas velocidades).

En las Tablas 6 y 7 se recoge la mencionada distancia mínima de estabilización en función del ángulo de balance, el cambio de rumbo y velocidad. Aunque nuestro programa podría generar trayectorias con un cambio de rumbo mayor de 120 ° se ha decidido no considerarlas como soluciones del problema debido a que en el Documento 8168 no las consideran.

<i>Cambio de rumbo* (grados)</i>	<i>Velocidad verdadera (km/h)</i>																
	$\alpha =$	240	260	280	300	320	340	360	380	400	440	480	520	560	600	640	
50		1,1	1,3	1,5	1,6	1,8	2,1	2,3	2,5	2,7	2,7	3,3	3,8	4,4	5,1	5,8	6,5
55		1,2	1,4	1,6	1,8	2,0	2,2	2,5	2,7	3,0	3,0	3,6	4,2	4,9	5,6	6,3	7,2
60		1,3	1,5	1,7	1,9	2,2	2,4	2,7	3,0	3,3	3,9	4,6	5,3	6,1	6,9	7,8	8,8
65		1,4	1,6	1,9	2,1	2,4	2,6	2,9	3,2	3,5	4,2	5,0	5,8	6,6	7,6	8,6	9,6
70		1,5	1,8	2,0	2,3	2,5	2,8	3,2	3,5	3,8	4,6	5,4	6,3	7,2	8,2	9,3	10,3
75		1,6	1,9	2,2	2,4	2,8	3,1	3,4	3,8	4,2	5,0	5,9	6,8	7,8	8,9	10,1	11,2
80		1,8	2,0	2,3	2,6	3,0	3,3	3,7	4,1	4,5	5,4	6,3	7,4	8,5	9,7	11,0	12,2
85		1,9	2,2	2,5	2,8	3,2	3,6	4,0	4,4	4,9	5,8	6,9	8,0	9,2	10,5	11,9	13,3
90		2,0	2,3	2,7	3,1	3,5	3,9	4,3	4,8	5,3	6,3	7,4	8,7	10,0	11,4	12,9	14,4
95		2,2	2,5	2,9	3,3	3,7	4,2	4,7	5,2	5,7	6,8	8,1	9,4	10,8	12,4	14,0	15,6
100		2,3	2,7	3,1	3,6	4,0	4,5	5,0	5,6	6,2	7,4	8,7	10,2	11,8	13,4	15,2	16,9
105		2,5	2,9	3,4	3,9	4,4	4,9	5,5	6,1	6,7	8,0	9,5	11,1	12,8	14,6	16,6	18,5
110		2,7	3,2	3,7	4,2	4,7	5,3	5,9	6,6	7,3	8,7	10,3	12,1	13,9	15,9	18,1	20,2
115		3,0	3,5	4,0	4,6	5,2	5,8	6,5	7,2	7,9	9,5	11,3	13,2	15,2	17,4	19,8	22,2
120		3,3	3,8	4,4	5,0	5,7	6,4	7,1	7,9	8,7	10,5	12,4	14,5	16,7	19,1	21,7	24,3

* Utilícese el valor de 50° para cambios de rumbo inferiores a 50°.

Tabla 7 Distancia mínima de estabilización entre puntos de recorrido de paso (altas velocidades).

Otro aspecto a resaltar para el cálculo de la mínima distancia de estabilización es que para poder simplificar la resolución del problema, se ha supuesto, como se indicó en las hipótesis del primer capítulo, que la aeronave recorrerá la trayectoria generada a velocidad constante, en nuestro caso se tomará la mínima proporcionada por

las Tabla 7, es decir, 130 km/h.

Con toda esta información será posible realizar un algoritmo el cual una vez calculado el camino más corto entre dos puntos, inicial y final, estudie las distancias entre cada uno de los nodos que componen la solución, modificando dichos puntos de paso si fuera necesario de forma que cumplan con los requisitos de la distancia mínima de tramo RNAV.

6.1.2. Algoritmo de Generación de Mínima Distancia de Tramo RNAV

Se va a proceder a continuación a describir el algoritmo usado en esta sección. En primer lugar, es necesario resaltar que la sección dedicada al cálculo de soluciones alternativas va a ser útil en este apartado. El algoritmo analiza la solución óptima y las modificaciones que sería necesario realizar a esta trayectoria para que pudiese ser recorrida por la aeronave. Si esto no fuese posible, entonces se analiza una de las soluciones alternativas, y así hasta que hubiese una que cumpliera los requisitos exigidos por la normativa.

Destacar también que dada una solución con n puntos de recorrido de paso, incluyendo el punto inicial y final, se estudiarán $n-2$ puntos, dado que la distancia entre el punto de paso $n-1$ y el nodo final no puede verse modificada, al ser el nodo final aquel punto que se desea alcanzar.

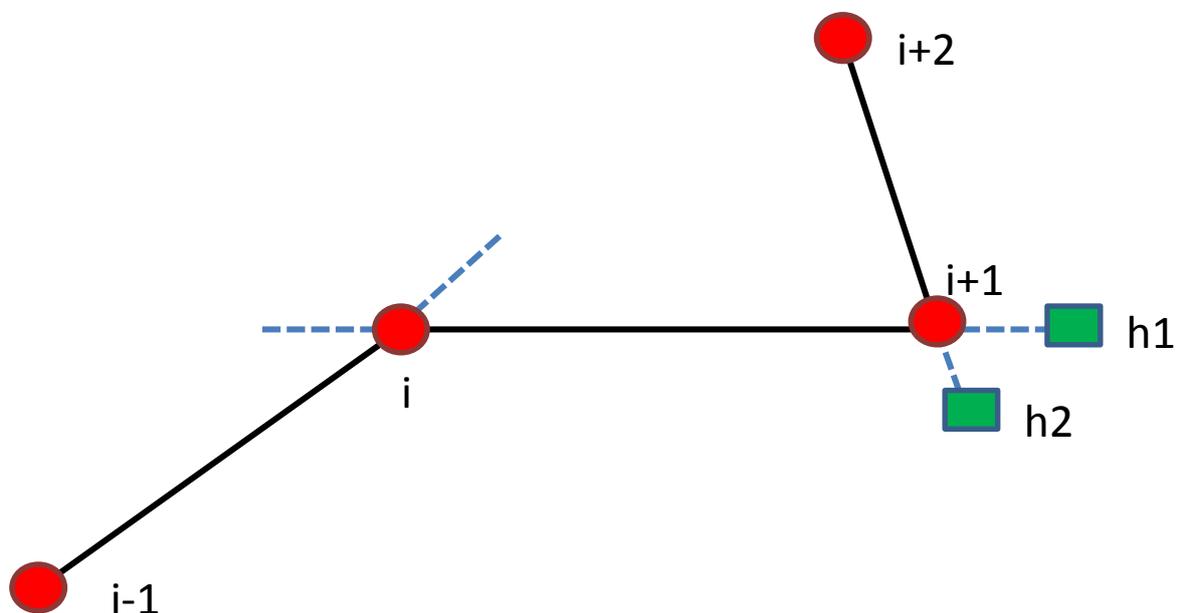


Figura 39 Ejemplo del algoritmo de Distancia mínima de estabilización.

A continuación se muestra el funcionamiento del algoritmo desarrollado:

- Dado un punto de recorrido cualquiera, i . Se calcula el cambio de rumbo en dicho punto y en el punto de recorrido siguiente, $i+1$, proporcionándonos las distancias de estabilización correspondientes. La mínima distancia de tramo RNAV será la suma de dichas distancias de estabilización. Se calcula la distancia que separa ambos puntos de paso, i e $i+1$. Si dicha distancia es mayor que la distancia mínima de tramo, se pasa al estudio del punto de paso siguiente, $i+1$, si por el contrario dicha distancia es menor, entra en funcionamiento el algoritmo “rnav”, que es el código que se ha implementado para cumplir con la mínima distancia de estabilización.
- Se calcula el punto, $h1$, situado en la recta que une los puntos de recorrido i e $i+1$ tal que la distancia que lo separe del nodo i sea igual a la distancia mínima de estabilización. A su vez se calcula el punto, $h2$, situado en la recta que une los puntos $i+1$ e $i+2$ de forma que la distancia entre el punto $h2$ y el nodo i sea la mínima distancia de estabilización.
- Para cada uno de ellos, $h1$ y $h2$, se calcula si la recta que une al punto de paso i con cada uno de ellos intersecciona con alguno de los obstáculos de nuestro problema. Si no intersecciona, dicho punto pasará a ser candidato para ser solución de nuestro problema.

- Una vez comprobado el punto anterior, se verifica que el punto candidato a ser solución es visible desde el punto de recorrido de paso siguiente, $i+2$, ello nos llevaría a concluir que se cumple la mínima distancia de estabilización en caso de que el punto $i+1$ fuese sustituido por el punto candidato a ser solución del problema. En caso de que tanto el punto $h1$ como el $h2$ sean candidatos a ser solución del problema, siempre se toma el punto $h1$.
 - Pasamos a comprobar el punto de paso $i+1$.
 - Si la solución no cumple los requisitos de distancia mínima, entonces se estudia la trayectoria en sentido contrario, es decir, desde el punto final al inicial, ya que esto permite que los puntos $h1$ y $h2$ sean distintos a los estudiados en el caso inicial, pudiendo arrojar una solución satisfactoria, que inicialmente no era factible.
 - Si tampoco existe solución estudiando el camino inverso, entonces se pasa a otra de las soluciones alternativas, la de menor distancia a recorrer.
- **Ejemplo 1**

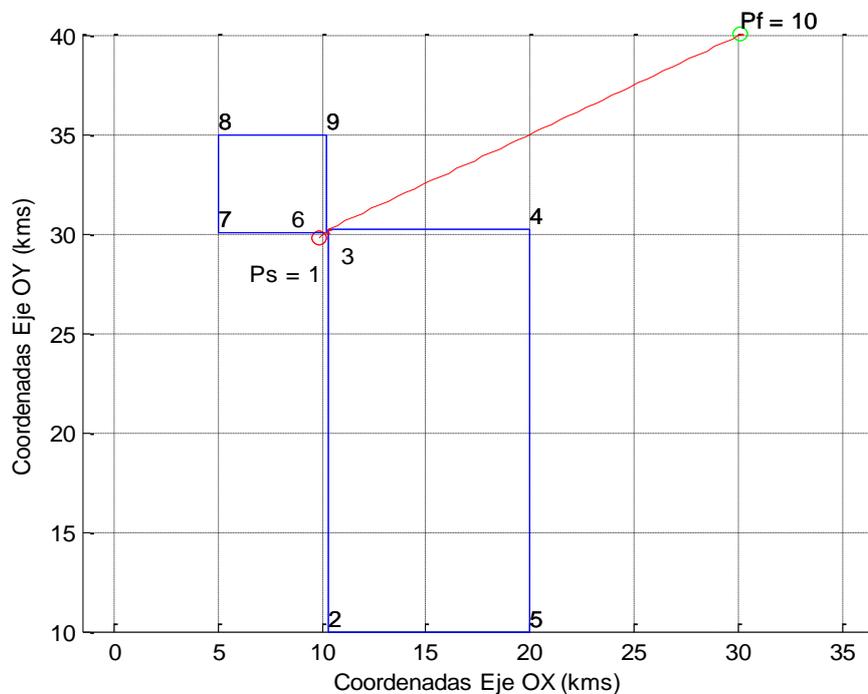


Figura 40 Ejemplo RNAV, 1.

En la Figura 40, se recoge una situación en la que al no ser la distancia entre los puntos de paso mayor que la distancia mínima de tramo, estos se han visto desplazados de su posición original en su respectivo vértice. En la Figura 41 se puede observar con mayor claridad lo comentado para los puntos de paso 1, 6 y 3. Se puede observar que para los puntos de paso 1-6 la distancia que los separa es mayor que la distancia mínima de tramo, y por lo tanto el punto de paso 6 no se ve desplazado. En cambio, para el caso del tramo comprendido entre los puntos de paso 6 y 3, la distancia es menor que la mínima, lo que implica que el punto de paso 3, en lugar de estar en el vértice, como correspondería debido al método de generación de grafos se encuentra desplazado.

Los puntos de paso calculados inicialmente mediante el algoritmo A* tenían la peculiaridad de que dos puntos de paso no consecutivos no eran visibles entre sí, de hecho, si lo fuesen, el punto o puntos situados entre ambos no tendrían sentido, pues provocaría que el camino no fuese el óptimo. Sin embargo, como se ha llevado a cabo una modificación de los puntos de paso para que estos estuviesen adaptados a la normativa de la OACI, puede ser que dos puntos de paso no consecutivos sí sean visibles entre sí. Es por ello que se debe llevar a cabo una segunda iteración la cual eliminará los puntos de paso que no sean necesarios. Para ello, el “software” calcula la visibilidad de los nodos con las nuevas coordenadas de los puntos de paso, y se comprueba la misma entre los

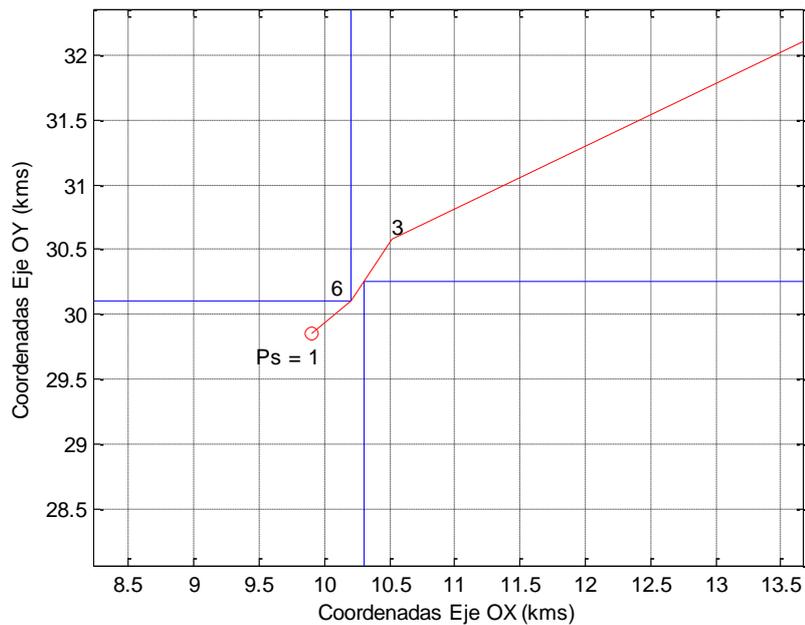


Figura 41 Ejemplo RNAV, 2.

puntos de paso solución del problema. Para evitar errores la iteración se lleva a cabo mediante el programa usado para la generación del “Grafo de Visibilidad”.

- **Ejemplo 2**

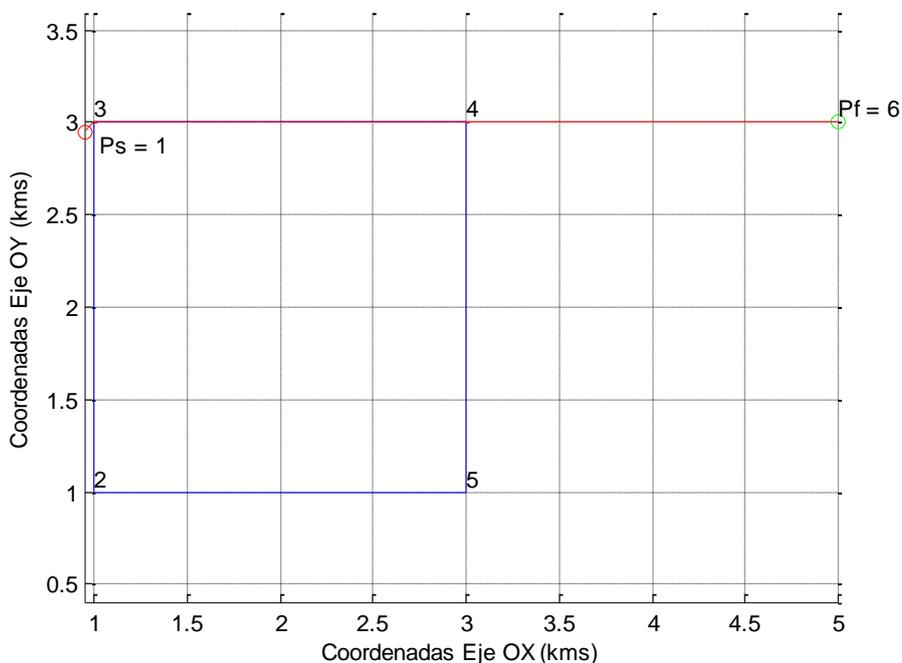


Figura 42 Ejemplo eliminación punto de paso innecesario, 1.

En la Figura 42 se muestra la solución sin tener en cuenta la distancia mínima de estabilización, pero se puede observar que la distancia entre los nodos 1 y 3, que son los dos primeros puntos de paso es muy pequeña, de hecho, inferior a la mínima permitida por la normativa. Se puede comprobar también en la figura mencionada que los puntos de paso solución de nuestro problema serán los nodos 1, 3, 4 y 6. Por el contrario, en la Figura 43 se recoge lo mencionado con anterioridad, que la distancia entre los nodos 1 y 3 es menor de la permitida, por lo tanto el programa ha buscado una posición alternativa del nodo 3, el cual al verse modificado, permite la

visibilidad entre el mismo y el punto de paso 6, lo cual permite una solución distinta a la inicial en la que se obvia el punto de paso 4.

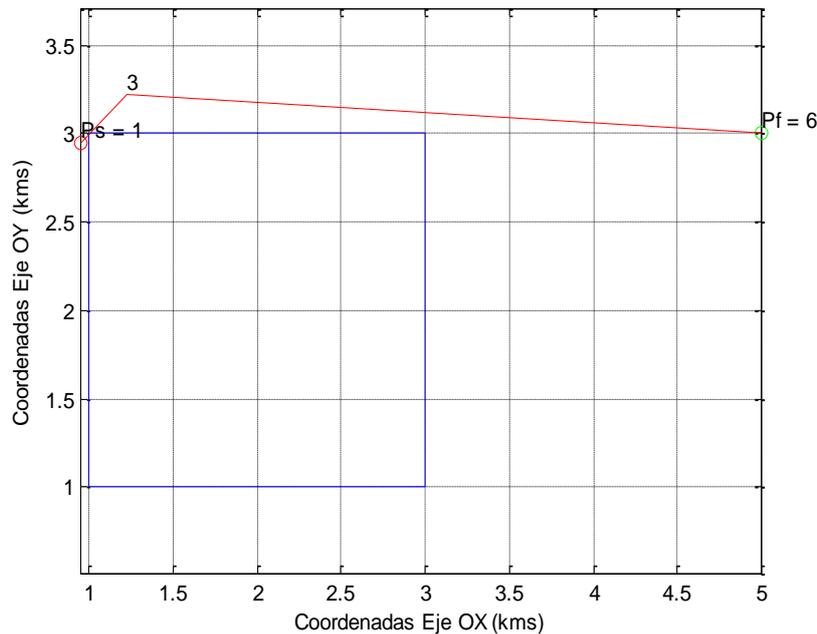


Figura 43 Ejemplo eliminación punto de paso innecesario, 2.

Al igual que en el resto de apartados, se ha llevado a cabo una batería de casos para determinar que realmente el programa funciona tal y como debe funcionar. Esta batería de casos viene recogida en el Apéndice D.

6.2. Protección Lateral de la Aeronave

El otro aspecto a tener en cuenta es la protección lateral de la que se debe de dotar a la aeronave. El método del “Grafo de Visibilidad” genera unas posibles trayectorias las cuales pasan por los vértices de los obstáculos y a veces incluso por las aristas de los mismos. Este hecho provoca que cualquier pequeña desviación de las condiciones supuestas conllevaría la colisión de la aeronave. Estas condiciones no deseadas pueden venir provocadas por los siguientes factores.

- Viento.
- Errores de precisión del GPS.
- Transitorio de la respuesta de los actuadores.

Por todo esto se ha estudiado la posibilidad de recrear los obstáculos de manera que los vértices y aristas que recorriese la aeronave no perteneciesen a los obstáculos sino al área de protección [2].

6.2.1. Normativa

La protección lateral de la aeronave viene recogida en la bibliografía con el nombre de semianchura del área (1/2 A/W) de franqueamiento de obstáculos en todas las aplicaciones RNAV se basa en lo siguiente:

$$\frac{1}{2} \cdot \frac{A}{W} = 1.5 \cdot XTT + BV$$

Dónde

- XTT: Tolerancia perpendicular a la derrota 2σ , siendo σ la desviación típica de la distribución normal.
- BV: Valor intermedio basado en las características de la aeronave y la fase de vuelo para tratar desviaciones más allá de valor 3 desviaciones estándar (3σ).

