



AMBATO

PLANIFICACIÓN DE TRAYECTORIAS SUAVES PARA VEHÍCULOS AÉREOS NO TRIPULADOS DE ALA FIJA

Gloria Isabel Vanegas Zabala, Ph. D.
Franklin Eduardo Samaniego Riera, Ph. D.

Revisión Técnica:
Editorial CEDIA

Corrección de Estilo:
Editorial CEDIA

Diagramación: Paz Cordero González

Coordinación:
Laura Malache S. - Editorial CEDIA

Una publicación de la Editorial CEDIA,
arbitrada por pares académicos de doble ciego.

CEDIA
cedia Gonzalo Cordero 2-111 y
J. Fajardo
Cuenca - Ecuador
cedia.edu.ec

Primera edición

ISBN: 978-9978-77-769-5

Ambato, Ecuador
Septiembre de 2025



This book is licensed by Creative Commons,
Attribution-NonCommercial 4.0 International
<https://creativecommons.org/licenses/by-nc/4.0/>

PLANIFICACIÓN DE TRAYECTORIAS SUAVES PARA VEHÍCULOS AÉREOS NO TRIPULADOS DE ALA FIJA

Gloria Isabel Vanegas Zabala, Ph. D.
Franklin Eduardo Samaniego Riera, Ph. D.

CONTENIDO

1. Introducción	16
1.1. Aplicaciones UAV	17
1.2. Planificación de trayectoria	19
1.3. Planificación de trayectoria discreta y continua	21
1.4. Planificación de trayectorias suaves	21
2. Trayectorias de vuelo suaves 3D, y su aproximación Bézier hacia las curvas Clotoides 3D	24
2.1. Introducción	25
2.2. Definición de curvas suaves	25
2.2.1. Curvas en el espacio	27
2.2.2. Clotoides panares 2D	29
2.2.3. Curvas Bézier	30
2.3. Control de seguimiento y Actitud	32
2.3.1. Generación de trayectorias suaves	32
2.3.2. Aproximación a la Clothoide 3D	34
2.4. Experimentos y Resultados	35
2.5. Conclusiones	41
3. Planificación de trayectorias 3D limitadas por restricciones de maniobrabilidad en UAVs	42
3.1. Introducción	43
3.2. Descripción del modelo UAV	44

3.2.1.	Sistema no-holonómico UAV	45
3.2.2.	Radio de Curvatura	46
3.3.	Definición de curvas	47
3.3.1.	Algoritmo de De Boor	48
3.3.2.	Aproximación de curvatura con semicircunferencias	50
3.4.	Experimentos y resultados	54
3.4.1.	Planificación de trayectorias	55
3.5.	Conclusiones	57
4.	Optimización multiobjetivo para la construcción de Trayectorias suaves 3D	60
4.1.	Introducción	61
4.2.	Preliminares	63
4.2.1.	Optimización multiobjetivo	63
4.3.	Definición del problema	65
4.4.	Metodología	67
4.4.1.	Definición del segmento esférico	67
4.4.2.	Definición de segmentos de línea recta	76
4.5.	Experimentos y resultados	77
4.5.1.	Ejemplo aplicativo	79
4.6.	Conclusiones y trabajos futuros	85
Bibliografía		91

A mis padres, por su amor incondicional y por acompañarme con su apoyo y
ejemplo en cada etapa de mi vida.

G.V.Z.

A mis padres, gracias por ser el mejor ejemplo en todo sentido.

F.S.R.

Preámbulo

Este libro realiza un primer conjunto de propuestas orientadas a la construcción de trayectorias suaves para entornos euclidianos tridimensionales. Dichas trayectorias se emplean con el objetivo de proporcionar una guía para el seguimiento de vuelo para vehículos aéreos no tripulados (UAV o unmanned aerial vehicle en inglés, de donde vienen sus siglas) con características no-holonómicas (aviones de ala fija). Las trayectorias desarrolladas buscan reducir posibles movimientos bruscos durante el vuelo de este tipo de aeronaves. En este marco, en el libro se aborda diferentes temáticas, incluyendo las siguientes:

- Curvas suaves
- Curvas Bézier
- Curvas clotoides 3D
- Trayectorias suaves 3D
- Optimización multiobjetivo

La teoría y matemática desarrollada y propuesta en este libro intenta ser simple y de fácil entendimiento. En este sentido, se trabaja un nivel matemático razonablemente simple, confiando en la intuición del lector. Finalmente, se presenta diversos experimentos que han probado de forma satisfactoria los algoritmos planteados, por medio de plataforma de programación y cálculo numérico MATLABTM ¹.

El libro presenta un enfoque académico apropiado para la introducción y el estudio de la planificación de trayectoria suave 3D, para ingenieros que buscan alcanzar nuevos entendimientos y métodos en esta rama de la ciencia en automatización y robótica.

¹Matlab es una marca registrada de Math Works, Inc.

Los contenidos de este libro están fuertemente ligados a 2 trabajos doctorales, que fueron realizados en la Universidad Politécnica de Valencia (UPV), en el Instituto de Automática e Informática Industrial (ai2), y en el Departamento de Ingeniería de Sistemas y Automática (DISA).

Es importante destacar y resaltar el apoyo brindado por la Pontificia Universidad Católica del Ecuador - Sede Ambato, ya que sin su soporte no habría sido posible culminar la publicación de este trabajo. Por este motivo, les extendemos nuestro sincero sentimiento de agradecimiento y estima.

Finalmente, es importante resaltar que este libro presenta el primer volumen de nuestro trabajo. De modo que, en el segundo volumen se presentará contribuciones orientadas al desarrollo algorítmico, matemático y experimental implicado en la planificación de trayectoria suave 3D.





01

Introducción

En los últimos años se han producido diversas e importantes aplicaciones que buscan completar la tarea de planificación de trayectoria suave y navegación aérea en el espacio tridimensional 3D. De manera similar, se ha desarrollado diversos e importantes avances prácticos en el desarrollo de UAV (vehículos aéreos no tripulados o drones), así como en la navegación autónoma, que han sido presentados y aceptados en la comunidad científica. De ahí que, en la actualidad, debido a la transformación digital y la automatización, el estudio y la investigación de UAV adquieran una relevancia estratégica por su impacto en diversos ámbitos. Entre estos avances, vale señalar aquellos enfocados en la competitividad, por medio de innovaciones en inteligencia artificial, sensores, navegación autónoma y eficiencia energética, que han posicionado a países y empresas a la vanguardia de la industria 4.0.