<i>Fase de Vuelo</i>	<i>BV para CAT A-E</i>	<i>BV para CAT H</i>
En ruta	3704 m	1852 m

Tabla 8 Valores intermedios.

En la Tabla 8 se recogen los valores intermedio que se les debe de proporcionar a las aeronaves en función de su categoría. Las Categorías A-E se refieren al caso de aeronaves de ala fija, mientras que la Categoría H describe el caso de que se esté trabajando con helicópteros. Por lo tanto, en nuestro proyecto se tomará los valores que se toman para las aeronaves de ala fija.

Conocido el valor intermedio, únicamente nos falta saber el valor de la tolerancia perpendicular a la derrota, XTT, para poder determinar la semianchura del área de franqueamiento de obstáculos. Para este caso habrá que diferenciar entre dos tipos de RNAV, esta diferenciación se recoge en la Tabla 9, en nuestro caso se ha considerado que se emplea RNAV de tipo 1-2, por lo que la semianchura del área de franqueamiento de los obstáculos es de 9.26 kilómetros.

<i>Fase de Vuelo</i>	<i>Especificación de navegación</i>	<i>XTT</i>	<i>1/2 A/W</i>
En ruta	RNAV 5	4650 m	10690 m
En ruta	RNAV 1 y 2	3704 m	9260 m

Tabla 9 Semianchura del área de franqueamiento de obstáculos.

Cabe destacar que usualmente se emplean dos tipos de área para la protección de las aeronaves. Una denominada primaria y otra conocida como secundaria. En nuestro caso solo se ha implementado la primaria, ya que se indica en la bibliografía que la segunda es opcional, su uso sería obligatorio en el caso de procedimientos de salida, aproximación y llegada.

Para poder simplificar el problema en lugar de proporcionarle la protección lateral a la trayectoria lo cual podría suponer que la solución dada por el programa no fuese válida (en primer lugar se calcularía la trayectoria y posteriormente se le aplicaría la protección, por lo que podría no existir espacio suficiente para la trayectoria con su correspondiente protección), se ha determinado que la protección se le añadirá a los obstáculos a modo de recrecido. De esta forma la solución proporcionada por el programa sería válida inicialmente y no habría que iterar.

En la literatura estudiada se ha encontrado que para llevar a cabo la implementación del recrecido de los obstáculos se usa principalmente el método conocido como "Minkowski Sum". Este método consiste en sumar

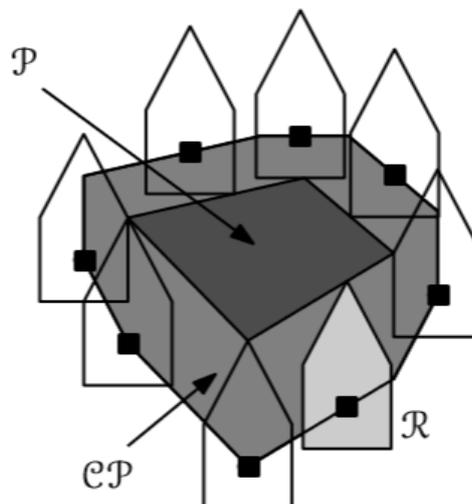


Figura 44 Ejemplo "Minkowski Sum".

a lo largo de todo el perímetro de cada obstáculo una protección.

En la Figura 44 se puede observar con una mayor claridad el método. En primer lugar, se pueden diferenciar claramente dos polígonos, el cuadrilátero central (obstáculo), el cual es el área a dotar de protección lateral, y un pentágono (protección), que en este caso es el polígono que se va a sumar al cuadrilátero a fin de proporcionarle la protección mencionada anteriormente. Se observa cómo la protección en este caso no es igual en todas las zonas del cuadrilátero, sino que depende de la zona del pentágono que se le haya sumado.

Se ha decidido dividir el problema en tres partes:

- **Protección Lateral de las Aristas:**

En nuestro caso, debido al tipo de problema que se pretende resolver, la protección que mejor se adaptaría a nuestras características sería una circunferencia, pero su implementación conlleva dos problemas asociados:

- El primero de ellos es que nunca se obtendría una circunferencia como tal, sino que sería un polígono regular con n lados, siendo n un valor muy elevado.
- El segundo de los problemas surge del elevado número de lados que conllevaría el uso de circunferencias, pues al existir muchos vértices, estos se van a encontrar muy cercanos entre sí, por lo que habrá que tenerlos en cuenta a la hora del cálculo de la distancia mínima de estabilización, siendo su utilidad nula, pues se verían desplazados de su ubicación inicial.

Otra opción que se ha barajado ha sido la utilización para la protección de un cuadrado, pero en este caso, en el momento en que los obstáculos no sean perpendiculares o paralelos a los ejes de coordenadas designados, se tendría un recrecido mayor que el determinado por la normativa.

Es por ello que la solución adoptada para este trabajo a la hora de proporcionarle un recrecido al obstáculo se basa en calcular la recta paralela a la arista del obstáculo tal que la distancia entre ambas sea la semianchura proporcionada por la normativa. Esta solución conlleva que la zona próxima a los vértices de los obstáculos tenga una protección mayor de la requerida inicialmente, este hecho será interesante cuando se estudie la protección exterior e interior, pues nos permitirá no tener que aplicarlas al existir dicho exceso.

Una vez recrecidos los obstáculos, estos variarán su tamaño, por lo que donde antes existían varios obstáculos, ahora puede existir un único obstáculo. Esto provoca que una vez recrecidos los mismos, se deba de verificar si los obstáculos se intersectan o no, y en caso de que sí ocurriese convertirlo en un único obstáculo. Para ello se ha usado el programa "PolygonClip". El cual nos permite la unión de dos o más polígonos en caso de que sus aristas se intersecten. Para este código no se ha llevado a cabo una batería de casos como se ha hecho con el resto, sino que se presupone que su funcionamiento es el correcto.

Con esto se tendría la protección lateral de los obstáculos en caso de tramos rectos, pero también se deberá de tener en cuenta la protección en el caso de virajes. Para este tipo de protección se ha estudiado la Parte III-Sección 2, Capítulo 2 en ella se indica que hay que diferenciar dos tipos de protección en caso de viraje, la interior y la exterior.

- **Protección Exterior:**

Con respecto a la protección exterior, se indica en la Parte III-Sección 3, Capítulo 8 que para procedimientos en ruta se debe de usar las espirales de viento o los círculos limitadores, siendo esta última una versión simplificada de la primera, y por lo tanto la que se pretende estudiar.

La construcción de los círculos limitadores se recoge en la Parte I-Sección 2, Capítulo 2. En primer lugar se van a definir una serie de parámetros y factores empleados para el cálculo de los mismos.

- Ángulo de inclinación lateral, α , en ruta viene dado siendo su valor 15° .
- Régimen de viraje, R , en grados/segundos. Siendo el máximo $3^\circ/\text{segundo}$.
- La velocidad, V se obtiene de la fórmula inferior.

$$R = 6355 \cdot \tan \alpha / (V \cdot \pi)$$

- Radio de viraje, r , a un ángulo de inclinación lateral designado con el aire en calma, en km.

$$r = \frac{V}{20 \cdot R \cdot \pi}$$

- El viento empleado es el viento normalizado OACI, w , en km/h, donde h es la altitud dada en km.

$$w = 12 \cdot h + 87$$

- Efecto del viento, E , en km, durante el tiempo empleado en cambiar el rumbo, θ .

$$E = \frac{\theta}{R} \cdot \frac{w}{3600}$$

En la Figura 45 se puede observar un ejemplo de la construcción del límite exterior para el procedimiento de los círculos limitadores. En ella se observa que los parámetros necesarios para su construcción son la semianchura del área de franqueamiento de los obstáculos, $\frac{1}{2} A/W$, el efecto del viento, E , y el radio de viraje, r .

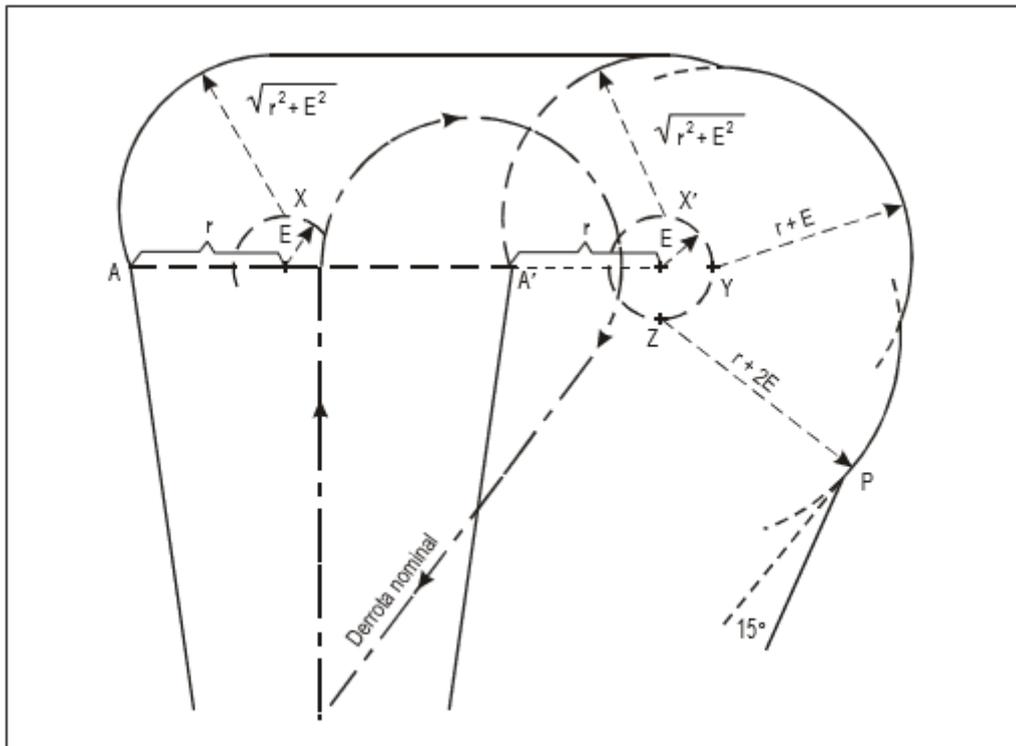


Figura 45 Círculos limitadores. Límite exterior.

Esta protección sería adicional a la de los tramos rectos, explicada en este mismo apartado. Como se ha comentado anteriormente, el procedimiento que se ha empleado anteriormente proporcionaba una protección excesiva en la zona próxima a los vértices de los obstáculos.

A continuación se van a explicar dos conceptos, la protección de exterior de viraje necesaria y disponible. El objeto de este análisis es comprobar si realmente es necesario aplicar los círculos limitadores o realmente con el exceso de protección existente en la zona aledaña a los vértices sería suficiente para evitar la colisión de la aeronave.

Sea la *Protección Exterior de Viraje Necesaria* la máxima en caso de que el viraje fuese en las condiciones más desfavorables posibles. En la bibliografía [2], se puede observar que la protección necesaria para proteger los vértices nunca será superior a:

$$\text{Protección Exterior de Viraje Necesaria} = r + 3 \cdot E$$

$$\text{Protección Exterior de Viraje Necesaria} = r + \frac{3 \cdot \theta}{R} \cdot \frac{w}{3600} = r + \frac{3 \cdot \theta}{V} \cdot \frac{20 \cdot w \cdot \pi \cdot r}{3600}$$

A continuación se va a explicar una simplificación a partir de la cual se nos va a permitir no tener que llevar a cabo la construcción de los límites exteriores, sino que podemos suponer que la aeronave estará suficientemente protegida con la protección lateral inicial.

Sea la *Protección Exterior de Viraje Disponible* la semianchura del área de franqueamiento de obstáculos, es decir el valor que se le va a proporcionar a los obstáculos en las aristas. Que era el RNAV tipo 1-2.

$$\text{Protección Exterior de Viraje Disponible} = \frac{1}{2} \cdot \frac{A}{W}$$

$$\text{Diferencia de Protección Exterior} = \frac{1}{2} \cdot \frac{A}{W} - \left(\frac{V}{20 \cdot \pi \cdot R} + \frac{3 \cdot \theta}{R} \cdot \frac{w}{3600} \right)$$

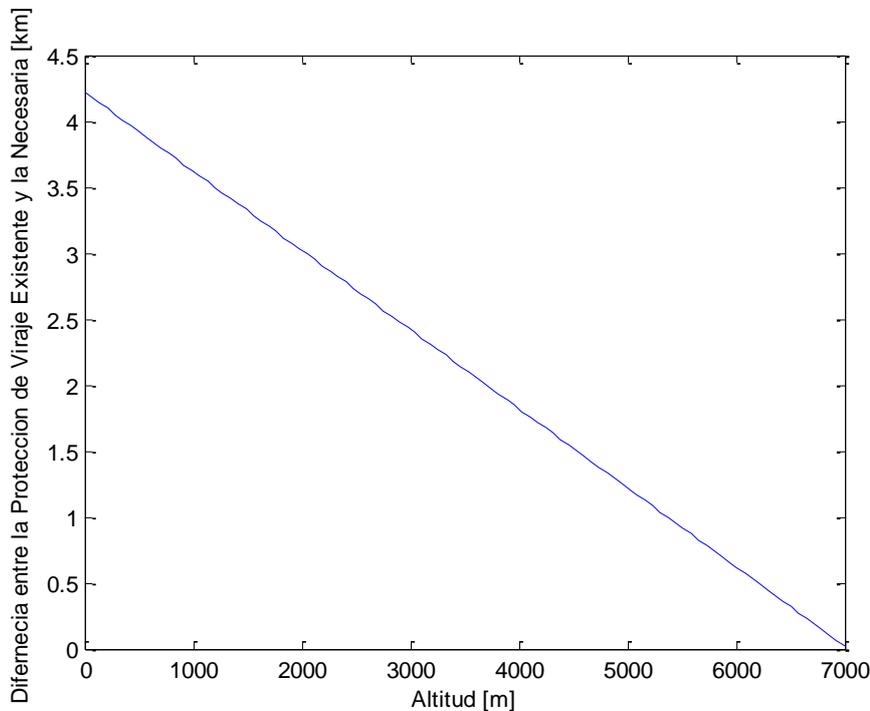


Figura 46 Diferencia entre la protección lateral disponible y necesaria, límite interior.

Siendo los valores de los parámetros,

- $\frac{1}{2} A/W$, en el caso más desfavorable, 9.26 km.
- $V=130$ km/h.
- $R=3^\circ/s$.
- $r=0.68$ km.
- $\theta=180^\circ$, pues es el máximo valor del cambio de rumbo.

La idea es obtener la diferencia entre la protección lateral disponible (semianchura del área de franqueamiento) y la protección lateral necesaria para realizar el viraje cumpliendo la normativa. Si esta diferencia es positiva, la protección dada por la protección lateral en las aristas es necesaria, si por el contrario dicha diferencia fuese negativa, entonces no se podría llevar a cabo la simplificación y sería necesario aplicar los círculos limitadores.

El valor de viento normalizado OACI, w , depende de la altitud de vuelo, h , la cual estará comprendida en nuestro estudio entre 0 y 7000 m, en la Figura 46 se puede observar que hasta una altitud de 7 kilómetros no sería necesario aplicar una protección adicional a la trayectoria, para altitudes mayores, debido a efectos del viento, se debería aplicar una protección adicional. En cuanto al cambio de rumbo, se ha considerado el más desfavorable posible, es decir, 120° . En nuestro caso se ha decidido que las trayectorias siempre se encontrarán por debajo de dicha altitud.

- **Protección Interior:**

Para el caso de la protección interior, se indica que debe de ser tal que una el área primaria anterior y posterior¹¹ al punto de recorrido, es decir, una recta. En la Figura 47 se observa los puntos de viraje anterior y posterior. En primer lugar se debe determinar el cálculo de dichos puntos. Dado el punto de paso i:

¹¹ Área primaria anterior y posterior: Se refiere al último punto y al primero de la semianchura antes de comenzar el giro respectivamente.

- El punto de viraje anterior se halla en la recta que une los puntos de paso i-1 e i a una distancia igual a $r \cdot \tan(\theta/2)$, siendo r el radio del viraje y θ el cambio de rumbo, $Pto_Ant=r \cdot \tan(\theta/2)$.
- El punto de viraje posterior por su parte está ubicado sobre la recta que une los puntos i e i+1 situado a una distancia igual a la menor de una de estas, $Pto_Post=\min [r, r \cdot \tan(\theta/2)]$.

Para calcular la protección interior necesaria se tiene que la distancia que separa los puntos anterior y posterior de viraje es:

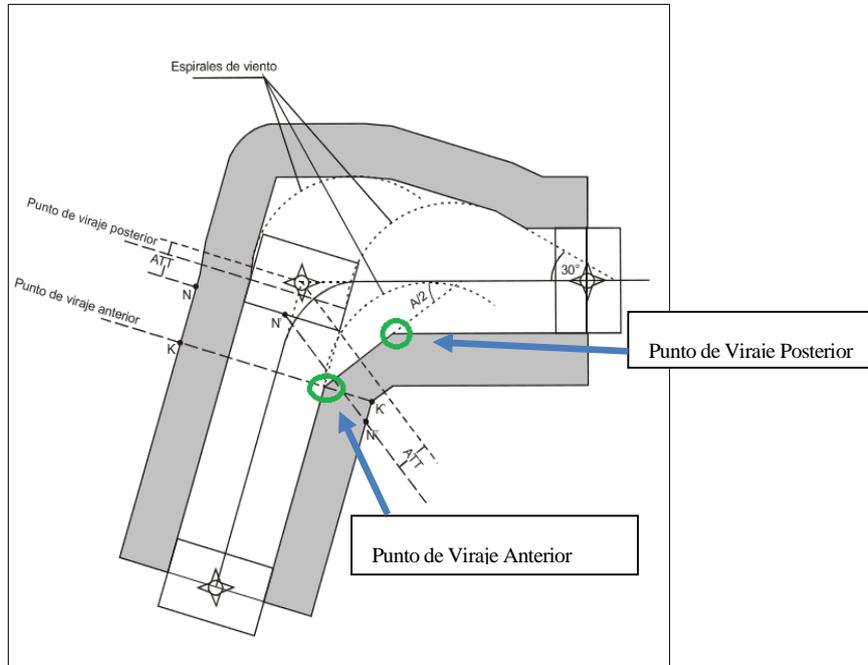


Figura 47 Puntos de viraje anterior y posterior.

$$rect = \sqrt{Pto_Ant^2 + Pto_Post^2 - 2 \cdot Pto_Ant \cdot Pto_Post \cdot \cos(\pi - \theta)}$$

Los que nos lleva a que la distancia mínima necesaria para proteger el viraje es:

$$Dist_Necesaria = Pto_Post \cdot \tan\left(\frac{\text{asin}(Pto_ant \cdot \sin(\pi - \theta))}{rect}\right)$$

Siendo el numerador de la última ecuación el ángulo formado por las rectas que unen el punto de viraje posterior con el punto de viraje anterior y el punto de paso.

Usando la formulación presentada para el caso de la protección exterior, se ha tomado un régimen de viraje, $R=3^\circ/s$, siendo para dicho valor el radio de giro, $r=0.68$ km. Para determinar si es necesaria una sobreprotección para la protección interior se va a llevar a cabo un estudio de la distancia de protección lateral disponible y la necesaria¹² en función del cambio de rumbo. Si es suficiente significa que la recta que une los puntos de viraje anterior y posterior puede trazarse sin intersectar al obstáculo, y por lo tanto nuestra protección sería suficiente para llevar a cabo el recrecido evitando la colisión y cumpliendo la normativa.

En este caso, al igual que para el límite exterior, se pretende ver que la diferencia entre la protección lateral disponible y necesaria es positiva, para ello se ha hecho uso de relaciones trigonométricas elementales. En la Figura 48 se recoge la diferencia entre la protección interior disponible y necesaria, este valor crece, pues al crecer el cambio de rumbo, la distancia entre el punto de paso y el vértice del obstáculo crece, nuestro problema está acotado para cambios de rumbo menores de 120° .

Esta protección necesaria es menor que la dada a los obstáculos con el recrecido comentado anteriormente por lo tanto su aplicación no será necesaria.