En este contexto, el impacto económico relacionado con esta tecnología implica la generación de empleos en sectores emergentes, enfocados en pilotos, desarrolladores de software o analistas de datos. Estos aspectos fortalecen las cadenas de valor relacionadas con la manufactura y los servicios. Por este motivo, invertir en tecnología UAV no solo optimiza procesos críticos, sino que también impulsa la soberanía tecnológica y la solución de los desafíos sociales y ambientales. Ahora bien, dicho propósito requiere de la colaboración de gobiernos, academia y sector privado con el objetivo de maximizar beneficios y mitigar riesgos.

1.1 Aplicaciones UAV

Los avances y aportes científicos relacionados directamente con el desarrollo de UAV son amplios y relevantes. A continuación, se describe de forma breve un conjunto

de trabajos disponibles. Uno de ellos, el trabajo realizado por Niven [1], detalla el desarrollo de un UAV de ala fija construido con materiales compuestos (fibra de carbono y espuma de PVC). Este diseño integra una cámara multiespectral con el objetivo de completar misiones de mapeo aéreo.

El trabajo presentado por Esakki aprovecha las técnicas de fabricación aditiva (AM) para la fabricación de un cuerpo unificado del bastidor de un quadrotor para minimizar el tiempo de montaje [2]. Por otra parte, Chung [3] describe la propuesta de un vehículo híbrido que combina despegue vertical (VTOL) con vuelo eficiente en crucero. Este prototipo usa motores eléctricos y una célula de combustible de hidrógeno, alcanzando 120 km de autonomía. Este diseño y su posterior fabricación han generado diferentes resultados atractivos en cuanto a la propulsión eléctrica, lo que ha permitido definir los requisitos de rendimiento, incluyendo la velocidad de pérdida, la velocidad máxima, la altitud de crucero y el radio y la velocidad de giro. Además, la carga alar y la carga de potencia asociada se obtienen a partir de los requisitos de rendimiento. De forma similar, el trabajo presentado por Yuan describe el diseño y construcción de un dron de tipo cuadricóptero, para lo cual se utiliza diferentes componentes comerciales de bajo costo y que posee una capacidad de carga útil de hasta 500g con una autonomía de 15 minutos. [4].

En Mingjie [5] se sintetiza el diseño típico de los UAV de ala fija VTOL en modo de vuelo plano. A la vez, se realiza una breve descripción de las diferencias en cuanto al modo de despegue y potencia. Por otro lado, Hakim [6] presenta estudio y desarrollo de un UAV de ala fija para realizar tareas implicadas en el proceso de vigilancia, cartografía y lanzamiento. Como resultado de este proceso de cartografía, se genera datos fotográficos que se convierten en un mapa ortofotográfico. En cambio, el estudio presentado por Yixuan describe el diseño y la fabricación de un UAV rentable para fines logísticos, construido principalmente con materiales de madera y con capacidad de carga útil de hasta 1.000g a una altitud de 40 m. [7].

El trabajo presentado por Gu en [8] describe el desarrollo integral de un sistema de tipo UAV VTOL, desde los aspectos que incluyen el diseño y la implementación de la aeronave, la integración de los dispositivos de a bordo, el soporte de la estación de tierra y la comunicación a larga distancia. Además, incluye el análisis aerodinámico, el diseño mecánico y el desarrollo del controlador. Por último, este desarrollo se

valida por medio de experimentos aerodinámicos relacionados con la estabilidad de vuelo, la resistencia y la autonomía deseada.

Chipade [9] expone el diseño conceptual y la demostración de vuelo de prueba de un novedoso UAV biplano cuadricóptero de paso variable. Este diseño combina las capacidades de despegue y aterrizaje vertical (VTOL) y de planeo de un cuadricóptero con las características de autonomía, resistencia y alta velocidad de crucero de una aeronave de ala fija. Además, se describe una misión de transporte y entrega de 6 kg de carga útil a un destino situado a 16 km del punto de origen.

Oliveira [10] diseña e implementa una plataforma de prototipado rápido multivehículo. El objetivo del trabajo es apoyar al desarrollo y las pruebas de control y navegación para UAV. De ahí que, la arquitectura de hardware concebida para el entorno de prototipado incluya un sistema óptico de captura de movimiento y un conjunto de ordenadores externos que gestionan la comunicación entre sistemas y ejecutan programas de usuario para varios quadrotors.

En estos trabajos, se puede apreciar una amplia difusión de las aplicaciones de UAV para el mercado. Este hecho ha permitido el desarrollo de algoritmos cada vez más rápidos con respuestas robustas.

Por otro lado, como se detalla en este libro, el uso y la aplicación de las metodologías revisadas permite asimilar de forma clara los conceptos implicados en la resolución del problema de planificación de trayectorias suaves. De esta manera, se intenta alcanzar un aprendizaje continuo, con el apoyo de diversos experimentos prácticos en diferentes escenarios.

1.2 Planificación de trayectoria

El problema de la planificación de trayectorias puede ser definido como la determinación de un camino por el que un robot móvil se desplaza. El objetivo es determinar el conjunto de espacios libres de colisión dentro de un ambiente o entorno de trabajo (es decir, el espacio euclíadiano definido en \mathbb{R}^2 o \mathbb{R}^3), en el que se han ubicado diversos obstáculos [11, 12, 13]. El robot inicia su trayectoria desde un punto de este espacio y debe alcanzar un punto final, evitando los diversos obstáculos que se encuentran dentro del entorno; tal como se ilustra en la Figura 1.