¹² La protección lateral disponible y necesaria tiene el mismo significado que en el caso de protección exterior.

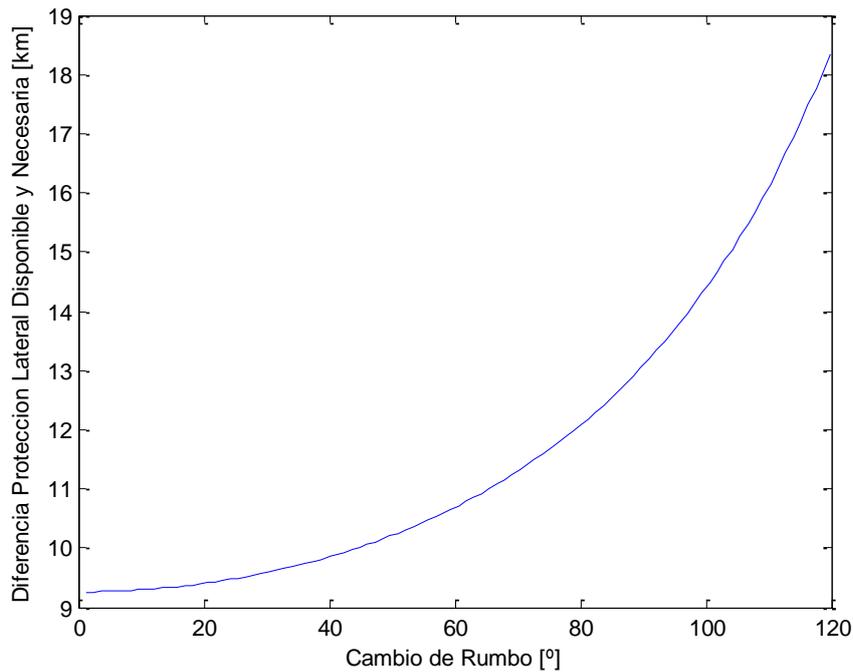


Figura 48 Diferencia entre la protección lateral disponible y necesaria para el viraje, límite exterior.

6.2.2. Algoritmo de Protección Lateral de la Aeronave

A continuación se va a presentar un breve esquema en el cual se recoge el algoritmo que se ha implementado para poder conseguir el recrecido.

- Se toma el obstáculo i , y para cada arista, j , del mismo se calcula una recta paralela, $j1$, y separada la distancia correspondiente en función de la CAT y tipo de RNAV,
- Una vez calculadas todas las rectas paralelas, para cada una de ellas, $j1$, se calcula su intersección con la siguiente, $j1+1$, obteniendo de este modo el nuevo vértice.

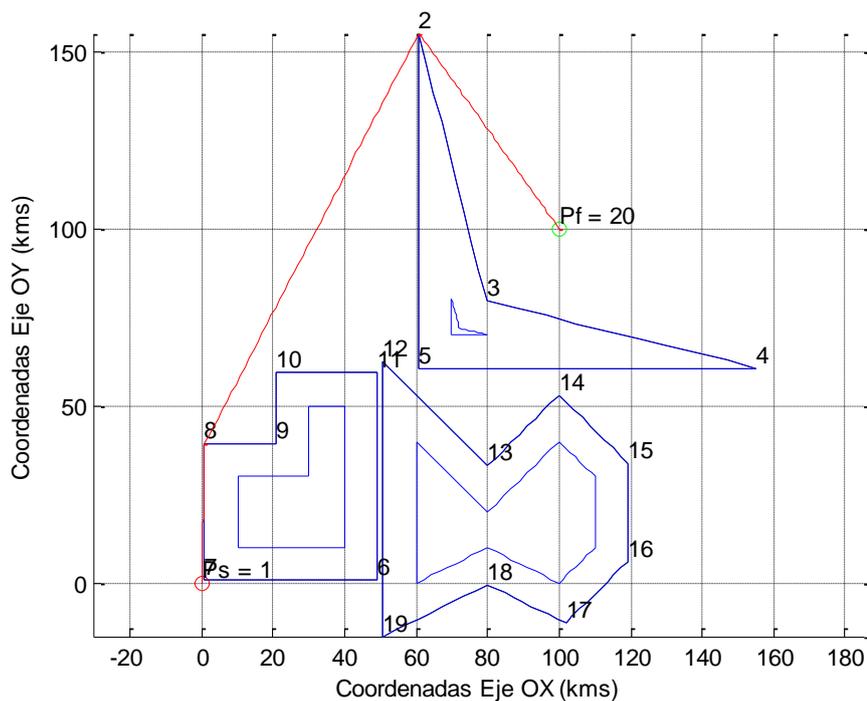


Figura 49 Ejemplo de recrecido sin intersección de los obstáculos.

- Se lleva a cabo este proceso para todos los vértices de todos los obstáculos.
- Si existe intersección entre dos o más obstáculos la función “PolygonClip” se encarga de convertirlos en un único obstáculo.

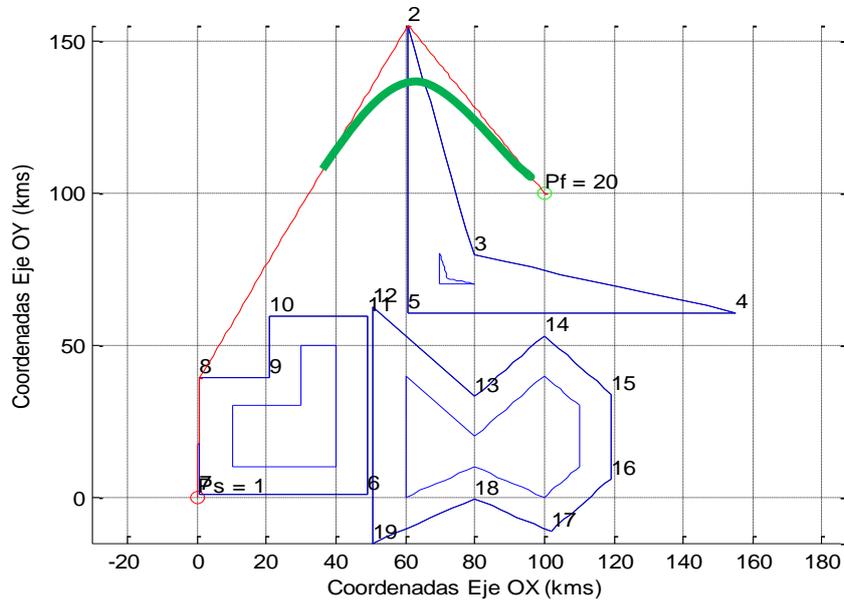


Figura 50 Punto de recorrido de paso. Ejemplo de trayectoria.

En la Figura 49 se muestra un ejemplo de cómo se implementaría el recrecido a los obstáculos, de modo que se tuviese una protección lateral que evitase la colisión en caso de que la aeronave se desviase de la trayectoria generada por el programa. Para este ejemplo se ha supuesto CAT A-E y un RNAV de tipo 1 o 2, siendo la semianchura de valor $\frac{1}{2} A/W = 10690$ m. Se muestran tanto los obstáculos iniciales como los mismos contemplando el recrecido, para una mejor visualización de cómo funciona este apartado y de su fin. En el código desarrollado, si el cambio de rumbo en un punto de paso fuese mayor que 120° , lo descartaría como solución posible y pasaría a una de las alternativas.

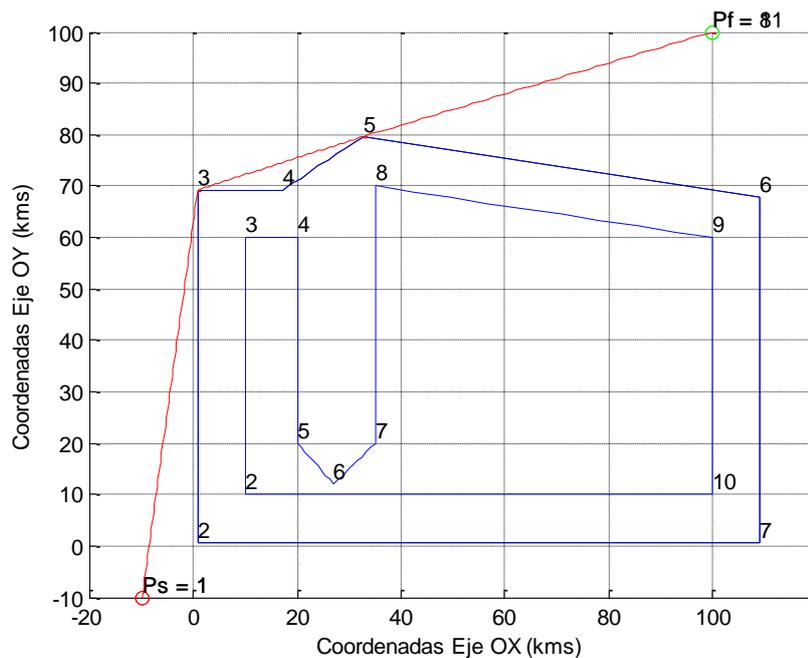


Figura 51 Ejemplo de obstáculo con tres vértices cóncavos y rectas paralelas entre sí.

Es necesario resaltar que en el caso de un cambio de rumbo elevado, como por ejemplo en el vértice 2 de la Figura 49, el recrecido provocaría que dicho vértice estuviese muy alejado de su respectivo vértice inicial, lo

que provocaría una sobreprotección elevada y por consiguiente una mayor distancia a recorrer, sin embargo, es necesario recordar que nuestros puntos de recorrido son de paso siempre, nunca de sobrevuelo. En la Figura 50 se puede observar cómo sería la trayectoria a realizar por la aeronave.

Un caso particular a la hora del cálculo del recrecido es cuando existen dos o más vértices cóncavos consecutivos. Si estos están separados una distancia mayor que el doble de la semianchura, $\frac{1}{2} A/W$, no existe ningún tipo de singularidad, pero si por el contrario, esta distancia es menor, pueden existir problemas a la hora de generar el recrecido. Para evitar este tipo de problemas, se ha optado por diferenciar dos casos. El primero de ellos es que las rectas adyacentes a los vértices cóncavos sean paralelas entre sí, en ese caso el vértice anterior al primero de los cóncavos y siguiente al último de los mismos se unen, eliminando de esta forma los vértices situados entre ambos, es decir, los cóncavos. En la Figura 51 se puede observar con mayor claridad el ejemplo comentado. Se puede comprobar que la numeración de los vértices difiere, si bien este hecho se debe a que tenemos dos

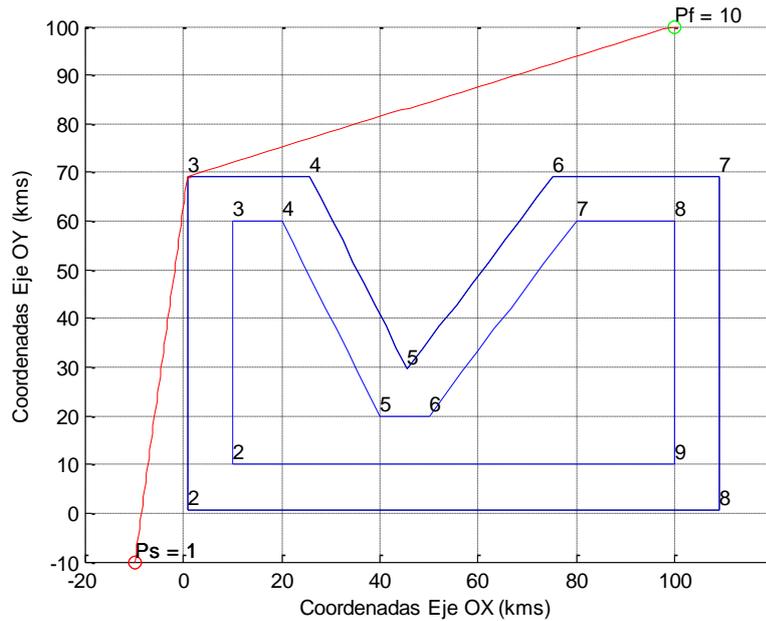


Figura 52 Ejemplo de obstáculo con dos vertices cóncavos y rectas no paralelas entre sí.

obstáculos con distinto número de vértices, mientras que el primero de ellos tenía nueve vértices, el obstáculo recrecido únicamente posee seis.

El segundo de los casos es aquel en el que la recta anterior al primer vértices cóncavo y la posterior al último de

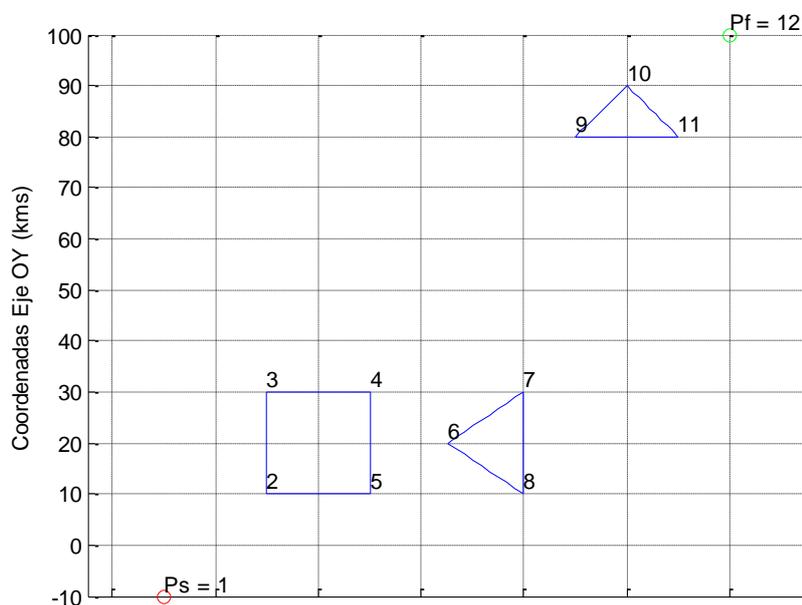


Figura 53 Ejemplo de intersección de obstáculos, 1.

ellos no son paralelas entre sí, en este caso se calcula la intersección entre ambas, sustituyendo todos los vértices cóncavos por este último obtenido de la mencionada intersección. Un ejemplo de este tipo de problema se puede observar en la Figura 52.

En la Figura 53 se puede observar el problema inicial, existen tres obstáculos bien diferenciados los cuales deben de ser evitados al generar la ruta.

En cambio, en la Figura 54 se puede observar que al recrecer los obstáculos, dos de ellos, interseccionan, por lo que en este caso será necesario el uso del programa mencionado anteriormente, "PolygonClip". El cual nos va a permitir que los dos obstáculos que se han cortado en dos puntos convertirlos en un único obstáculo.

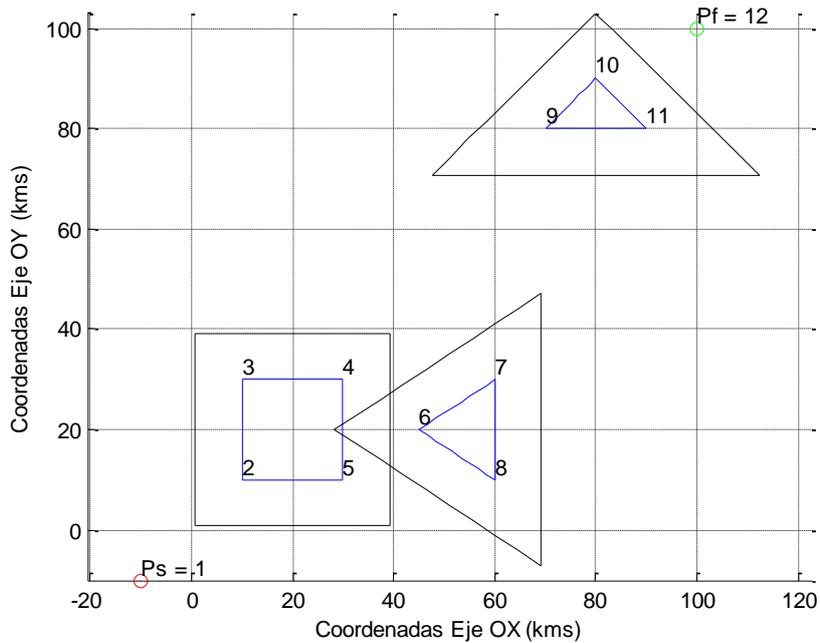


Figura 54 Ejemplo de intersección de obstáculos, 2.

Por último en la Figura 55 se puede observar cómo se quedarían los obstáculos una vez recrecidos y calculada la unión de ellos en caso de que interseccionen, así como la solución propuesta por el programa. La ruta calculada por el programa nos indica que se debe de pasar por los nodos, 1, 9, 4 y 13.

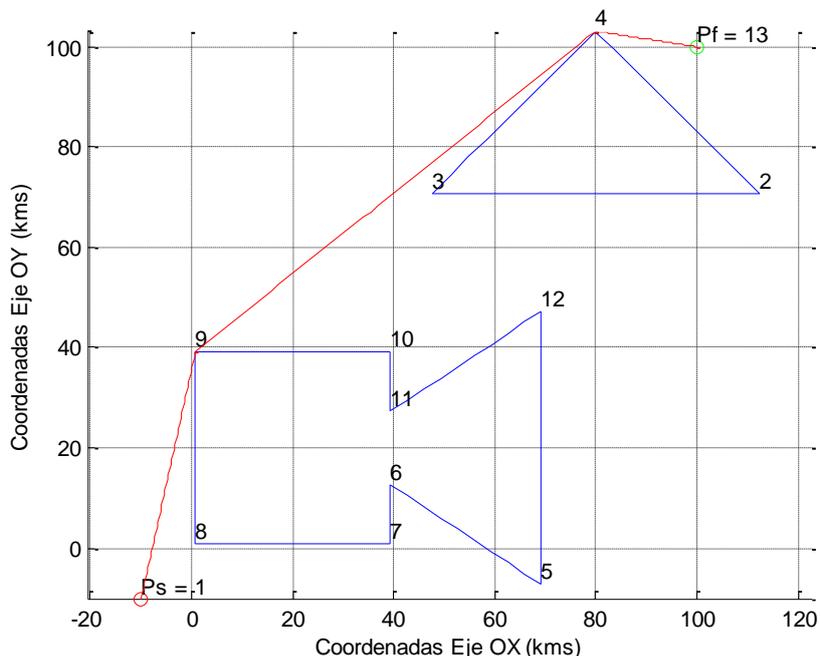


Figura 55 Ejemplo de intersección de obstáculos, 3.

Con este apartado se dan por concluidos los apartados en los que se explican las decisiones tomadas para la creación del código y pasamos a continuación a las aplicaciones y conclusiones extraídas de esta memoria.

7 APLICACIONES

En el apartado de Aplicaciones se van a comentar dos posibles usos que se le pueden dar al programa que se ha desarrollado y que se ha explicado a lo largo de la presente memoria. En primer lugar se presenta cómo esquivar una tormenta y en el segundo se va a mostrar un ejemplo en el cuál la aeronave debe recorrer un valle.

7.1. Esquivar una tormenta

En este ejemplo como se ha indicado anteriormente se va a mostrar cómo actuaría el programa en caso de que la aeronave tuviese que esquivar una tormenta para lograr llegar desde el punto inicial al final. En la Figura 56 se observa las distintas nubes que debe evitar la aeronave, se ha decidido no numerar los vértices ya que al existir muchos nodos, no se aprecia realmente la etiqueta.

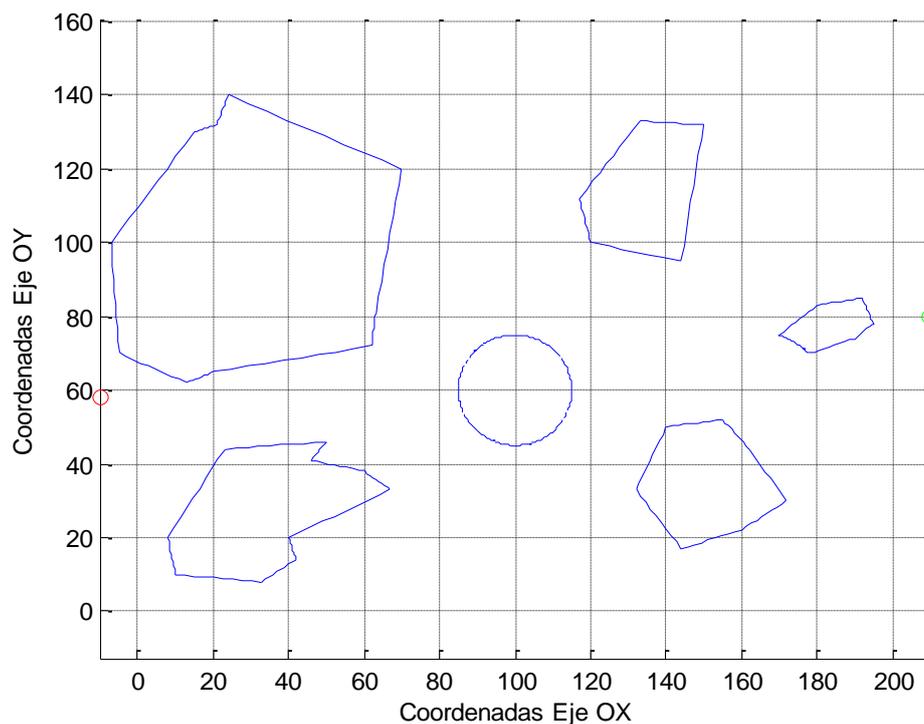


Figura 56 Ejemplo de aplicación para evitar una tormenta, 1.

Una vez que se tiene el problema inicial, se recoge en la Figura 57 cómo son los obstáculos una vez que se le ha añadido la protección lateral. A continuación en la Figura 58 se recoge la solución propuesta por el programa con el obstáculo recrecido. Por su parte, en la Figura 59 se recoge la solución con los obstáculos sin recrecer.

Para ello se deben recorrer los nodos 1, 3, 32, 80, 81, 82, 83, 84, 85, 86, 87, 103, 116, 117, y 120 cuyas coordenadas vienen recogidas en la Tabla 10.

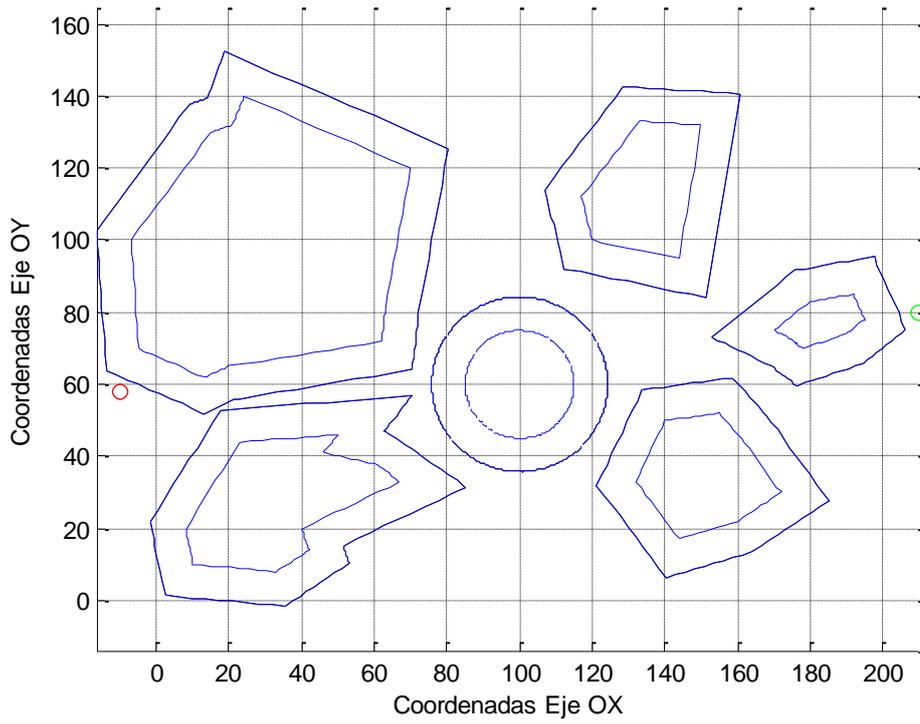


Figura 57 Ejemplo de aplicación para evitar una tormenta, 2.

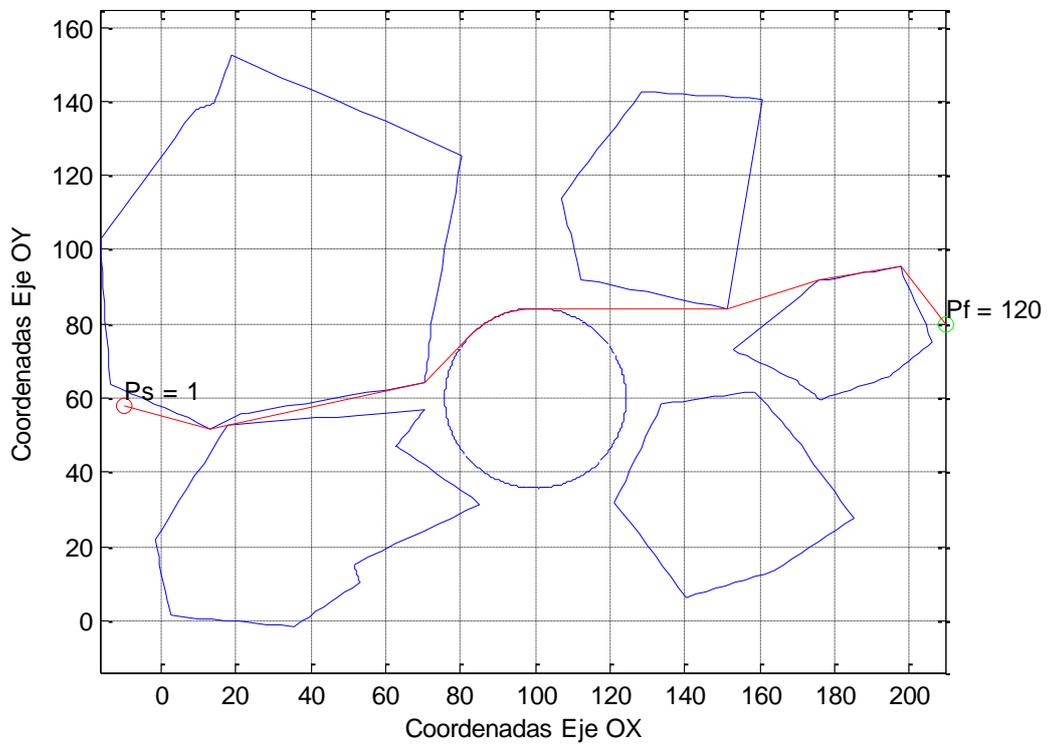


Figura 58 Ejemplo de aplicación para evitar una tormenta, 3.

7.2. Pasar por un valle

Esta aplicación ha sido uno de los objetivos desde la concepción del proyecto. Se trata de poder generar una trayectoria la cual permita a la aeronave pasar por un valle sin colisionar con las paredes que lo forman. En este caso es muy importante haber generado un área de protección lateral la cual permita que los errores de precisión no provoquen la colisión, pues el área para la maniobrabilidad es muy pequeña.

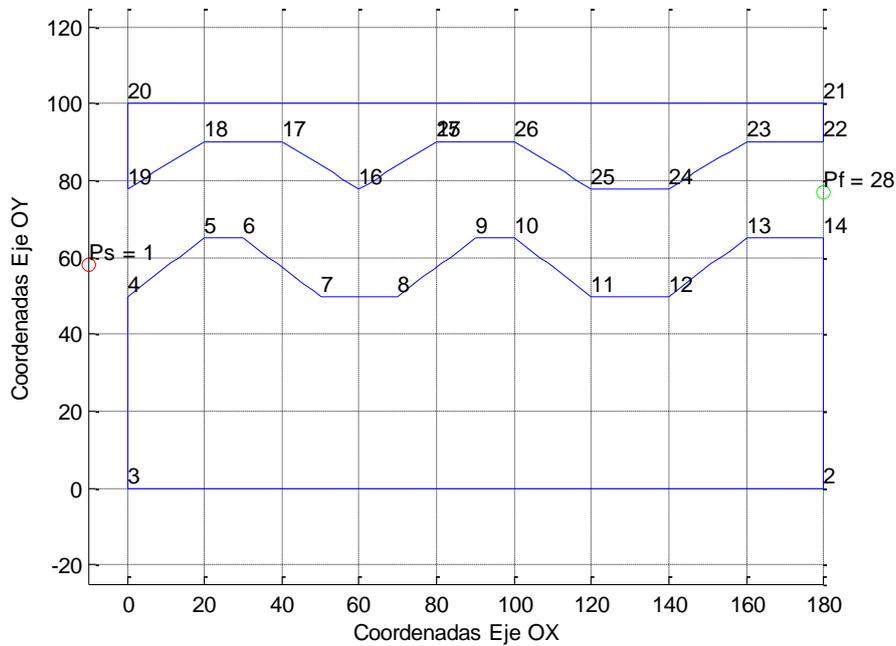


Figura 60 Ejemplo para recorrer un valle, 1.

En la Figura 60 se puede observar el problema inicial, en el que se ha representado el valle con los dos polígonos que aparecen en la misma. En cambio, en la Figura 61, se puede ver cómo quedaría el problema una vez que se le ha aplicado el recorcido a los obstáculos.

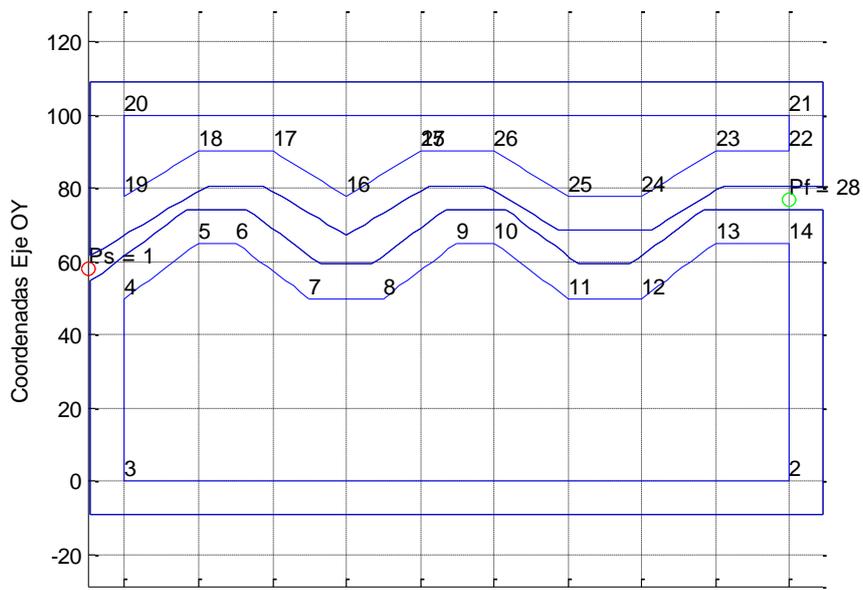


Figura 61 Ejemplo para recorrer un valle, 2.

En la Figura 62, no se aprecia bien, debido a que la descomposición mediante el “Ear Clipping” ha dado lugar a muchos triángulos, pero en ella se recoge cómo ha llevado a cabo el programa la descomposición para eliminar los obstáculos cóncavos.

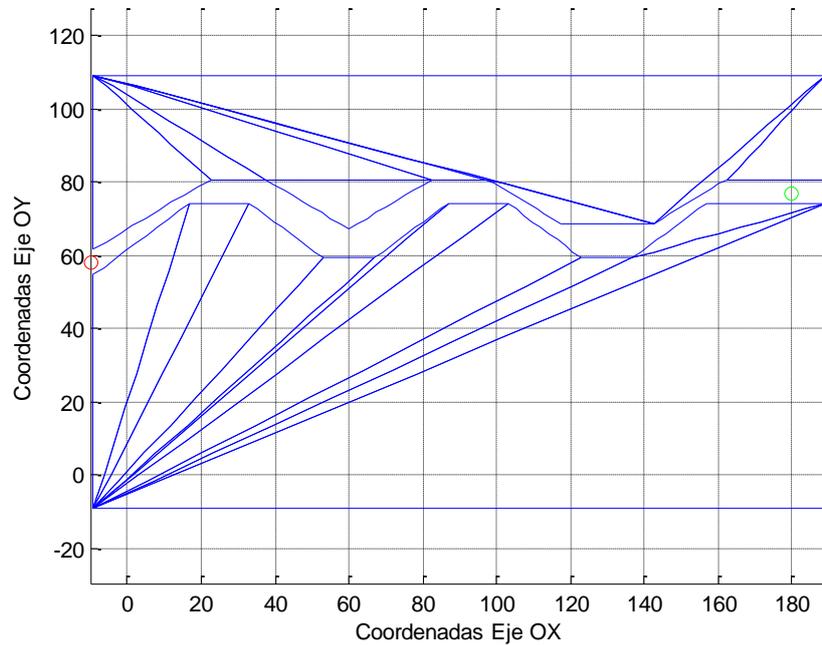


Figura 62 Ejemplo para recorrer un valle, 3.

Para finalizar se representa la solución propuesta por el programa en la Figura 63 con el obstáculo sin recrecido y en la Figura 64 con el recrecido del mismo. La ruta que hemos obtenido recorre los vértices 1, 9, 12, 39, 21, 24, 35, 64, 33 y 65, viniendo las coordenadas de estos vértices en la Tabla 11.

Trayectoria	
Coordenada X	Coordenada Y
-10,0000	58,0000
16,9153	74,2540
33,0847	74,2540
60,0000	67,2081
86,9153	74,2540
103,0847	74,2540
117,4368	68,7460
142,5632	68,7460
156,9153	74,2540
180,0000	77,0000

Tabla 11 Coordenadas de los puntos de paso solución de nuestro problema, 2.

El tiempo que se ha empleado en resolver este problema ha sido de 102.631 segundos, de los cuales 100.275 segundos han sido del subprograma VerObs22, el cual determina la visibilidad de cada nodo.

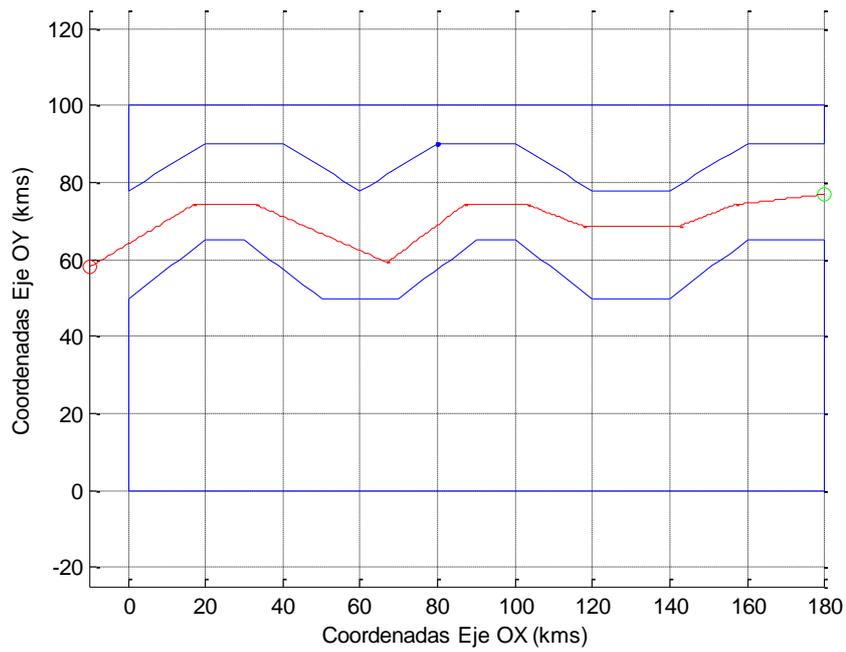


Figura 63 Ejemplo para recorrer un valle, 4.

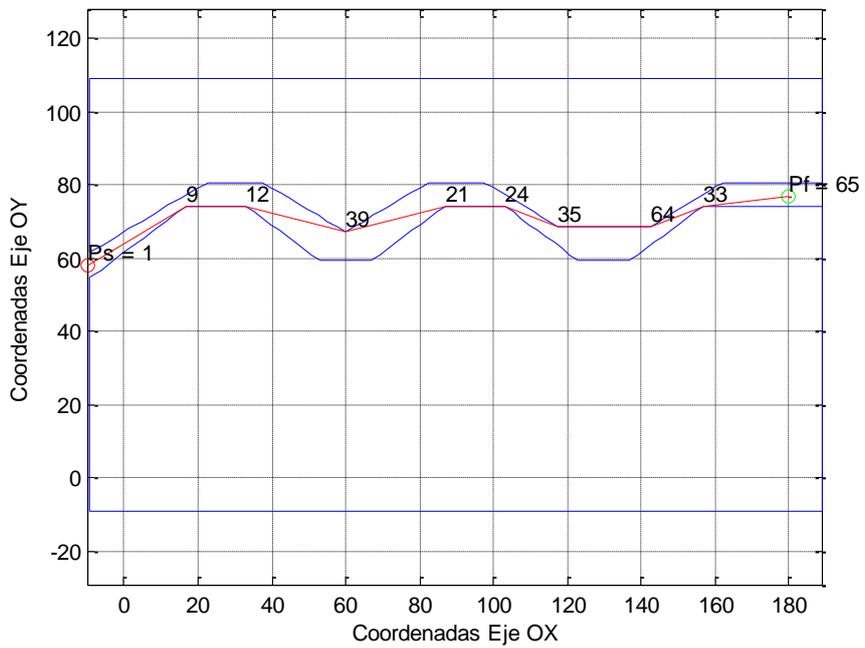


Figura 64 Ejemplo para recorrer un valle, 5.

8 CONCLUSIONES

En este trabajo se ha desarrollado un programa el cual dados un punto inicial y otro final es capaz de calcular la trayectoria óptima evitando todos los obstáculos existentes entre ambos puntos y manteniendo una distancia de seguridad.

Para abordar el problema la estrategia seguida ha consistido en dividirlo en tres partes.

- En primer lugar mediante el grafo de visibilidad se determina que nodos eran visibles desde cada uno de ellos.
- Posteriormente se aplica un algoritmo de búsqueda en grafos, en nuestro caso el A*, el cual dado el grafo nos proporciona la ruta óptima.
- Por último, se consideran las restricciones que vienen impuestas en el documento 8168 para la construcción de trayectorias volables de acuerdo a procedimientos GNSS-RNAV.

En el capítulo anterior se han mostrado dos ejemplos de situaciones en las que podría llevarse a cabo el uso de este programa. Estas situaciones han sido esquivar una tormenta y recorrer un valle y se ha podido comprobar que los resultados son óptimos, en cuanto al tiempo requerido por *Matlab* para resolver el problema son del orden de 100 segundos, por lo que podría solventarse un imprevisto, como una tormenta, en caso de necesidad. El tiempo aunque algo elevado es prometedor, en sucesivos proyectos sería adecuado bajarlos.

En cuanto al funcionamiento del programa, existen cinco anexos en los cuales se recoge una batería de casos la cual nos permite comprobar que realmente el resultado de cada caso es el esperado. Para generar esta batería de casos se ha ido examinando el código del programa línea a línea y se ha planteado un caso para comprobar cada una de las que pudiesen conllevar errores.

Siguiendo la línea de este proyecto existen una serie de situaciones que no se han abordado y que podrían dar lugar a futuros trabajos. En todo momento hemos considerado que la altitud de vuelo es constante, por lo tanto un posible estudio podría tener en cuenta el caso en el que la altitud de vuelo fuese variable. Para este caso, una posible solución pasaría por calcular dos planos, el primero de ellos vertical y que contuviese a la recta que une los puntos inicial y final y otro perpendicular a éste y que también contuviese a la recta indicada. Para finalmente calcular los caminos más cortos en dichos planos y elegir el más óptimo.

Otro posible caso a estudiar sería el caso de considerar que la tormenta no fuese un obstáculo estático, sino tener en cuenta que esta varía su posición con el tiempo.

Solo se ha considerado los puntos de recorrido de paso. En futuros trabajos se podría también tener en cuenta el caso de puntos de recorrido de sobrevuelo. Para ello las tablas recogidas en el documento 8168 serían diferentes de las empleadas para este proyecto.

Cabe recordar también que toda la normativa del documento 8168 considera únicamente aviones tripulados, por lo que las tablas que se han tomado para el cálculo de la mínima distancia de estabilización no es la adecuada para el caso de UAV, así como la protección lateral de la aeronave, la cual puede que sea muy elevada para los vehículos aéreos no tripulados. Por lo que un proyecto que podría realizarse en fechas venideras sería considerando el caso de UAV y por lo tanto haciendo uso de tablas que contuviesen valores adecuados para este tipo de vehículos.