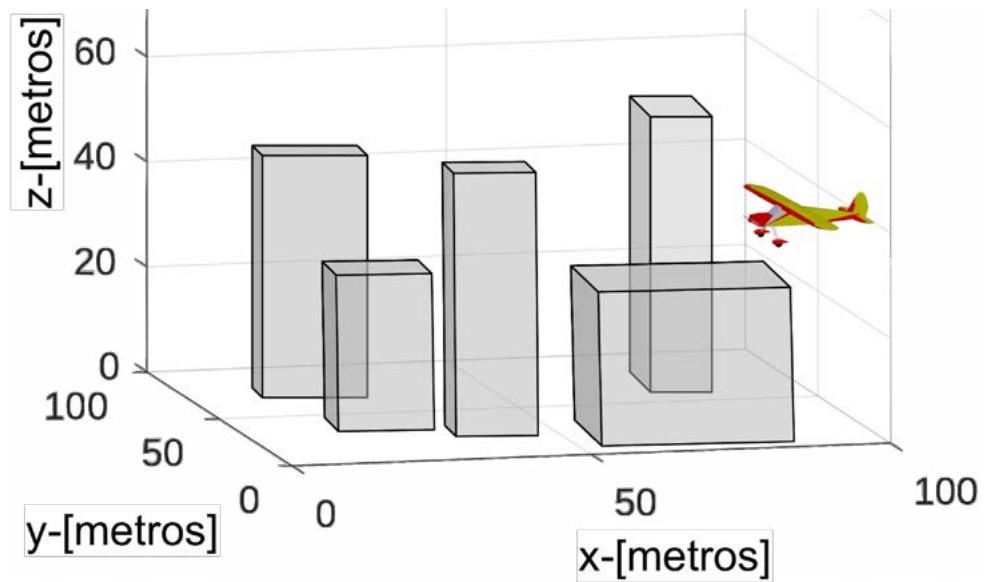


Figura 1.1:Ejemplo de escenario de vuelo con obstáculos.

La planificación de trayectoria determina una sucesión de configuraciones específicas que permiten trasladar al robot desde un estado inicial hasta un estado final. Se describe como estado a la descripción de la posición y orientación del robot, en referencia a un marco absoluto expresado por la combinación de las coordenadas cartesianas [14, 15, 16] del centro de gravedad del robot y la orientación angular desde su eje principal.

Este campo de estudio específico se encuentra enmarcado por la robótica móvil [17, 18, 19], y se resuelve con el apoyo de la automatización de sistemas [20, 21, 22]. Para ello, se requiere de un procesamiento amplio de volúmenes de información, proveniente de sensores y actuadores. En consecuencia, la capacidad computacional es un requerimiento relevante, pues se engloba dentro de la categoría de problemas de programación de naturaleza NP-completo [23, 24, 25].

En este contexto, resulta necesario estudiar y proponer nuevos algoritmos que disminuyan el esfuerzo computacional para la planificación de trayectoria, que transformen órdenes de alto nivel en comandos de movimiento de bajo nivel, ejecutables por el robot.

1.3 Planificación de trayectoria discreta y continua

El tratamiento del entorno de trabajo puede ser abordado a través de 2 metodologías fundamentales, que implican el tratamiento del espacio euclíadiano como información continua o discreta [26, 27]. Independiente de la metodología de trabajo, el objetivo es determinar de 2 conjuntos de información dentro del entorno de trabajo, siendo:

1. El conjunto de los espacios por los que el robot móvil no puede moverse (es decir, los espacios ocupados por diversos obstáculos)
2. El conjunto de espacios libres de colisión (es decir, los espacios por donde el robot puede moverse).

Para la determinación de estos conjuntos, una alternativa de estudio viable se enfoca en el análisis geométrico del entorno de trabajo [28, 29, 30]

La Planificación de trayectoria discreta [31, 32, 33] construye mallas con una forma definida dentro del entorno de trabajo, mientras que la planificación de trayectoria continua [34, 35, 36], recurre a los números pseudo aleatorios para determinar la existencia de obstáculos o espacios libres. Entonces, a partir de la definición del conjunto de espacios libres, se construyen trayectorias navegables por el robot.

El conjunto continuo de puntos libres de colisión, unidos a través de líneas rectas, constituye las trayectorias que el robot puede seguir para alcanzar su objetivo. Ahora bien, en muchos casos esta posible trayectoria implica giros bruscos a lo largo del entorno de trabajo. Por lo que, es importante destacar que solamente un robot de características holonómicas es capaz de completar esta clase de trayectorias.

1.4 Planificación de trayectorias suaves

El objetivo de la Planificación de trayectorias suaves [37, 38, 39] es la construcción de curvas continuas, es decir, líneas en las que la sucesión de puntos cambia de dirección de forma que el vector tangente tuerce su dirección sin formar aristas. En geometría diferencial, el vector tangente es el vector velocidad de la curva, que indica su dirección de movimiento. En específico, los robots que poseen características

no-holonómicas [40, 41, 42], requieren esta clase de curvas suaves, pues carecen de las habilidades necesarias para realizar maniobras bruscas, ya sea en tierra o en el aire.

Es importante resaltar que este libro desarrolla diferentes propuestas para la creación y construcción de curvas suaves, que sirven como aproximaciones a trayectorias suaves, navegables por UAV de características no-holonómicas, específicamente, aviones de ala fija.

El libro se organiza en 4 capítulos, donde el Capítulo 1, realiza una breve introducción de la temática de planificación de trayectoria. El Capítulo 2 describe una primera aproximación de curva suaves a través de Bézier hacia las curvas clotoídes. El Capítulo 3, propone una construcción de trayectorias suaves, tomando en cuenta las características restrictivas de maniobrabilidad de los UAV. Finalmente, en el Capítulo 4, se recurre al concepto del problema de optimización multiobjetivo para la construcción de curvas suaves y su aproximación hacia las trayectorias suaves.

02

Trayectorias de vuelo suaves 3D, y su aproximación Bézier hacia las curvas Clotoides 3D

RESUMEN

Este capítulo realiza el estudio de matemático y metodológico de las curvas en el espacio euclíadiano o 3D. Además desarrolla un breve estudio de las curvas Bézier y las curvas clotoides. En especial, se enfoca en la definición de las curvas clotoides, sus características y alcances. El objetivo es describir brevemente la construcción de trayectorias suaves, a partir de las curvas suaves.

2.1 Introducción

El área de los vehículos aéreos no tripulados (UAV) ha evolucionado de forma importante en los últimos años. Los UAV fueron inicialmente concebidos para fines militares, pero hoy en día existe un gran número de aplicaciones comerciales [43, 44].

No obstante, una de las principales desventajas de los UAV es que sus sistemas de control son no lineales y algunos de ellos también tienen restricciones no-holonómicas para la navegación en condiciones normales.

Para controlar estos vehículos, a menudo se utiliza curvas espaciales para generar trayectorias suaves. Ahora bien, la mayoría de estas curvas no son intuitivas, ya que, no tienen en cuenta las restricciones del vehículo y/o requieren procedimientos de optimización, que pueden no ser viables para aplicaciones en tiempo real. En este sentido, una amplia variedad de tipos de curvas, como las de Bézier o spline polinómicas han sido utilizadas con el objetivo de alcanzar una posición determinada en el espacio euclíadiano 2D o 3D. [45, 46, 47]

El análisis geométrico, en el contexto de las curvas, se representa por medio de un mapa continuo en el espacio dimensional (desde la dimensión 1 hasta la dimensión n), cuyo dominio tiene derivadas continuas, hasta un orden específico, concepto que se utiliza a menudo en la navegación robótica continua.

En específico, las curvas clotoides o espirales de Euler presentan algunos aspectos geométricos de aspecto y de seguridad interesantes, que han sido utilizados en diversas aplicaciones reales, como en el diseño de carreteras, ferrocarriles y montañas rusas [48, 49], además, del control de vehículos no-holonómicos [50, 51, 52].

La solución a las curvas clotoides se determina, a través del cálculo de las integrales de Fresnel, si bien no existe una solución cerrada para esta curva. Sin embargo, algunas aproximaciones, con errores de hasta 10^{-20} , permiten el cálculo en tiempo real [53, 54, 55, 56, 57]. Ahora bien, a pesar que no existe una solución analítica, es posible calcular de forma analítica algunas de sus propiedades geométricas, como la curvatura y el ángulo tangente en función de la longitud de arco.

Este capítulo se centra en la generación de trayectorias suaves 3D para pilotar UAV no-holonómicos (aviones de ala fija). En este contexto, debido a las interesantes características de las clotoides, sus propiedades y su suavidad, el objetivo es aproximar las clotoides 3D mediante curvas racionales Bézier para conseguir un rendimiento en tiempo real. El esquema de control se divide en dos etapas: planificador local y control cinemático, como se describe en la Figura 2.

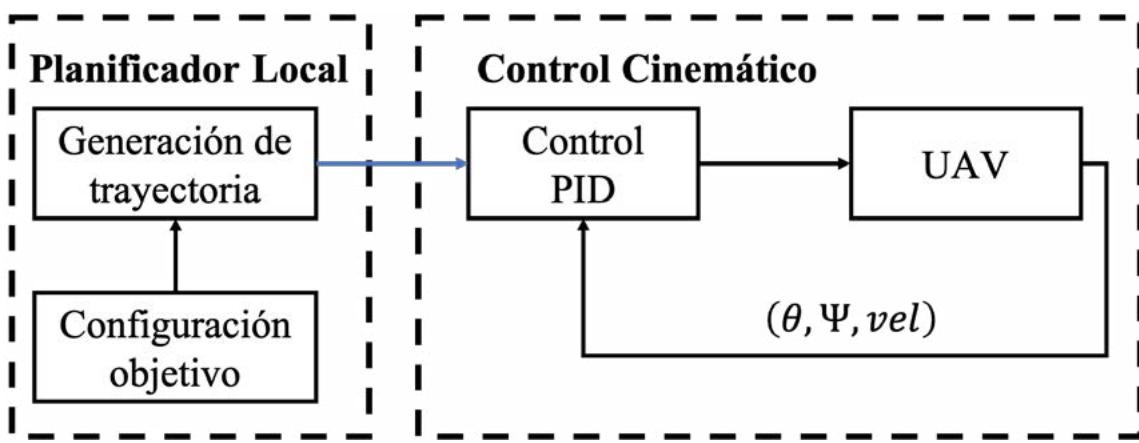


Figura 2.1: Diagrama de planificación y control.

El capítulo está organizado de la siguiente forma. La sección 2.2 introduce algunos preliminares sobre las curvas suaves, las clotoides y las curvas de Bézier; en la sección 2.3 se explica la metodología propuesta para aproximar las clotoides 3D mediante curvas de racionales Bézier; a continuación, en la sección 2.4, se realizan algunas pruebas de simulación de vuelo, cuyos resultados se muestran y discuten; finalmente en la sección 2.5 se extraen algunas conclusiones.

2.2 Definición de curvas suaves

Dentro del campo de la geometría, una curva se describe como una línea continua y suave que sigue una trayectoria definida por reglas matemáticas o por propiedades geométricas. Estas curvas, representadas mediante ecuaciones o construcciones geométricas, pueden ser planas 2D (como una parábola) o espaciales 3D (como una hélice).

2.2.1 Curvas en el espacio

Construir una curva en el espacio consiste en definir una función vectorial continua y diferenciable que describa un camino en el espacio tridimensional 3D. Esta curva se define mediante una parametrización, usualmente con respecto al tiempo o a otro parámetro escalar, y permite representar trayectorias que un objeto o sistema puede seguir en un entorno 3D. En este sentido, es importante destacar dos conceptos primordiales que se desprenden del concepto de curva en el espacio:

- 1) **Curvatura:** La curvatura mide cuánto se desvía una curva de ser una línea recta en un punto dado. Se trata de una cantidad que indica cómo cambia la dirección de la tangente a la curva conforme avanzamos sobre ella, de modo que, si la curvatura es grande, la curva está cambiando de dirección bruscamente, es decir, está “más doblada”.
- 2) **Torsión:** La torsión mide cuánto se “retuerce” o “sale del plano” una curva en el espacio.. Es decir, indica la variación de la dirección del plano osculador de la curva, mostrando si la curva se mantiene plana o se extiende tridimensionalmente. Así, una torsión distinta de cero indica que la curva gira fuera de un solo plano, haciendo que la curva sea verdaderamente espacial.

En relación directa con las fórmulas de Frenet- Serret, que describen el movimiento de un objeto o punto a lo largo de una curva en el espacio tridimensional, el lector puede comprender este concepto imaginando que camina por un sendero curvo en 3D: no solo avanza, sino que también puede girar o inclinarse, como se ilustra en la Figura 3.

De manera formal, cualquier curva espacial $\mathbf{C}(s)$, parametrizada por su longitud de arco s en un espacio tridimensional \mathbb{R}^3 , está determinada por su curvatura $\kappa(s)$ y su torsión $\tau(s) \neq 0$, de acuerdo con el teorema fundamental de las curvas espaciales [58]. Intuitivamente, una curva puede obtenerse a partir de una recta mediante su flexión (curvatura) y retorcimiento (torsión). Así, para $|\kappa(s)| > 0$ y torsión $\tau(s)$, existe