Por último, una posible línea de trabajo sería considerar el caso de que la velocidad de la aeronave no fuese constante, sino que se adaptase a las condiciones de la zona de vuelo. Para ello podría quizás adecuarse la velocidad teniendo en cuenta la distancia de los objetos, y también influiría que la distancia mínima de tramo RNAV variará pues esta depende de la velocidad. Por lo que la aeronave se podría adaptar a la que más le interesase en cada caso.

REFERENCIAS

- [1] De Berg, Mark, «Computational Geometry, » Springer, Third Edition, 2008.
- [2] OACI, «Aircraft Operations, » vol. 2, Fifth Edition, 2006.
- [3] Tournois, Jane, «2D Centroidal Voronoi Tessellations with Constraints, » INRIA Sophia Antipolis, p. 21-222, 2010.
- [4] Dijkstra, E. W., «Numerische Mathematik, » vol. 1 p. 269-271, 1959.
- [5] Mojtahedzadeh, Rasoul, «Robot Obstacle Avoidance using the Kinect, » Royal Institute of Technology School of Computer Science and Communication, 2011.
- [6] Dolgov, Dmitri, «Practical Search Techniques in Path Planning for Autonomous Driving, » Stanford University.
- [7] Ghosh, Subir Kumar, «Visibility-based Robot Path Planning, » School of Technology & Computer Science Tata Institute of Fundamental Research, Mumbai.
- [8] Hoff III, Kenneth E., «Fast Computation of Generalized Voronoi Diagrams Using Graphics Hardware, » University of North Carolina at Chapel Hill, Department of Computer Science.
- [9] Koenig, Sven, «Fast Replanning for Navigation in Unknown Terrain, » IEEE Transactions on Robotics and Automation, vol. XX. nº Y, 2008.
- [10] Tsourdos, Antonios, «Cooperative Path Planning of Unmanned Aerial Vehicles, » Wiley, First Edition, 2011.
- [11] LaValle, Stephen, «Path Planning, » Cambridge, 2006.

APÉNDICE A. BATERÍA DE CASOS PARA UN OBSTÁCULO

Una vez analizado el código se ha llegado a la conclusión de que los casos a estudiar para llevar a cabo la comprobación del correcto funcionamiento son los siguientes.

- 1^{er} caso. Obstáculo genérico

En este primer caso del Apéndice A se recoge un caso cualquiera en el cual no existe singularidad $[p_i, -p_i]$.

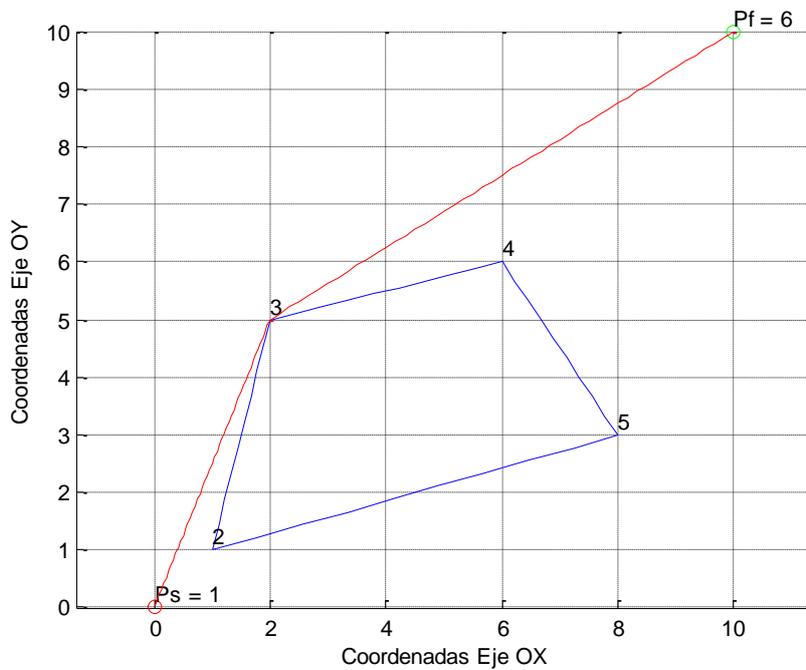


Figura 65 Caso 1 de la batería de pruebas para un obstáculo.

Nodos				
2	3	2	3	4
3	5	4	5	2
5	-	6	6	6

Tabla 12 Caso 1 de la batería de pruebas para un obstáculo. Visibilidad.

La solución propuesta por el programa recorre una distancia de 14.8191 unidades de longitud y para ello pasa por los puntos de paso 1, 3, 6, siendo este último el nodo final.

- 2º caso. Obstáculo que genere una singularidad en la recta analítica.

Este caso es debido a que la recta analítica puede ser vertical y por ello generamos una singularidad al ser el término dependiente de la ecuación de la recta, A, infinito. Lo que genera conflictos al calcular si el nodo "visible" está situado entre el nodo actual y la recta analítica o no. Los nodos con ángulos máximos y mínimos desde el nodo inicial son el 3 y el 5 respectivamente.

$$y = A \cdot x + B$$

$$A \rightarrow \infty$$

$$B \rightarrow \infty$$

Nodos				
3	3	2	3	4
4	5	4	5	2
5	6	6	-	6

Tabla 13 Caso 2 de la batería de pruebas para un obstáculo. Visibilidad.

La solución propuesta por el programa recorre una distancia de 11.6619 unidades de longitud y para ello pasa por los puntos de paso 1, 5, 6, siendo este último, el nodo final.

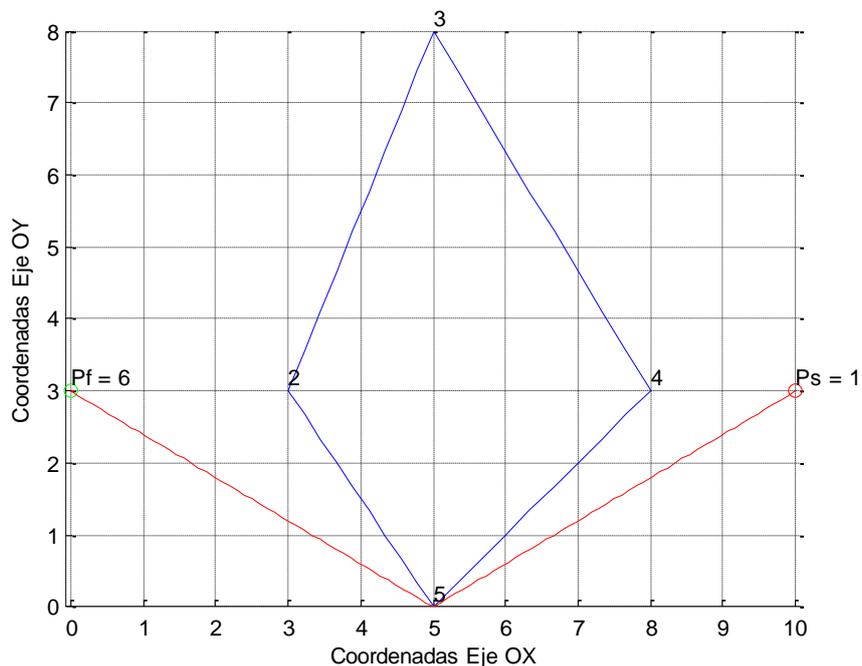


Figura 66 Caso 2 de la batería de pruebas para un obstáculo.

- 3er caso. Punto final sobre la recta que genera singularidad.

En el caso anterior se generaba una singularidad al calcular la recta. En este caso se va a tener la misma singularidad, con la única diferencia que en este caso el punto final se va a encontrar sobre la recta. Por lo tanto, el obstáculo será el mismo que en el caso anterior, con lo que únicamente variará la posición de los puntos inicial y final. En este caso debido a la ubicación de dichos puntos aparece la singularidad de $-\pi$, π .

El problema está planteado para comprobar que desde los vértices del obstáculo 2 y 4 se pueden observar el nodo final sin que la singularidad de la recta analítica conlleve ningún problema.

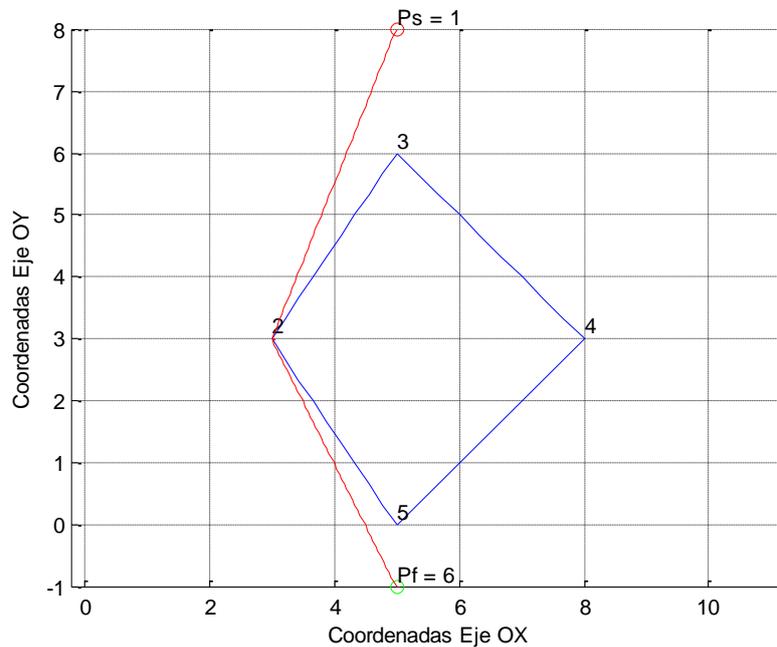


Figura 67 Caso 3 de la batería de pruebas para un obstáculo.

La solución propuesta por el programa recorre una distancia de 10.7703 unidades de longitud y para ello pasa por los puntos de paso 1, 2, 6, siendo este último, el nodo final.

Nodos				
2	3	2	3	4
3	5	4	5	2
4	6	-	6	6

Tabla 14 Caso 3 de la batería de pruebas para un obstáculo. Visibilidad.

- 4º caso. Vértices del obstáculo, Ps y Pf alineados

Este ejemplo sirve para verificar una de las hipótesis planteadas al inicio del trabajo. Si entre dos vértices existe uno alineado la trayectoria pasa por él, ya que este hecho no aumenta la distancia a recorrer.

Nodos				
2	3	2	3	4
5	5	4	5	2
-	-	6	6	-

Tabla 15 Caso 4 de la batería de pruebas para un obstáculo. Visibilidad.

La solución propuesta por el programa recorre una distancia de 9 unidades de longitud y para ello pasa por los puntos de paso 1, 2, 3, 6, siendo este último el nodo final.

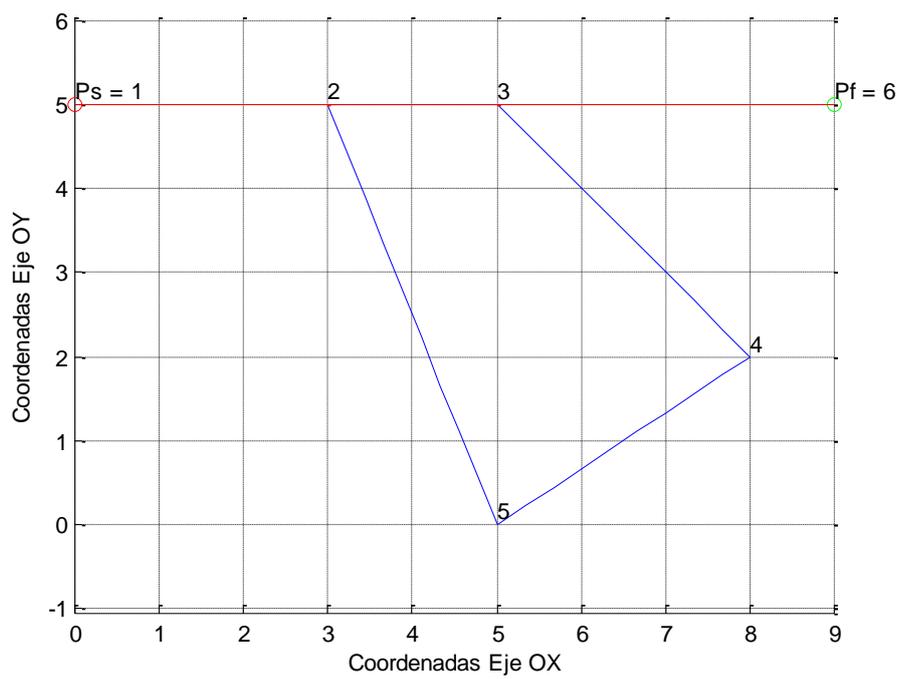


Figura 68 Caso 4 de la batería de pruebas para un obstáculo.

APÉNDICE B. BATERÍA DE CASOS PARA “N” OBSTÁCULOS

Al igual que para un obstáculo, para comprobar el correcto funcionamiento del programa, y que éste calcule lo que realmente debe calcular, se ha llevado a cabo un estudio de todo el código línea por línea, para obtener los puntos críticos del mismo y de esta manera plantear un caso para verificar que realmente el resultado obtenido es el esperado. Esta batería de casos se encuentra en el apartado que se presenta a continuación.

Salvo que sea totalmente necesario para verificar alguna condición, la batería de casos que se va a presentar constará únicamente de dos obstáculos dispuestos de la manera que se crea más conveniente para demostrar el correcto funcionamiento del programa.

La importancia de esta batería de casos reside en observar que los nodos “vecinos” son los adecuados y no otros, más que en comprobar que el programa genera la trayectoria adecuada.

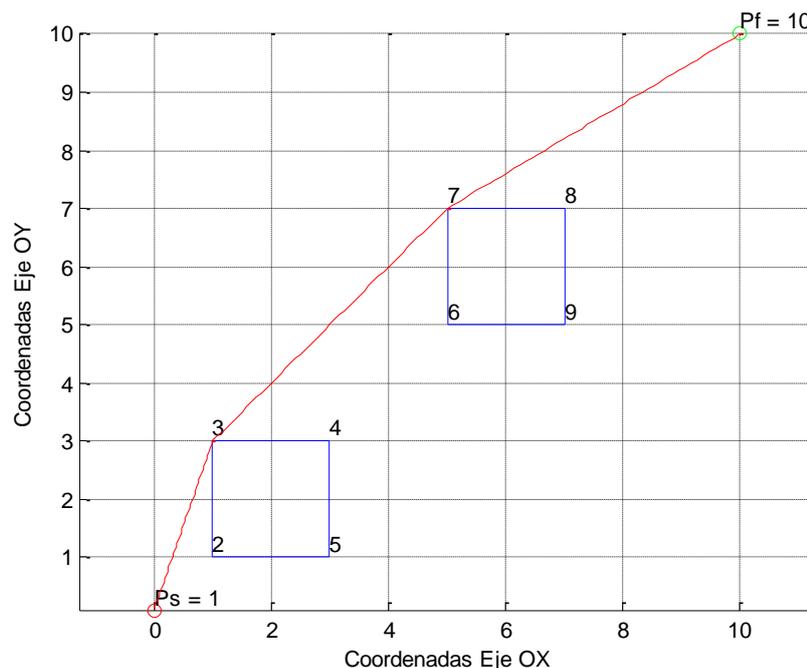


Figura 69 Caso 1 de la batería de pruebas para “n” obstáculos.

- 1er caso. Caso genérico (No existe singularidad $-p_i, p_i$).

Permite comprobar que el programa funciona de una manera correcta en un primer momento. Podría ser cualquiera de los casos expuestos en el apartado dedicado a “n” obstáculos.

Los nodos visibles desde cada uno de ellos serán por lo tanto los siguientes.

La solución propuesta por el programa recorre una distancia de 14.6501 unidades de longitud y para ello pasa por los puntos de paso 1, 3, 7, 10, siendo este último, el nodo final.

Nodos								
2	3	2	3	2	7	6	7	6
3	5	3	5	4	9	8	9	8
5	-	6	6	6	4	4	10	4
-	-	7	7	7	5	5	-	5
-	-	9	9	9	-	10	-	10

Tabla 16 Caso 1 de la batería de pruebas para “n” obstáculo. Visibilidad.

- 2º caso. Intercambio de los puntos inicial y final. Existe singularidad $-p_i, p_i$.

Como se indica, este caso se emplea para comprobar que efectivamente el programa es capaz de reconocer las singularidades introducidas al calcular los ángulos y actuar en consecuencia.

Nodos								
2	3	2	3	2	7	6	7	6
3	5	4	5	4	9	8	9	8
5	-	7	6	6	4	3	10	5
9	-	-	7	7	5	4	-	10
-	-	-	-	9	-	5	-	-
-	-	-	-	-	-	10	-	-

Tabla 17 Caso 2 de la batería de pruebas para “n” obstáculo. Visibilidad.

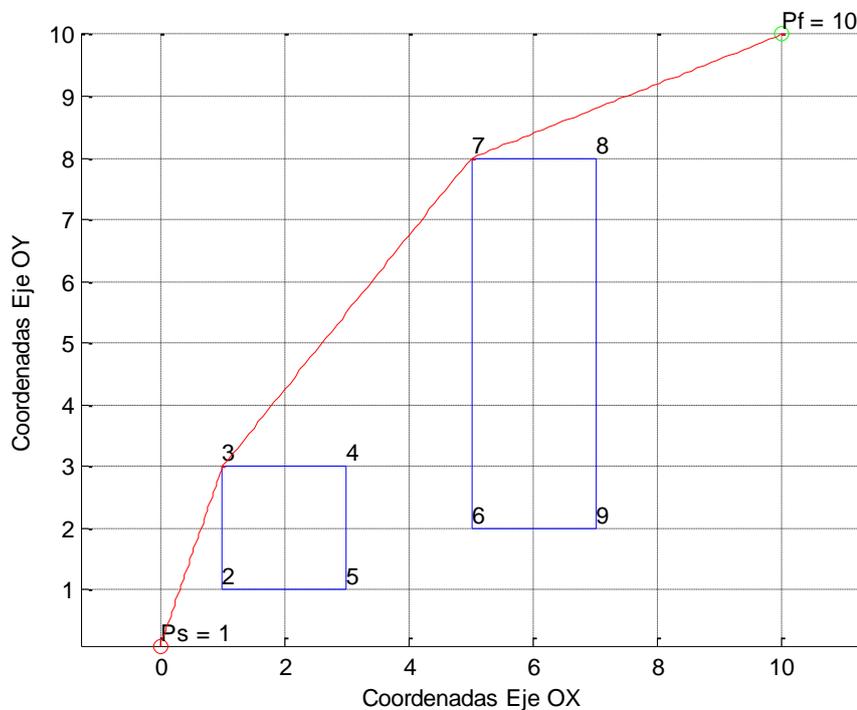


Figura 70 Caso 2 de la batería de pruebas para “n” obstáculos.

La solución propuesta por el programa recorre una distancia de 14.9506 unidades de longitud y para ello pasa por los puntos de paso 1, 3, 7, 10, siendo este último el nodo final.

- 3er caso. Nodos inicial y final alineados con las aristas de los obstáculos.

Permite comprobar si realmente la trayectoria recorre todos los vértices situados en una misma recta o simplemente va al último de ellos. Además, existirán rectas que posean singularidad. Este hecho tendrá especial importancia cuando los obstáculos posean algún ángulo cóncavo.

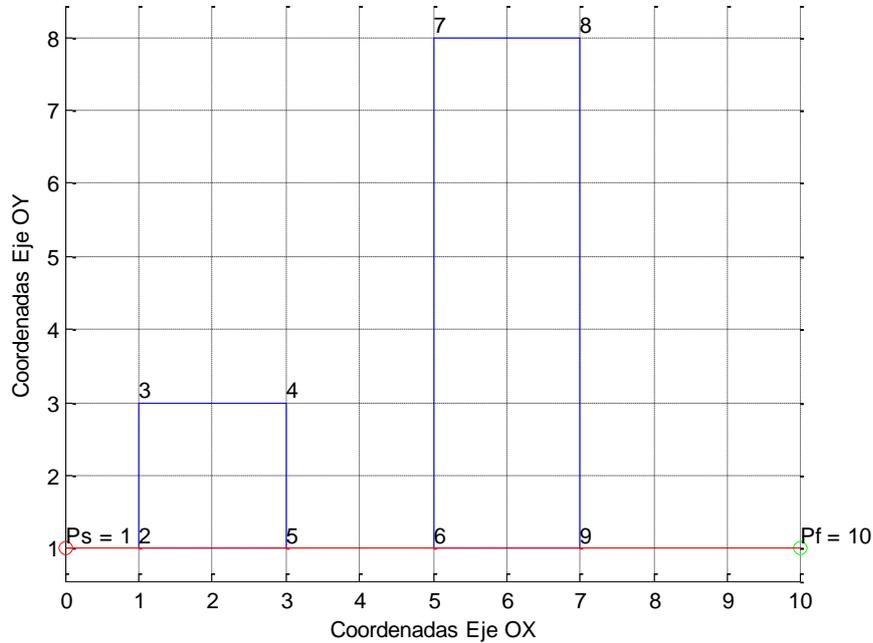


Figura 71 Caso 3 de la batería de pruebas para “n” obstáculos.