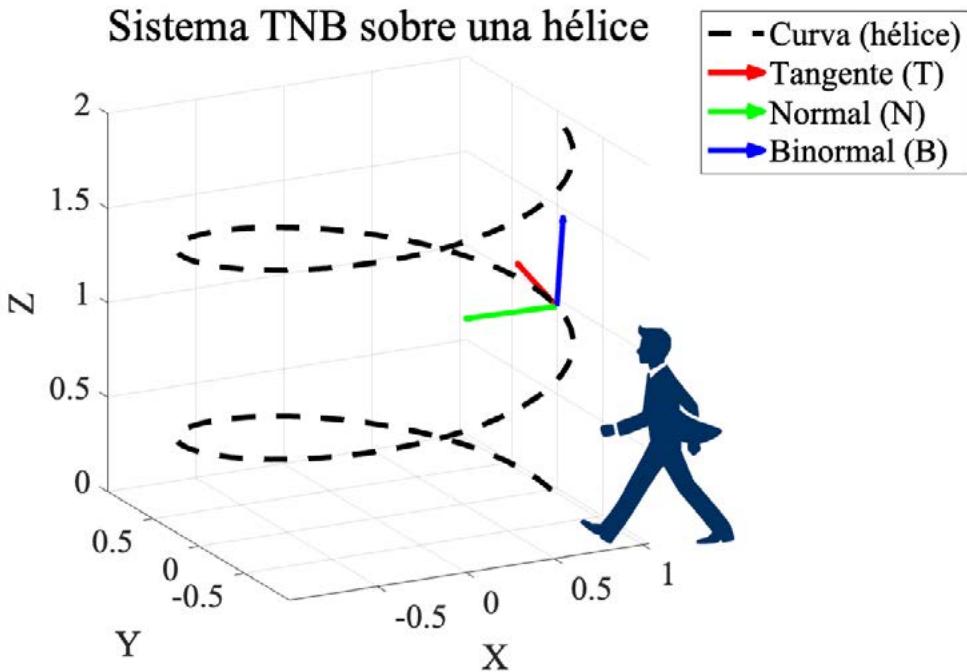


Figura 2.2: Diagrama de planificación y control.

una única curva espacial definida por las ecuaciones de Frenet-Serret, tal que:

$$\begin{bmatrix} \mathbf{T}'(s) \\ \mathbf{N}'(s) \\ \mathbf{B}'(s) \end{bmatrix} = \begin{bmatrix} 0 & \kappa(s) & 0 \\ -\kappa(s) & 0 & \tau(s) \\ 0 & -\tau(s) & 0 \end{bmatrix} \begin{bmatrix} \mathbf{T}(s) \\ \mathbf{N}(s) \\ \mathbf{B}(s) \end{bmatrix} \quad (2.1)$$

siendo, $\mathbf{T}(s)$, $\mathbf{N}(s)$ y $\mathbf{B}(s)$ los vectores tangente, normal y binormal que conforman un sistema ortogonal unitario de mano derecha, respectivamente. Mientras, $\mathbf{T}'(s) = d\mathbf{T}(s)/ds$, $\mathbf{N}'(s) = d\mathbf{N}(s)/ds$ y $\mathbf{B}'(s) = d\mathbf{B}(s)/ds$ son las primeras derivadas de tales vectores.

Este sistema ortogonal se representa como $\mathbf{R}(s) := [\mathbf{T}(s) \ \mathbf{N}(s) \ \mathbf{B}(s)]$ y puede integrarse a partir de la ecuación (2.1) a partir del par específico de funciones $\kappa(s)$ y $\tau(s)$, dado un valor inicial de $\mathbf{R}(0) := [\mathbf{T}(0) \ \mathbf{N}(0) \ \mathbf{B}(0)]$. Una vez determinado el campo de vectores tangentes, la posición de la curva puede obtenerse integrando dicho vector, como se indica en la ecuación :

$$\mathbf{C}(s) := \mathbf{C}(0) + \int_0^s \mathbf{T}(\xi) d\xi \quad (2.2)$$

Por otro lado, partiendo de una curva $\mathbf{C}(s)$, se puede obtener los vectores tangente $\mathbf{T}(s)$, normal $\mathbf{N}(s)$ y binormal $\mathbf{B}(s)$. Entonces, teniendo en consideración la teoría de la geometría diferencial de las curvas [58], se define las siguientes igualdades para el triángulo móvil de cualquier curva $\mathbf{C}(s)$:

$$\mathbf{T}(s) := \frac{\mathbf{C}'(s)}{\|\mathbf{C}'(s)\|} \quad (2.3)$$

$$\mathbf{N}(s) := \frac{\mathbf{T}'(s)}{\|\mathbf{T}'(s)\|} = \frac{[\mathbf{C}'(s) \times \mathbf{C}''(s)] \times \mathbf{C}'(s)}{\|[\mathbf{C}'(s) \times \mathbf{C}''(s)] \times \mathbf{C}'(s)\|} \quad (2.4)$$

$$\mathbf{B}(s) := \mathbf{T}(s) \times \mathbf{N}(s) = \frac{\mathbf{C}'(s) \times \mathbf{C}''(s)}{\|\mathbf{C}'(s) \times \mathbf{C}''(s)\|} \quad (2.5)$$

donde, $\mathbf{C}'(s) = d\mathbf{C}(s)/ds$, $\mathbf{C}''(s) = d^2\mathbf{C}(s)/ds^2$ y $\mathbf{C}'''(s) = d^3\mathbf{C}(s)/ds^3$, representan las derivadas del vector de posición $\mathbf{C}(s)$.

2.2.2 Clotoídes panares 2D

Una espiral de Euler o clotoíde plana se define en \mathbb{R}^2 como la curva cuya curvatura varía linealmente con respecto a la longitud de arco, tal que:

$$\kappa(s) := \kappa_0 + \sigma_\kappa s, \quad (2.6)$$

donde, κ_0 es la curvatura inicial y $\sigma_\kappa := d\kappa(s)/ds$ es la brusquedad de la clotoíde, que está relacionada con su factor de homotecia o escala K , debido a $\sigma_\kappa := \pi/K^2$, mientras el ángulo tangente de la clotoíde esta dado por:

$$\beta(s, \mathbf{p}) := \kappa_0 s + \frac{\sigma_\kappa}{2} s^2, \quad (2.7)$$

donde, $\mathbf{p} = \{\kappa_0, \sigma_\kappa\}$ es el vector de parámetros (para compactar la notación). Además de esto, el vector tangente, se puede expresar como:

$$\mathbf{T}(s, \mathbf{p}) := \begin{bmatrix} \cos(\beta(s, \mathbf{p})) \\ \sin(\beta(s, \mathbf{p})) \end{bmatrix} \quad (2.8)$$

A partir de las ecuaciones (2.2), (2.7) y (2.8), definidas para una clotoíde planar (con-

tenida en el plano XY) [59], denotadas como $\mathbf{C}(s, \mathbf{p})$, pueden derivarse de las integrales de Fresnel:

$$\mathbf{C}(s, \mathbf{p}) = \begin{bmatrix} \mathcal{C}(s, \mathbf{p}) \\ \mathcal{S}(s, \mathbf{p}) \end{bmatrix} := \begin{bmatrix} \int_0^s \cos(\kappa_0 \xi + \frac{\sigma_\kappa}{2} \xi^2) d\xi \\ \int_0^s \sin(\kappa_0 \xi + \frac{\sigma_\kappa}{2} \xi^2) d\xi \end{bmatrix} \quad (2.9)$$

siendo, $\mathcal{C}(s, \mathbf{p})$ y $\mathcal{S}(s, \mathbf{p})$ las integrales de Fresnel en coseno y seno. Es importante resaltar que, sin pérdida de generalidad, se asume que la clotoide está centrada en el origen, es decir, $\mathbf{C}(0) = \mathbf{0}$, para la ecuación (2.2).