Nodos								
2	3	2	3	2	7	6	7	6
3	5	4	5	4	9	8	9	8
-	-	-	6	6	2	3	10	10
-	-	-	7	7	4	4	-	-
-	-	-	-	-	5	5	-	-

Tabla 18 Caso 3 de la batería de pruebas para “n” obstáculo. Visibilidad.

La solución propuesta por el programa recorre una distancia de 10 unidades de longitud y para ello pasa por los puntos de paso 1, 2, 5, 6, 9, 10, siendo este último, el nodo final.

- 4º caso. Comprueba cómo actúa el programa ante singularidades desde un vértice en el obstáculo al que pertenece y en otro distinto.

Este caso varía con respecto al anterior en que se debe tener en cuenta también que algunos vértices se encuentran en la singularidad $-p_i, p_i$. En la Tabla 19 se recoge la visibilidad desde cada uno de los nodos.

La solución propuesta por el programa recorre una distancia de 14.1421 unidades de longitud y para ello pasa por los puntos de paso 1, 2, 3, 6, 7, 10, siendo este último el nodo final.

- 5º caso. Existen puntos en la singularidad y puntos que no.

En el caso que se presenta a continuación en cambio la singularidad se encontrará para un determinado vértice en su propio obstáculo.

Nodos								
2	3	2	3	2	7	6	7	6
5	5	4	5	4	9	8	9	8
-	-	6	6	9	3	10	10	3
-	-	-	9	-	4	-	-	4
-	-	-	-	-	-	-	-	5

Tabla 19 Caso 4 de la batería de pruebas para “n” obstáculo. Visibilidad.

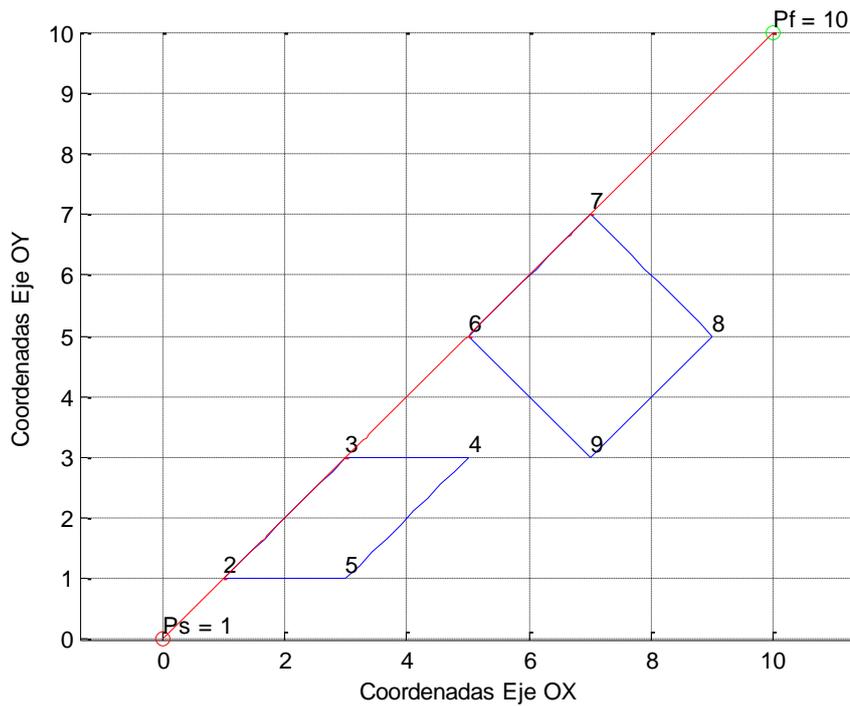


Figura 72 Caso 4 de la batería de pruebas para “n” obstáculos.

Nodos								
2	3	2	3	2	7	6	7	6
3	5	4	5	4	9	8	9	8
5	-	6	6	6	3	3	10	3
-	-	7	7	7	4	4	-	4
-	-	9	9	9	5	5	-	5
-	-	-	-	-	-	10	-	10

Tabla 20 Caso 5 de la batería de pruebas para “n” obstáculo. Visibilidad.

La solución propuesta por el programa recorre una distancia de 10.3246 unidades de longitud y para ello pasa por los puntos de paso 1, 3, 7, 10, siendo este último el nodo final.

Cabe destacar que el caso en el que se intercambian los nodos inicial y final no es necesario, ya que el vértice 6 tiene un caso igual que si se estudiase la visibilidad desde el nodo 10, que sería el inicial si se realizase el intercambio.

Con este caso se da por finalizado este apartado con el que se pretendía llevar a cabo un análisis del código a implementar para comprobar su funcionamiento. Por lo tanto se concluye indicando que el código funciona según lo esperado para obstáculos convexos.

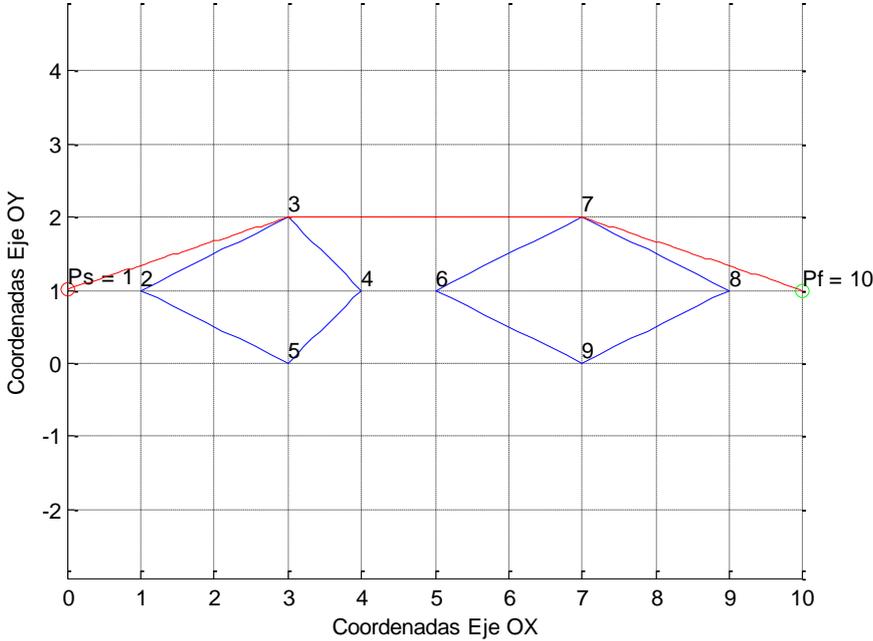


Figura 73 Caso 5 de la batería de pruebas para “n” obstáculos.

APÉNDICE C. BATERÍA DE CASOS PARA EL “EAR CLIPPING”

Al igual que en los dos apéndices anteriores, en éste se procede a mostrar qué el programa realmente funciona como debe hacerlo. Para ello se ha estudiado el código implementado línea a línea lo que nos ha llevado a plantear los ejemplos que se muestran a continuación.

En este apéndice se ha decidido estudiar el problema con un obstáculo al igual que en apéndice A con el fin de que el tiempo de ejecución por parte del programa sea menor y que la solución sea más sencilla y por lo tanto más fácil de comprobar a simple vista si realmente el programa funciona correctamente. Además esta parte del proyecto analiza uno a uno los obstáculos, por lo tanto es indiferente trabajar con uno o más obstáculos, pues el programa los analizaría por separado.

Cabe destacar también el hecho de que aunque únicamente se resuelva el problema con un obstáculo, no impide que se comprueben dos o más condiciones para el mismo ejemplo.

En nuestro programa se han diferenciado para un mismo obstáculo tres tipos de vértices. El vértice que se introduce en primer lugar, el que se introduce en último, y el resto de vértices. El otro elemento que hay que tener en cuenta para un correcto funcionamiento del programa es si al calcular el ángulo que forman las dos aristas de un vértice estas contienen la singularidad existente en $-\pi$, π originada al calcular dicho ángulo mediante la función “atan2”.

- 1er caso de la batería de pruebas.

En este caso se van a comprobar dos condiciones. La primera de ellas será para el primero de los vértices del obstáculo, y comprobaremos si realmente el procedimiento de triangulación funciona correctamente siendo este cóncavo y no teniendo en cuenta la singularidad mencionada.

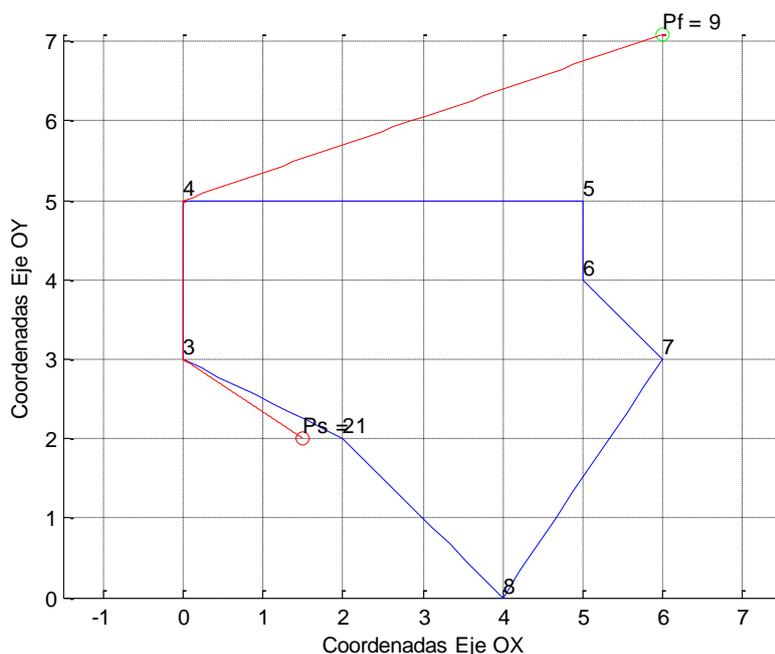


Figura 74 Caso 1 de la batería de pruebas para “Ear Clipping”.

La segunda comprobación se llevará a cabo para un vértice cualquiera y con ella se pretende estudiar el funcionamiento del programa en caso de que dicho vértice sea cóncavo y la arista de unión con uno de sus vértices contiguos sea vertical, este hecho se debe a que en caso de ser vertical existe una singularidad en nuestro problema. Además, en este caso se tendrá en cuenta la singularidad debida a $-\pi$, π .

En este primer caso de la batería de pruebas se tiene que la solución óptima consiste en pasar por los nodos 1,3,4 y 9, suponiendo ello una distancia total de 10.1597 unidades de longitud.

- 2º caso de la batería de pruebas.

La condición que hemos comprobado en el caso anterior para un vértice cualquiera va a ser comprobada a continuación tanto para el primer vértice del obstáculo como para el último. Por lo tanto ambos vértices van a estar unidos por una arista vertical, no conteniendo ninguno de los dos ángulos la singularidad originada por la función “atan2”.

Al mismo tiempo se va a llevar a cabo la comprobación de que para un vértice cualquiera funciona la triangulación en caso de que este sea cóncavo y el ángulo contiene a la singularidad comentada anteriormente.

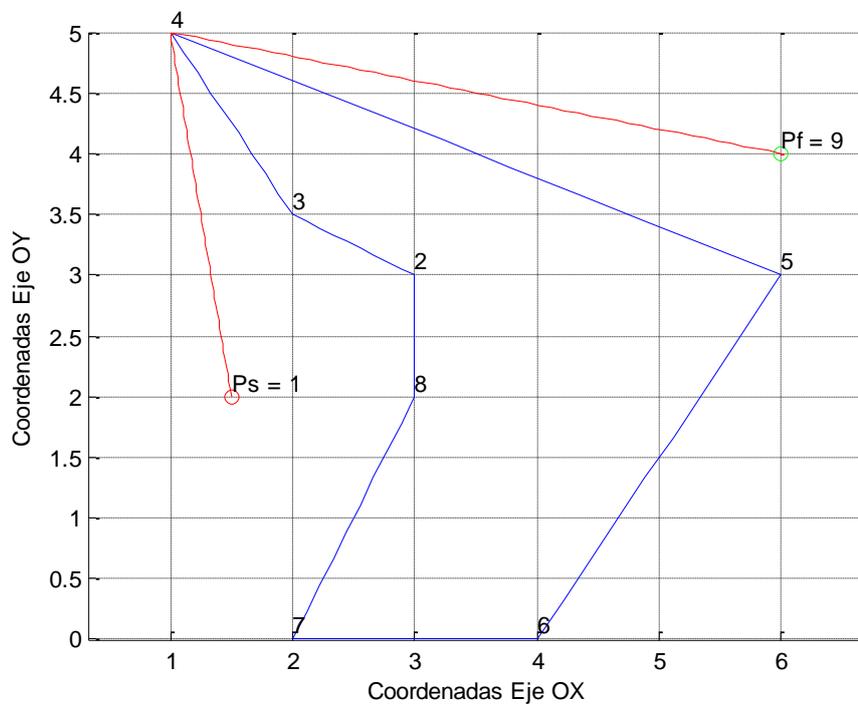


Figura 75 Caso 2 de la batería de pruebas para “Ear Clipping”.

En este caso de la batería de pruebas la solución únicamente nos hace pasar además de por los nodos inicial y final, por el nodo número 4 recorriendo para ello un total de 10.3215 unidades de longitud.

- 3er caso de la batería de pruebas.

Para este caso se van a comprobar las siguientes condiciones. En primer lugar que el tanto el primer como el último vértice estén unidos por una arista vertical, así como que estos vértices contengan a la singularidad generada por la función “atan2”.

Por otra parte se va a comprobar que para un vértice cualquiera todo funcione según lo previsto en caso de que una de las aristas sea verticales y no exista la singularidad $-\pi$, π .

En este caso la solución obtenida nos lleva a recorrer una distancia de 10.3541 unidades de longitud, teniendo que pasar para ello por los nodos 1, 6, 7 y 9, que es el nodo final.

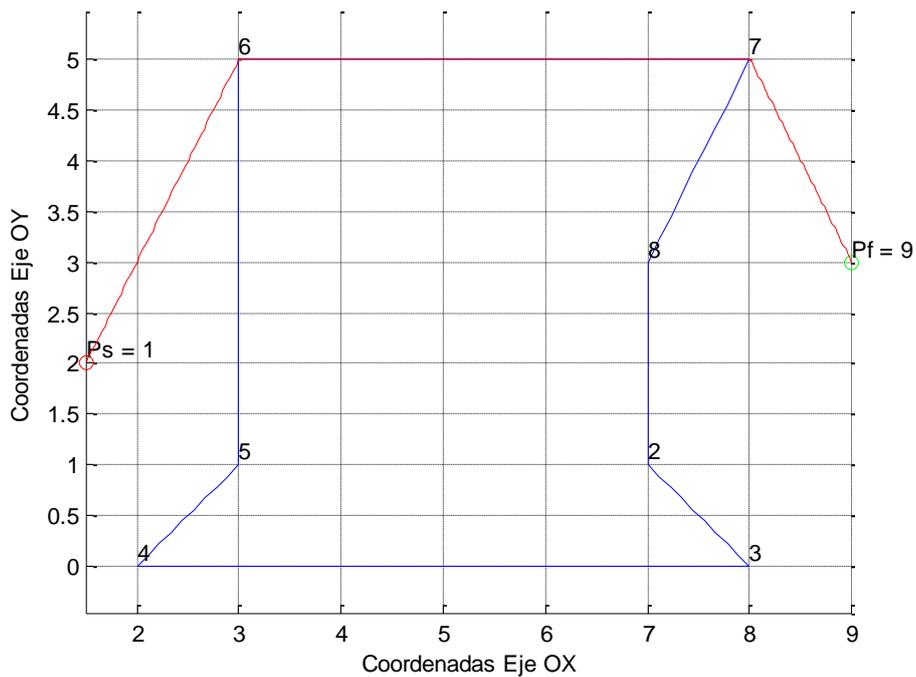


Figura 76 Caso 3 de la batería de pruebas para “Ear Clipping”.

- 4º caso de la batería de pruebas.

Este ejemplo va a determinar dos casos más que funcionan correctamente. Tanto para el primer como el último vértice se va a comprobar que aun conteniendo a la singularidad generada por la función “atan2” y siendo sus aristas de dirección cualquiera la triangulación resulta efectiva.

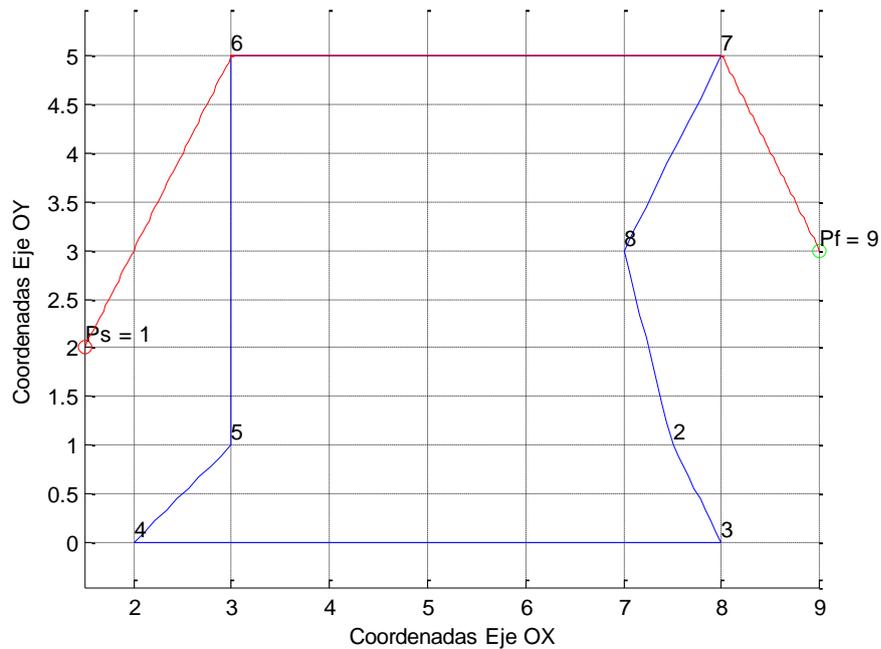


Figura 77 Caso 4 de la batería de pruebas para “Ear Clipping”.

La ruta dada por el programa, como se puede ver en la Figura 77, nos lleva a recorrer los nodos 1, 6, 7 y 9, empleando para ello una distancia total de 10.3541 unidades de longitud.

- 5º caso de la batería de pruebas.

Este es el último caso que deberemos de comprobar en este apéndice del proyecto. En él se va a comprobar que tanto para un vértice cualquiera como para el último del obstáculo se lleva a cabo la triangulación de la manera deseada. Las características del obstáculo serán tales que la aristas tendrán dirección cualquiera y los ángulos formados por las mismas no contendrán a la singularidad $-\pi$, π .

La solución dada por el programa nos lleva a tener que visitar los nodos 1, 7, 6 y 9, siendo la distancia invertida para tal propósito de 6.2803 unidades de longitud.

Con este caso se concluye la batería de pruebas necesaria para comprobar que el programa era capaz de estudiar el camino de menor longitud en caso de que los obstáculos fuesen convexos.