Los autores de [60] introdujeron la espiral de Euler 3D o clotoide ($C3D$), definida en \mathbb{R}^3 , como una curva cuya curvatura varía como en la ecuación (2.6), mientras su torsión varía, tal que:

$$\tau(s) := \tau_0 + \sigma_\tau s \quad (2.10)$$

donde, τ_0 es la torsión inicial, y $\sigma_\tau := d\tau/ds$ es la primera derivada geométrica o brusquedad de torsión.

El principal inconveniente de las clotoídes 3D tal y como se presentan en [60], es que requiere una integración numérica para ser resuelta. En consecuencia, si se busca construir un planificador, a través de una curva 3D que alcance una posición u orientación determinada, los parámetros de la clotoide 3D deben optimizarse para que la clotoide satisfaga tales restricciones, lo que consume tiempo y no es adecuado para operaciones en tiempo real.

2.2.3 Curvas Bézier

Con el objetivo de facilitar la comprensión del concepto de curvas Bézier, este tipo de curvas se puede explicar como un conjunto de líneas suaves definidas por puntos de control que atraen la curva hacia ellos, creando formas curvas predecibles. La Figura 4 presenta un ejemplo. En ella, se puede apreciar que la curva (línea negra discontinua) inicia y finaliza en puntos específicos (puntos rojos), sin embargo, no incide en ninguno de los puntos intermedios (puntos verdes). Esto significa que la curva realiza una aproximación hacia los puntos de control.

Las curvas de Bézier constituyen un método matemático para representar curvas de

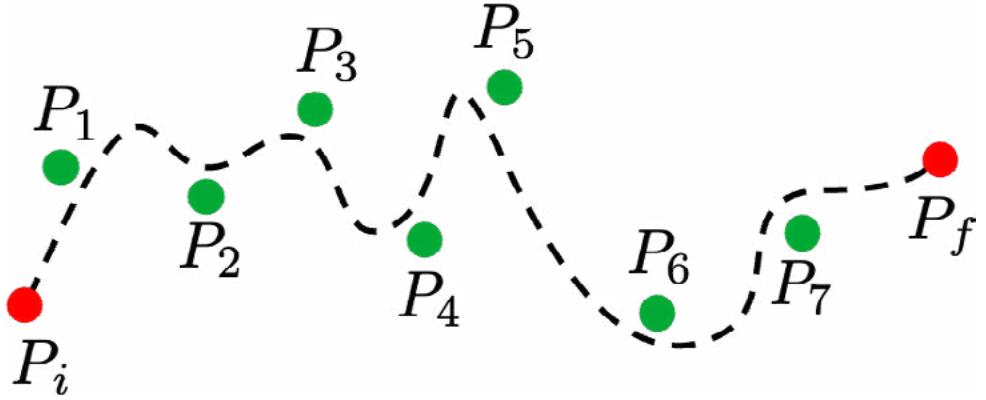


Figura 2.3: Forma clásica de la Curva de Bézier.

forma numéricamente estable [61, 62]. En este sentido, una curva de Bézier caracterizado por tener un solo componente polinómico. Los polinomios de Bernstein de grado n (con $n > 0$) también pueden interpretarse como funciones básicas B-splines de grado n definidas sobre el dominio de dichos polinomios.

La curva de Bézier de grado n puede expresarse de la siguiente manera:

$$\begin{aligned} \mathbf{B}_B(s, \mathbf{P}) &= \sum_{i=0}^n \binom{n}{i} \mathbf{P}_i \left(\frac{s_f - s}{s_f} \right)^{n-i} \left(\frac{s}{s_f} \right)^i \\ &= \mathbf{P}_0(1-t)^n + \binom{n}{1} t + \cdots + \mathbf{P}_n t^n, \end{aligned} \quad (2.11)$$

$t \in [0, 1]$

donde, $\mathbf{P} := [\mathbf{P}_0^T, \mathbf{P}_1^T, \dots, \mathbf{P}_n^T] \in \mathbb{R}^{3(n+1)}$ son el conjunto de puntos de control \mathbf{P}_i , s_f es la longitud de la curva y \mathbb{R}^3 es el espacio 3D en el que se define la curva. En otras palabras, el grado n de la curva de Bézier significa una interpolación entre los $n + 1$ puntos de control 3D. Por lo tanto, las curvas de Bézier pueden expresarse en términos de polinomios de Bernstein de grado n , tal que:

$$\mathbf{B}_B(s, \mathbf{P}) = \sum_{i=0}^n b_{i,n}(s) \mathbf{P}_i, \quad s \in [0, s_f] \quad (2.12)$$

siendo, los polinomios $b_{i,n}(s) \in \mathbb{R}$

$$b_{i,n}(s) := \binom{n}{i} \left(\frac{s_f - s}{s_f} \right)^{n-i} \left(\frac{s}{s_f} \right)^i, \quad i = 0, \dots, n \quad (2.13)$$

La principal ventaja de las curvas de Bézier con respecto a las clooides es que tienen propiedades de escalado y rotación [63]. Por lo tanto, se puede ajustar los \mathbf{P}_i de una Bézier unitaria y luego utilizarlos para generar una gran variedad de curvas debido a las propiedades mencionadas.

2.3 Control de seguimiento y Actitud

Una problemática de estudio importante, dentro de la tecnología UAV, tiene que ver con el control de seguimiento y actitud de vuelo [64]. Esto es especialmente relevante en los aviones (vehículos voladores de ala fija), porque poseen restricciones no-holonómicas y, por tanto, el control de trayectoria y actitud es la opción natural para pilotar un UAV.

2.3.1 Generación de trayectorias suaves

Los autores de [60] desarrollaron una curva suave para unir dos configuraciones punto-vector en el espacio 3D. El punto de partida son las ecuaciones de Frenet-Serret, como se describe en (2.1), y el objetivo es calcular una curva en el espacio como en la ecuación (2.2). De esta forma, se ha construido una curva que une dos configuraciones arbitrarias de puntos-vectores, por medio del algoritmo de optimización del descenso del gradiente [65]. El procedimiento obtiene los valores óptimos de la curvatura κ_0 y torsión τ_0 iniciales, así como la brusquedad de curvatura σ_κ y la brusquedad de torsión σ_τ . Por tanto, su solución podría tomar cualquier valor inicial y final de κ y τ . Sin embargo, cuando se planifican curvas para vehículos no-holonómicos, como los robots aéreos o subacuáticos que se mueven en un espacio 3D, debido a sus restricciones no-holonómicas, dichos vehículos no pueden seguir curvas arbitrarias con cambios instantáneos de κ, τ de orientación. En consecuencia, en muchos casos prácticos, los valores de la κ y τ iniciales son establecidos, basán-

dose en la κ y la τ actuales del robot. En ese caso, utilizando la solución propuesta en [60]] sólo se puede conseguir la posición final o la orientación final, pero no ambas simultáneamente.

En este capítulo se intenta construir una curva, parametrizada a través de un vector de parámetros genérico \mathbf{p} , denominado $\mathbf{C}(s, \mathbf{p})$. En realidad, el vector de parámetros, se divide en dos subconjuntos de parámetros $\mathbf{p} := [\mathbf{p}_0 \tilde{\mathbf{p}}]$, donde, $\mathbf{p}_0 = (\kappa_0, \tau_0)$ es una lista de parámetros que definen las condiciones iniciales; y $\tilde{\mathbf{p}} = (\sigma_\kappa, \sigma_\tau)$ es el parámetro de diseño. El objetivo es calcular $\tilde{\mathbf{p}}^*$ tal que $\mathbf{T}(s, \mathbf{p}) = \mathbf{T}^*$ para un $s > 0$ dado. Sin pérdida de generalidad se asume que el marco local de la curva es coincidente con el marco de coordenadas global, lo que significa que el punto inicial es $\mathbf{C}(0, \mathbf{p}) = [0, 0, 0]^T$.

Utilizando las propiedades de las clotoides [66], se ha proporcionado una solución inicial para el procedimiento de optimización. Entonces, el punto de partida es el cálculo individual de la brusquedad requerida por dos clotoides, en los planos XY y YZ . De esta forma, se obtiene σ_κ y σ_τ , por separado, para un ángulo de cabeceo $\theta(s)$ y un ángulo de guiñada $\psi(s)$, siendo:

$$\gamma_\tau = \sqrt{\frac{|\theta(s)|}{\pi/2}} \quad \gamma_\kappa = \sqrt{\frac{|\psi(s)|}{\pi/2}} \quad (2.14)$$

$$\sigma_{\kappa_0} = (\theta(s)) \frac{\pi \gamma_\kappa^2}{s^2} \quad (2.15)$$

$$\sigma_{\tau_0} = (\psi(s)) \frac{\pi \gamma_\tau^2}{s^2} \quad (2.16)$$

Por lo tanto, el problema a resolver se puede plantear como un problema de minimización de la orientación con la conjetura inicial $\tilde{\mathbf{p}} = (\sigma_{\kappa_0}, \sigma_{\tau_0})$:

$$\tilde{\mathbf{p}}^* = \arg_{\tilde{\mathbf{p}}} \min \| \mathbf{T}^* - \mathbf{T}(s, \mathbf{p}) \| \quad (2.17)$$

2.3.2 Aproximación a la Clothoide 3D

A continuación, se realiza la descripción del método propuesto para aproximar las clotoides 3D utilizando curvas racionales de Bézier (RB), con un número determinado de puntos. El método realiza una ampliación de la formulación proporcionada en [63], considerando pesos independientes en cada \mathbf{P}_i de una curva RB. De esta forma, las curvas RB se pueden expresar como:

$$\mathbf{B}_{RB}(s, \mathbf{P}, \mathbf{W}) := \left[\sum_{i=0}^n b_{i,n}(s) \mathbf{W}_i \right]^{-1} \sum_{i=0}^n b_{i,n}(s) \mathbf{W}_i \mathbf{P}_i, \quad (2.18)$$

donde, $\mathbf{W}_i \in \mathbb{R}^3$ es una matriz diagonal de pesos.

Los parámetros a aprender son la posición de los pesos \mathbf{P}_i y los \mathbf{W}_i . Dado que presentan una relación no lineal, se busca aprender estos parámetros por medio de un método de dos pasos. Primero se busca aprender la posición de los \mathbf{P}_i , asumiendo $\mathbf{W}_i = \mathbf{I}$, lo que corresponde a la expresión de una curva de Bézier convencional. Después, aprender los pesos de la curva RB, considerando que los \mathbf{P}_i aprende los pesos de la curva RB, considerando que los \mathbf{P}_i son fijos, con el objetivo de refinar la estimación, incluyendo una mayor flexibilidad para ajustarse a la curva clotoide 3D original. Entonces, para estimar la curva de Bézier convencional, se fija la posición del primer y último \mathbf{P}_i a los valores inicial y final de la curva, es decir, $\mathbf{P}_0 = \mathbf{C}(0, \mathbf{p})$ y $\mathbf{P}_n = \mathbf{C}(s_f, \mathbf{p})$, donde, s_f es la longitud total de la curva. Dado que se asume $\mathbf{W}_i = \mathbf{I}$, entonces, los puntos de la curva se pueden obtener mediante un ajuste por mínimos cuadrados (LS) $\mathbf{Y} = \mathbf{X}\mathbf{P}$, tal que:

$$\mathbf{Y} = \begin{bmatrix} \mathbf{C}(0, \mathbf{p}) - \frac{b_{0,n}(0)}{\sum_i^n b_{i,n}(0)} \mathbf{P}_0 - \frac{b_{n,n}(0)}{\sum_i^n b_{i,n}(0)} \mathbf{P}_n \\ \vdots \\ \mathbf{C}(s_f, \mathbf{p}) - \frac{b_{0,n}(1)}{\sum_i^n b_{i,n}(1)} \mathbf{P}_0 - \frac{b_{n,n}(1)}{\sum_i^n b_{i,n}(1)} \mathbf{P}_n \end{bmatrix} \quad (2.19)$$

$$\mathbf{X} = \begin{bmatrix} \frac{b_{1,n}(0)}{\sum_i^n b_{i,n}(0)} & \cdots & \frac{b_{n-1,n}(0)}{\sum_i^n b_{i,n}(0)} \\ \vdots & \ddots & \vdots \\ \frac{b_{1,n}(1)}{\sum_i^n b_{i,n}(1)} & \cdots & \frac{b_{n-1,n}(1)}{\sum_i^n b_{i,n}(1)} \end{bmatrix} \quad (2.20)$$