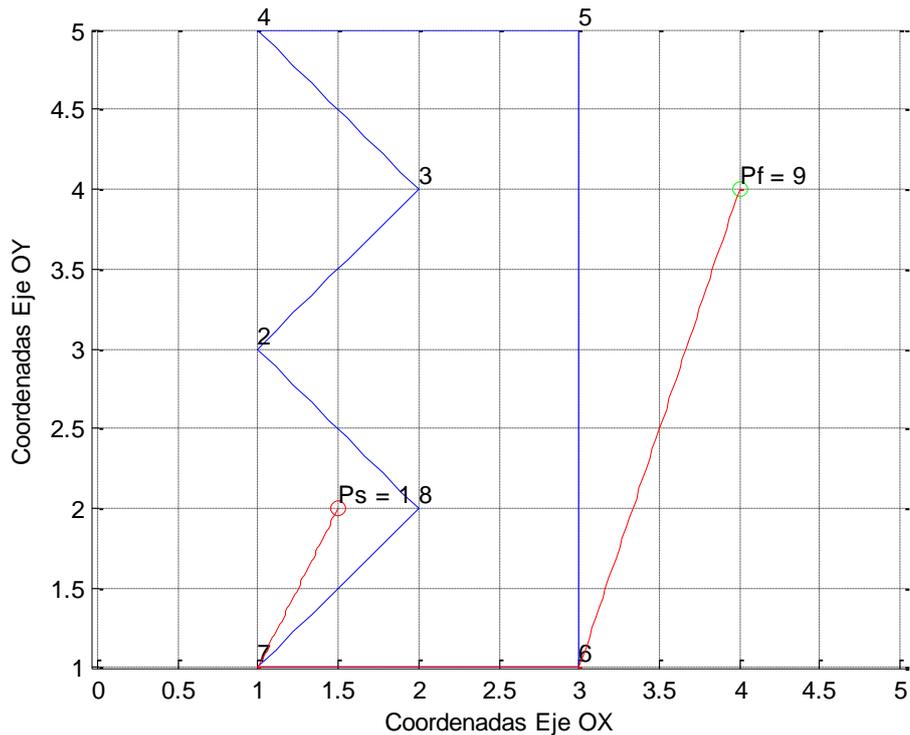


Figura 78 Caso 5 de la batería de pruebas para “Ear Clipping”.

APÉNDICE D. BATERÍA DE CASOS

COMPROBACIÓN DE LA MÍNIMA DISTANCIA DE TRAMO RNAV

En este epígrafe como indica el título se van a mostrar, al igual que en los apéndices anteriores los diferentes casos que nos van a permitir determinar que el programa funciona correctamente y que los resultados obtenidos son los esperados. Para una mayor simplificación se van a usar ejemplos en los que haya el menor número de obstáculos, uno, dos o tres. Se debe comprobar que en cualquier dirección se aumente la distancia entre los puntos de paso, y que en cada dirección funcionen los dos puntos obtenidos mediante la prolongación de las rectas que interseccionan en el punto de paso a modificar.

- 1er caso de la batería de pruebas.

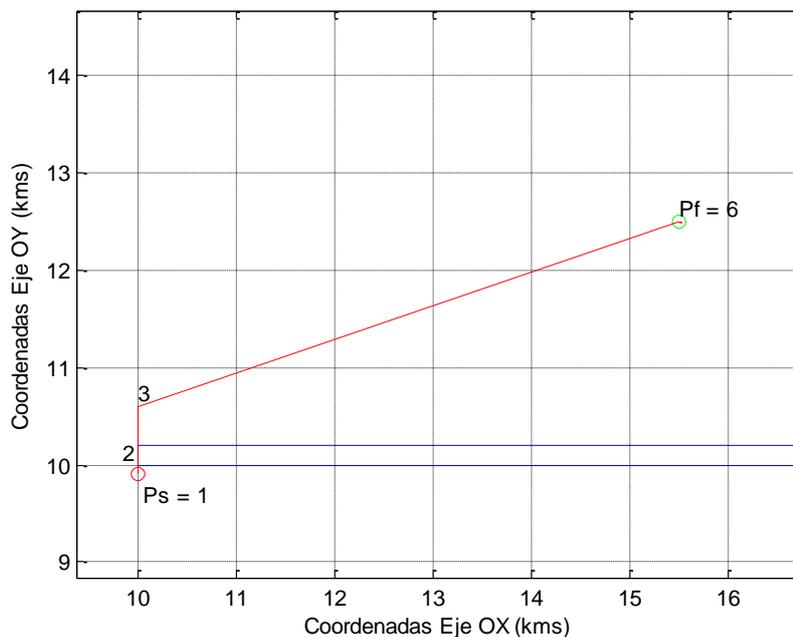


Figura 79 Caso 1 de la batería de pruebas para distancia de tramo RNAV.

En este primer caso se recoge una posible situación. En ella se comprueba que si la recta de unión entre dos puntos tiene una orientación de 90 grados como en el caso de la recta que une los vértices 2 y 3. La solución esperada se obtiene mediante la prolongación de la recta que une los nodos 2 y 3. Como se puede observar, el programa, ha ubicado el punto de paso 3 en un lugar diferente al vértice donde se encontraba inicialmente para así cumplir con la normativa.

- 2º caso de la batería de pruebas.

En esta ocasión el caso es similar al anterior, salvo que en lugar de llevarse a cabo la prolongación de la recta 2-3, al no ser posible, por existir un obstáculo impidiéndolo, se modifica la posición del punto de paso 3 sobre la

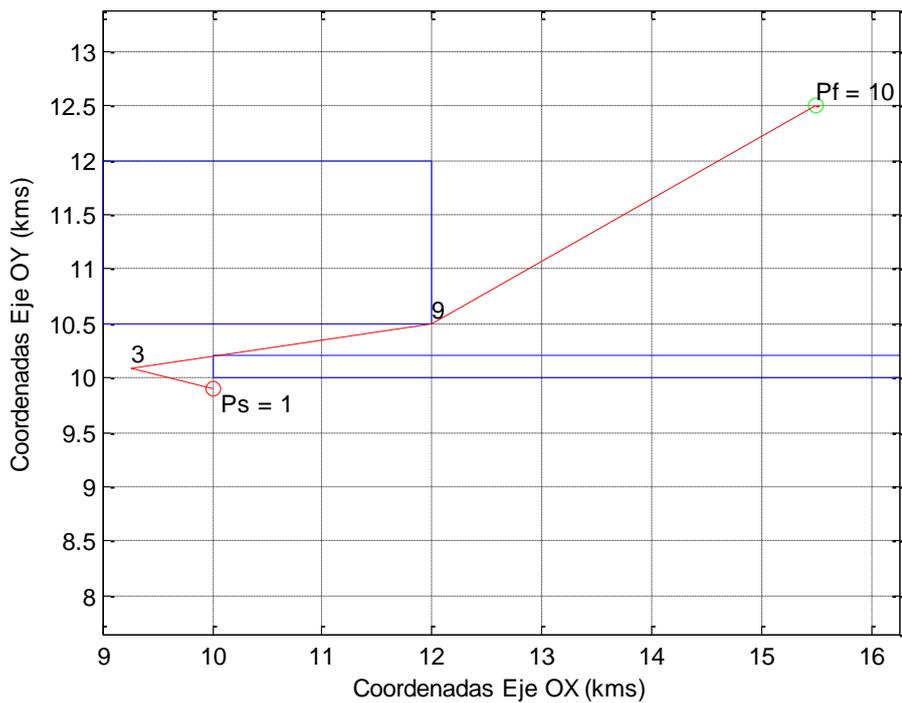


Figura 80 Caso 2 de la batería de pruebas para distancia de tramo RNAV.

recta que une los vértices 3 y 9.

En este caso también se puede comprobar el correcto funcionamiento del programa. De hecho, en un principio, la trayectoria recorría el vértice 2, pero al modificar la posición del vértice 3, éste se hace visible desde 1, siendo innecesario recorrer el mencionado punto de paso 2.

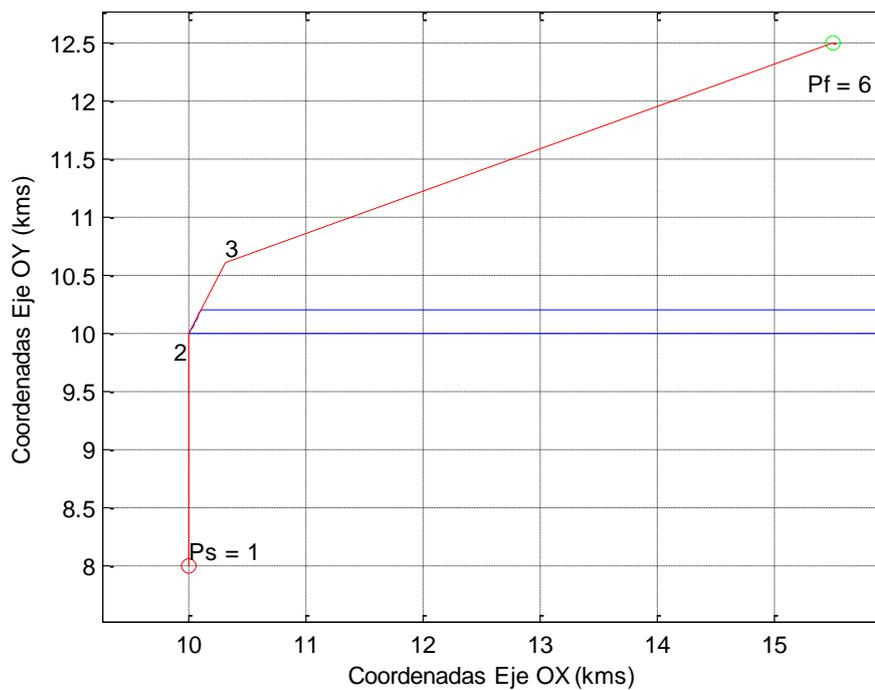


Figura 81 Caso 3 de la batería de pruebas para distancia de tramo RNAV.

- 3er caso de la batería de pruebas.

En este caso, la recta que une los puntos dos y tres, tiene una cierta pendiente, pero al ser su distancia menor que la mínima de estabilización, también hay que proporcionarle una mayor distancia. En la Figura 81 se observa que la solución es la esperada

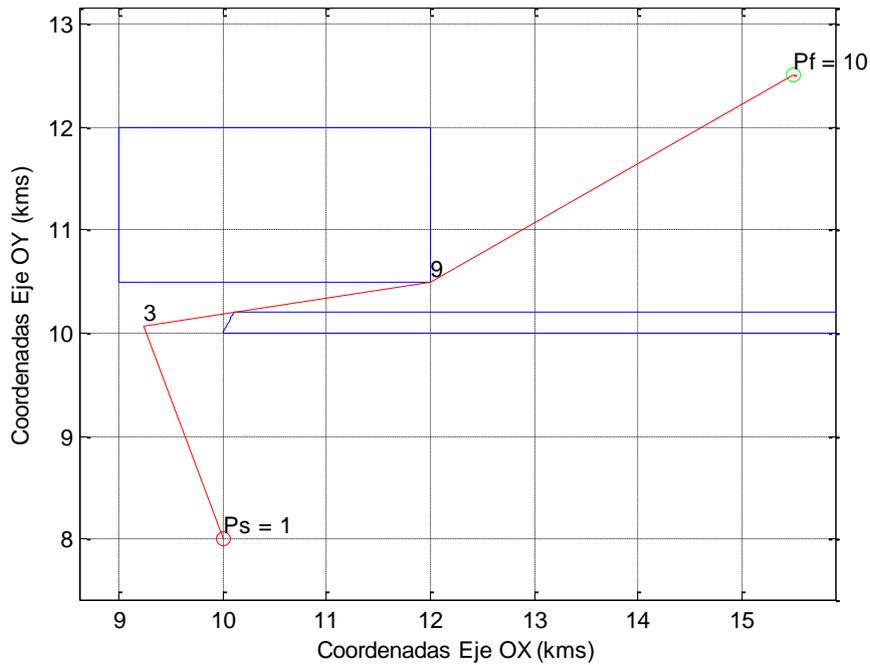


Figura 82 Caso 4 de la batería de pruebas para distancia de tramo RNAV.

- 4º caso de la batería de pruebas.

En esta ocasión tendremos una situación parecida a la anterior, con la diferencia de que en lugar de prolongar la recta que une los puntos de paso 2 y 3, se prolonga la que une los puntos 3 y 9 ya que al existir un obstáculo, impide que la solución sea la del caso anterior.

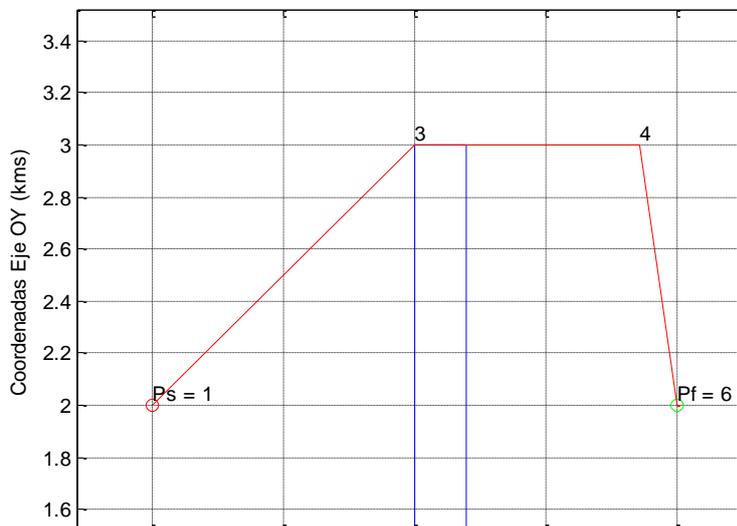


Figura 83 Caso 5 de la batería de pruebas para distancia de tramo RNAV.

- 5º caso de la batería de pruebas.

En esta ocasión se va a comprobar el correcto funcionamiento en el caso de que dos de los puntos de paso estén situados en una recta paralela al eje OX, siendo la distancia que los separa menor de la permitida.

- 6º caso de la batería de pruebas.

En este caso se estudia un caso similar al anterior con la salvedad de que no será posible la primera de las soluciones, al encontrarse un obstáculo en las proximidades, el cual impide la prolongación de la recta 3-4, en su lugar la prolongación se da en la dirección de la recta 4-9.

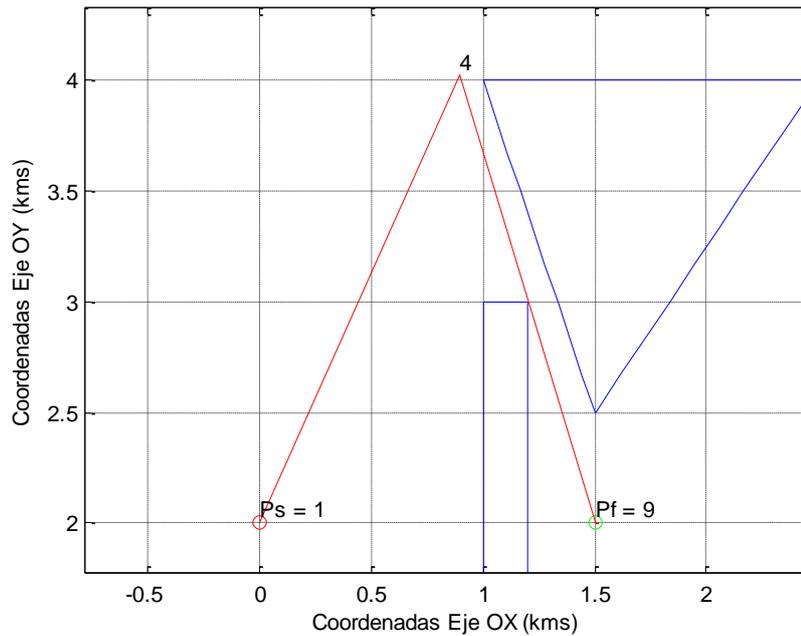


Figura 84 Caso 6 de la batería de pruebas para distancia de tramo RNAV.

Se puede comprobar cómo no es necesario recorrer el punto de paso 3, debido a que en este caso el punto de paso 4 es visible desde el nodo inicial.

- 7º caso de la batería de pruebas.

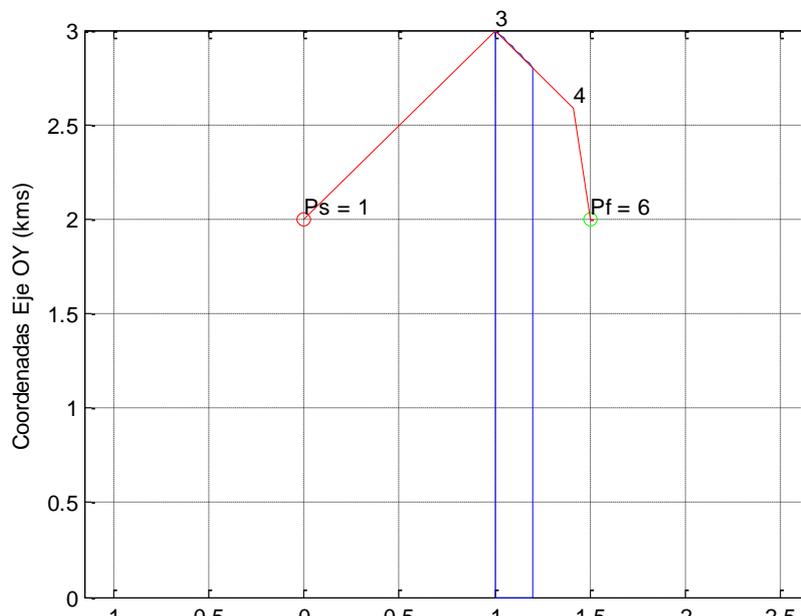


Figura 85 Caso 7 de la batería de pruebas para distancia de tramo RNAV.

Tanto este caso como el siguiente van a mostrar el correcto funcionamiento del programa en el caso de que la recta que una dos puntos de paso no tenga la distancia necesario y su orientación sea con una pendiente negativa y en la dirección creciente del eje OX.

- 8º caso de la batería de pruebas.

Como se ha comentado en el caso anterior, éste es idéntico, con la diferencia de que no será posible prolongar la recta que unía los vértices 3-4, por lo que se ha prolongado la recta 4-6.

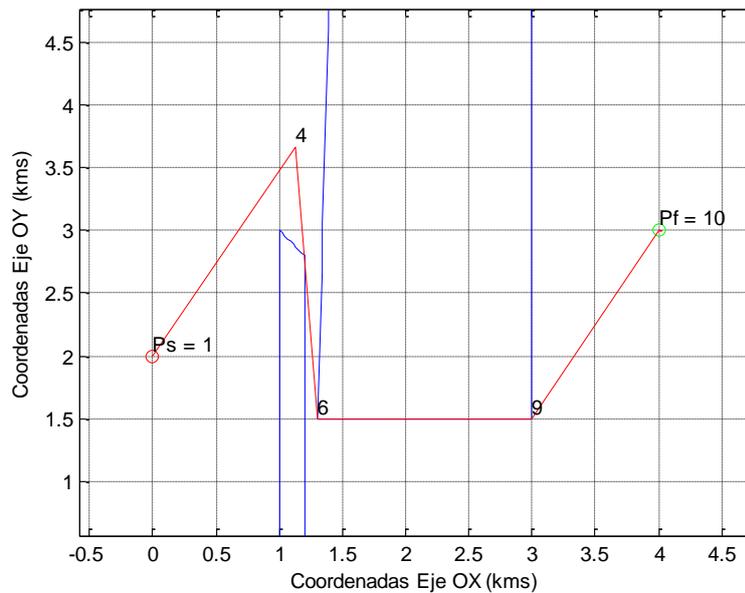


Figura 86 Caso 8 de la batería de pruebas para distancia de tramo RNAV.

- 9º caso de la batería de pruebas.

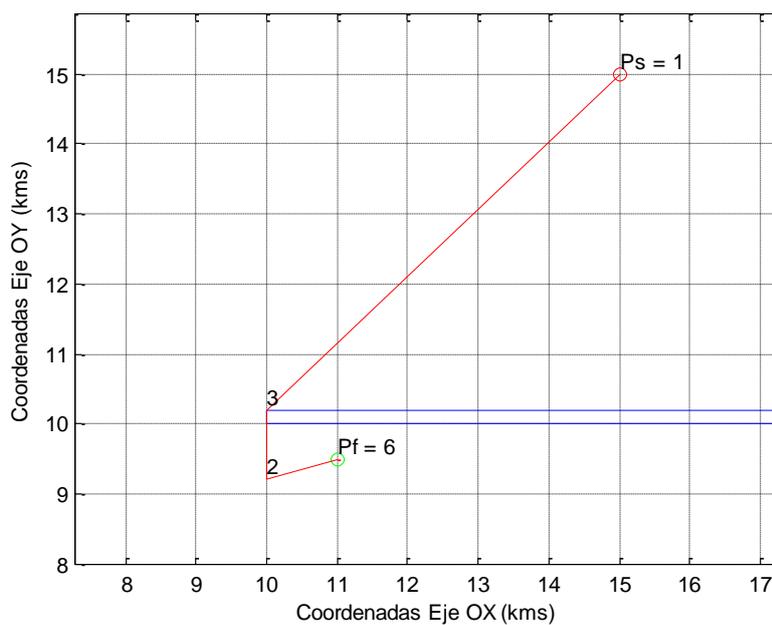


Figura 87 Caso 9 de la batería de pruebas para distancia de tramo RNAV.

Este caso es similar al primero de los estudiados, con la salvedad de que los puntos inicial y final están

intercambiados.

- 10° caso de la batería de pruebas.

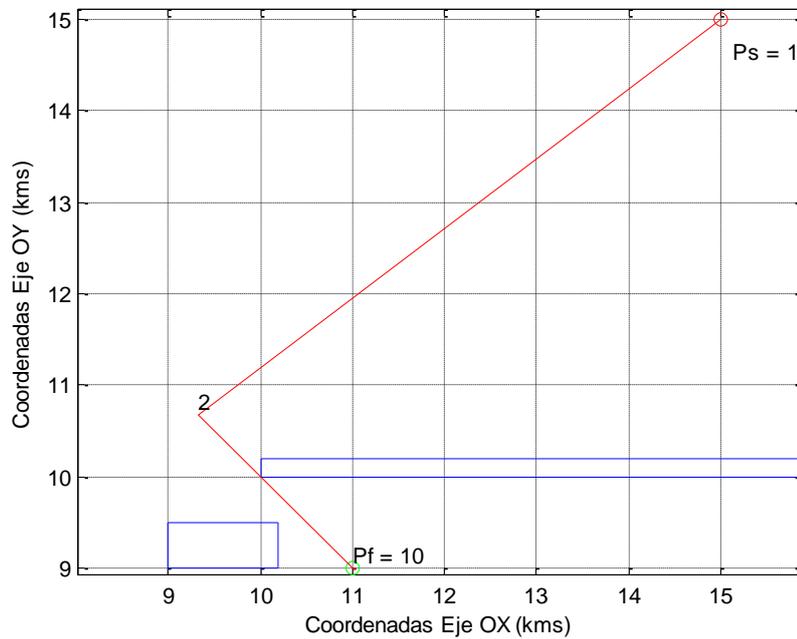


Figura 88 Caso 10 de la batería de pruebas para distancia de tramo RNAV.

Este caso es una mezcla del segundo y del noveno, pues no se va a poder prolongar la recta que une los vértices 3 y 2 como en el caso anterior, por el contrario se prolonga la recta 2 y 10.

- 11° caso de la batería de pruebas.

A continuación se muestra un ejemplo en el cual se recoge el hecho de que la recta a prolongarse forme un ángulo comprendido entre 180 y 270 grados con respecto al eje OX. Se puede comprobar que se produce la

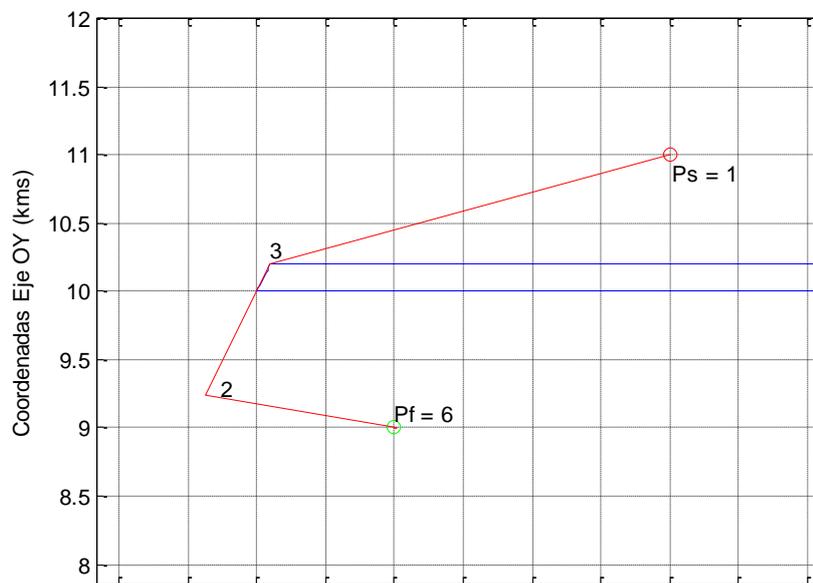


Figura 89 Caso 11 de la batería de pruebas para distancia de tramo RNAV.

prolongación de la recta que une los puntos de paso 3-2.

- 12º caso de la batería de pruebas.

Como ha ocurrido en el resto de situaciones, en ésta se va a comprobar que, si fuese imposible prolongar la recta que une los dos puntos de paso que no cumplen la mínima distancia de estabilización, se puede prolongar la recta que une el segundo de los puntos de paso con el siguiente de ellos.

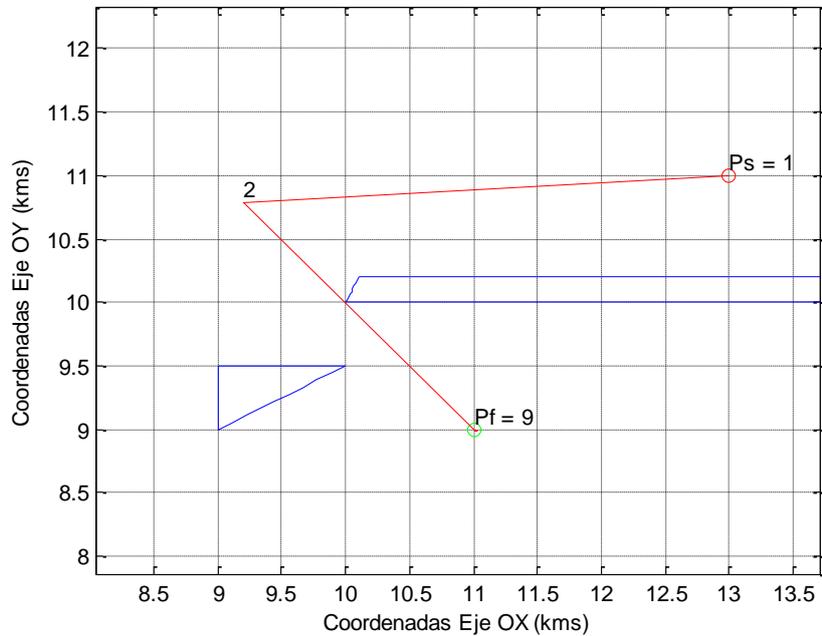


Figura 90 Caso 12 de la batería de pruebas para distancia de tramo RNAV.

- 13er caso de la batería de pruebas.

En esta ocasión se va a comprobar el caso en que la recta que une dos puntos de paso sea horizontal y la distancia entre los puntos menor que la mínima permitida.

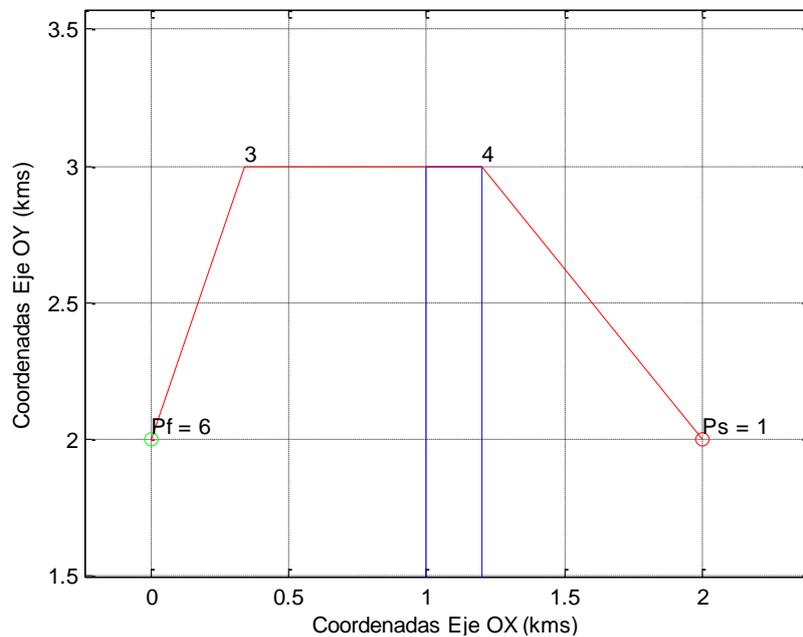


Figura 91 Caso 13 de la batería de pruebas para distancia de tramo RNAV.

- 14º caso de la batería de pruebas.

Esta situación va a permitir comprobar un caso igual al anterior con la salvedad de que no se va a poder prolongar la recta que une los dos puntos de paso que no están separados por la distancia permitida, sino que se deberá prolongar la recta que une el segundo de los puntos de paso con su siguiente.

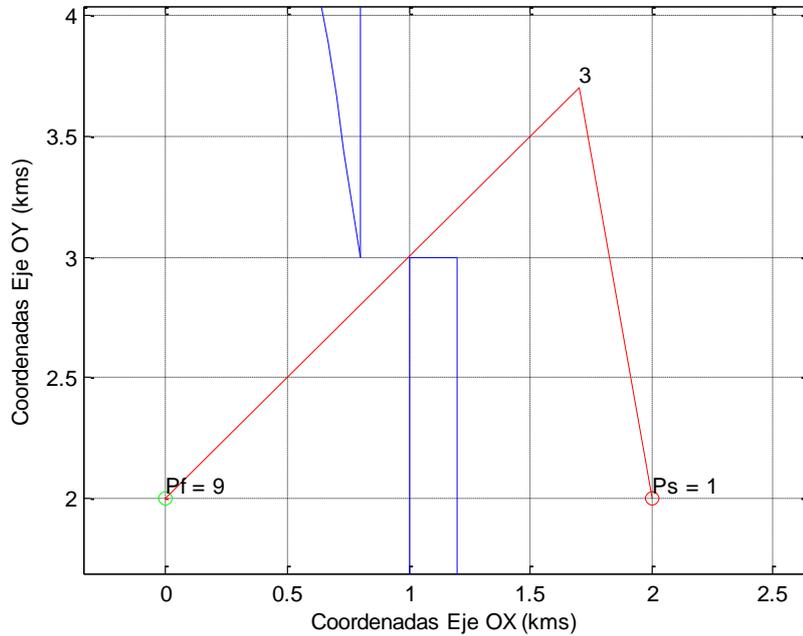


Figura 92 Caso 14 de la batería de pruebas para distancia de tramo RNAV.

- 15º caso de la batería de pruebas.

Con este ejemplo se desea mostrar el correcto funcionamiento en caso de que la recta que une dos puntos de paso tenga una orientación comprendida entre 90 y 180 grados, pudiendo prolongarse la recta que une ambos puntos.

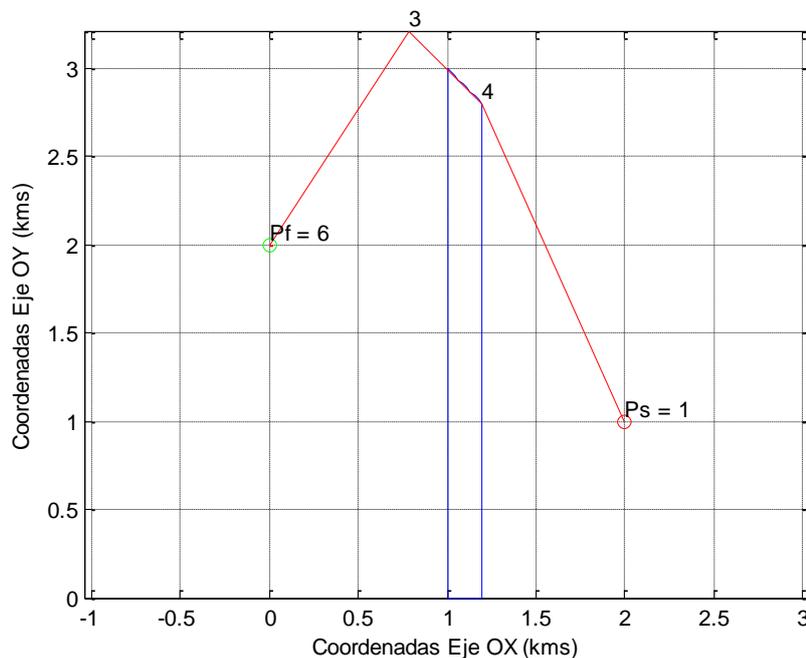


Figura 93 Caso 15 de la batería de pruebas para distancia de tramo RNAV.

- 16° caso de la batería de pruebas.

Este es el último caso correspondiente a este apéndice, en él se va a mostrar el caso en el cual la recta entre los puntos que no cumplen la distancia mínima de separación entre dos puntos de paso no pueda prolongarse debido a que exista un obstáculo que lo impida.

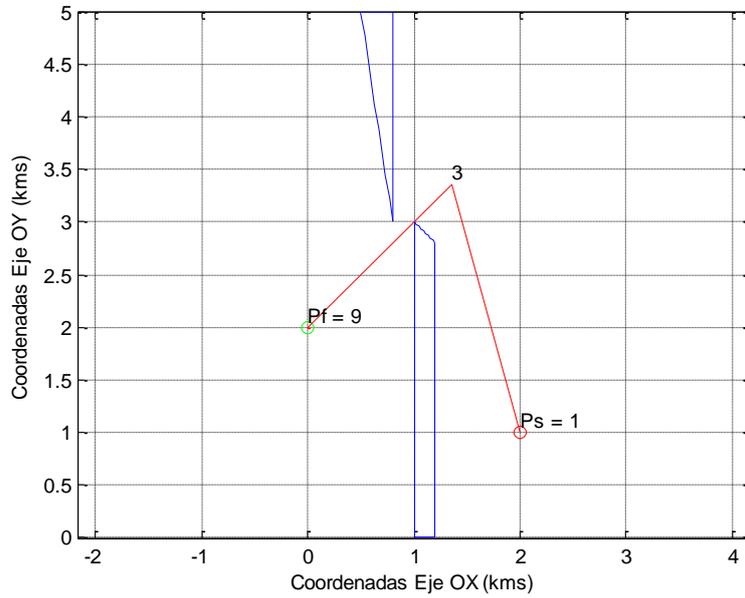


Figura 94 Caso 16 de la batería de pruebas para distancia de tramo RNAV.

APÉNDICE E. BATERÍA DE CASOS PARA LA COMPROBACIÓN DE LA PROTECCIÓN LATERAL

En este último apéndice se va a mostrar que la aplicación de la normativa acerca de la protección lateral, es decir, el recreído, se lleva a cabo de la manera adecuada. Para ello de nuevo se pasa a revisar el código línea a línea obteniendo del mismo los casos que deben de ser estudiados, para de este modo determinar que el funcionamiento es el correcto.

Para este apéndice se van a utilizar ejemplos en los que existan varios obstáculos, pues debido a la estructura del código, el programa realiza el cálculo de la protección lateral de todos los obstáculos al mismo tiempo.

La mayor complicación de esta parte del proyecto ha sido determinar hacia qué lado se debía aplicar el recreído, es decir, si las aristas vienen definidas mediante la ecuación de la recta, $y=ax+b$, si el recreído debía de sumársele al término independiente o si por el contrario era necesario sustraérsele.

Al no influir este recreído a la resolución del problema, no se va a presentar el problema completo resuelto, sino que únicamente se mostrarán los obstáculos con su recreído correspondiente.

- 1er caso de la batería de pruebas.

En este primer caso se van a estudiar dos obstáculos, el primero de ellos es un octógono regular, el cual nos permite, gracias a sus propiedades geométricas, estudiar todas las posiciones relativas del recreído con respecto al obstáculo. Por su parte el segundo de ellos, va a ser un hexágono con dos lados cóncavos de manera que se comprueba que el código funciona aun siendo los obstáculos cóncavos.

Se puede observar a simple vista que el código ha ubicado el recreído en la posición adecuada y por lo tanto, la resolución de este caso es la esperada.

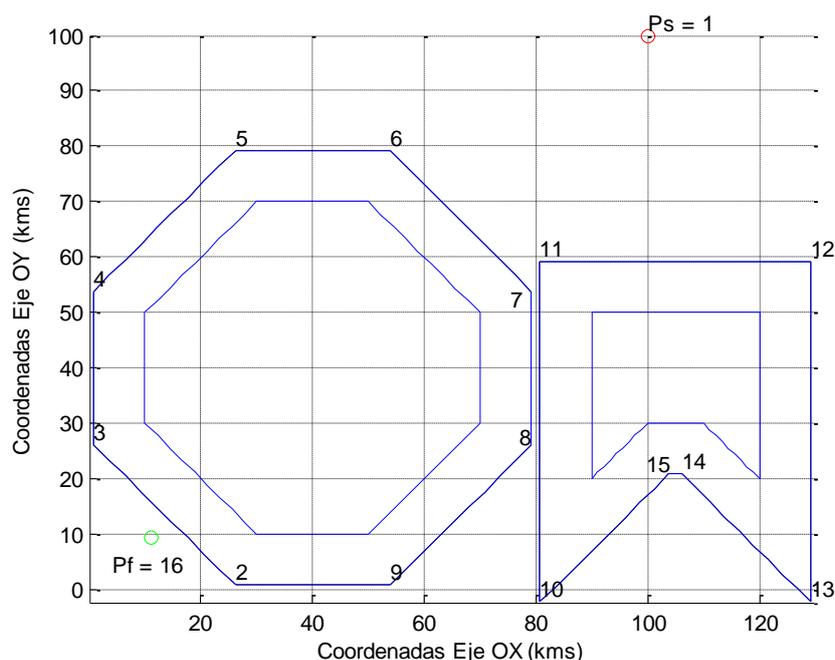


Figura 95 Caso 1 de la batería de pruebas para la protección lateral.

- 2º caso de la batería de pruebas

En el primero de los casos se tuvo en cuenta que el segundo de los obstáculos tuviese dos vértices cóncavos para estudiar la ubicación relativa del recrecido con respecto al obstáculo. En este caso se van a estudiar las otras seis posiciones relativas del recrecido frente al obstáculo.

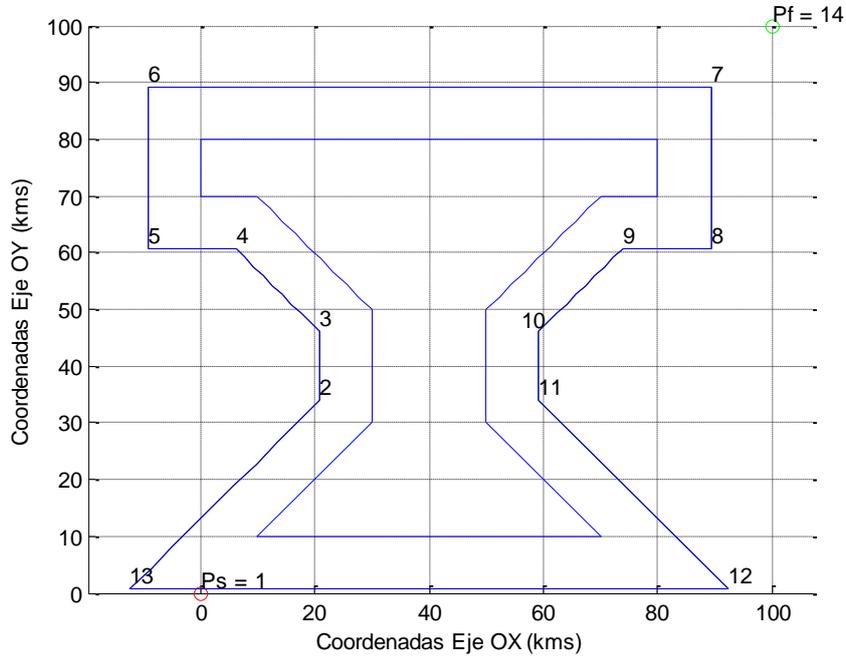


Figura 96 Caso 2 de la batería de pruebas para la protección lateral.

Para esta prueba también se obtiene la solución deseada, concluyendo que el código funciona como es debido, pues se han realizado todas las pruebas necesarias. En un primer momento puede parecer que es un apéndice corto, pero si tenemos en cuenta que en cada caso se han comprobado más de seis posiciones relativas del recrecido con respecto al obstáculo, podemos concluir que no es tan corto como puede pensarse.

Con este último apéndice se da por concluida la memoria del proyecto fin de carrera.

Córdoba a 9 de enero de 2015