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_1 \\ \vdots \\ \mathbf{P}_{n-1} \end{bmatrix} \quad (2.21)$$

Con la posición estimada de los \mathbf{P}_i , es posible estimar los valores de los pesos para cada coordenada. Primero se expresa la igualdad $\mathbf{C}(s, \mathbf{p}) = \mathbf{B}_{RB}(s, \mathbf{P}, \mathbf{W})$ en términos de los pesos, siendo:

$$\sum_{i=0}^n b_{i,n}(s) \mathbf{C}(s, \mathbf{p}) \mathbf{W}_i = \sum_{i=0}^n b_{i,n}(s) \mathbf{P}_i \mathbf{W}_i, \quad (2.22)$$

considerando que $\mathbf{W}_0 = \mathbf{I}$ y $\mathbf{W}_n = \mathbf{L}$, entonces, con el objetivo de asegurar que la curva estimada comience y termine en la posición inicial y final de la curva original, se obtiene la ecuación lineal $\mathbf{\hat{Y}} = \mathbf{\hat{X}}\mathbf{\hat{W}}$, que se puede resolver mediante LS:

$$\mathbf{\hat{Y}} = \begin{bmatrix} b_{0,n}(0)\mathbf{\hat{P}}_0(0) + b_{n,n}(0)\mathbf{\hat{P}}_n(0) \\ \vdots \\ b_{0,n}(s_f)\mathbf{\hat{P}}_0(s_f) + b_{n,n}(s_f)\mathbf{\hat{P}}_n(s_f) \end{bmatrix} \quad (2.23)$$

$$\mathbf{\hat{X}} = \begin{bmatrix} -b_{1,n}(0)\mathbf{\hat{P}}_1(0) & \dots & b_{n-1,n}(0)\mathbf{\hat{P}}_{n-1}(0) \\ \vdots & \ddots & \vdots \\ -b_{1,n}(s_f)\mathbf{\hat{P}}_1(s_f) & \dots & b_{n-1,n}(s_f)\mathbf{\hat{P}}_{n-1}(s_f) \end{bmatrix} \quad (2.24)$$

$$\mathbf{\hat{W}} = [\mathbf{W}_1^T \ \dots \ \mathbf{W}_{n-1}^T]^T \quad (2.25)$$

donde, $\mathbf{\hat{P}}_i(s) := (\mathbf{C}(s, \mathbf{p}) - \mathbf{P}_i)$ y abusando de la notación $\mathbf{\hat{W}}$ es efectivamente una versión vectorizada de la misma, al resolver el problema LS.

2.4 Experimentos y Resultados

La experimentación se ha llevado a cabo a través de las herramientas de simulación de vuelo FlightGear 2018 y el entorno de desarrollo integrado Matlab R2017b. El modelo de avión utilizado para las simulaciones dinámicas se basa en el UAV

Kadett 2400 [67], como se representa en la Figura 5. El sistema tiene 4 entradas (véase la Figura 6) para controlar las superficies aerodinámicas y la velocidad. Siendo, δ_e (elevador), δ_a (alerón), δ_r (timón) y δ_{th} (acelerador). Como cualquier vehículo que se mueve en un espacio cartesiano 3D, puede representarse mediante 6 estados $\{x, y, z, \varphi, \theta, \psi\}$, donde los tres primeros estados definen el vector de posición del centro de gravedad C_G con respecto a un sistema de coordenadas global CS_g situado en el origen y los tres últimos son los ángulos de Euler de balanceo, cabeceo y guiñada, respectivamente, que definen la orientación del sistema de coordenadas local del cuerpo CS_b con respecto a CS_g .

Debe observarse que CS_b está definido por 3 vectores ortogonales unitarios $\{X_b, Y_b, Z_b\}$, alineados con los tres ejes del vehículo y centrados en C_G , con Z_b apuntando hacia abajo, como puede verse en la Figura 5. Las velocidades angulares a lo largo de los ejes X_b , Y_b y Z_b se representan por p , q y r , respectivamente, como se detalla en [67].

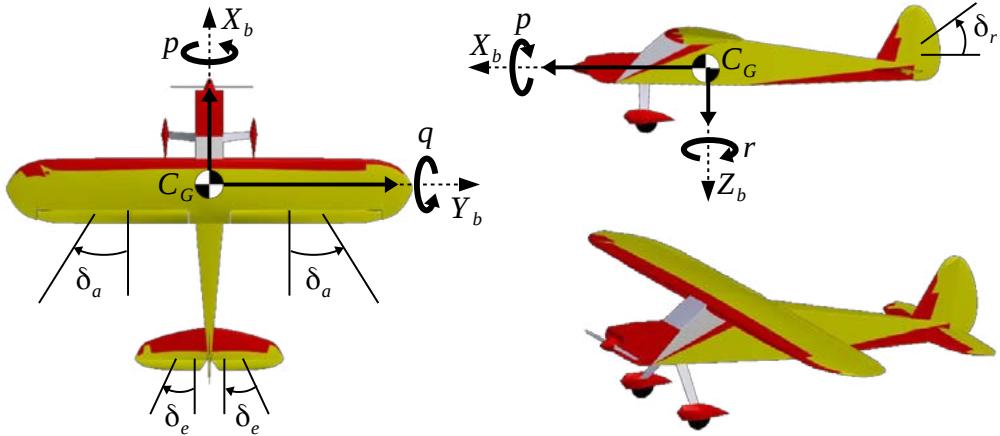


Figura 2.4: Definición del modelo UAV Kadett y sus variables.

La Figura 6 muestra la estructura de control utilizada para la aeronave. Dadas unas condiciones iniciales (vector tangente actual T) y una configuración objetivo (vector tangente final T^*), \square , se genera una trayectoria de referencia. El objetivo es mantener una velocidad constante vel a lo largo de la trayectoria, lo que se consigue mediante un controlador PID que afecta al acelerador. Además, los perfiles θ y ψ de la trayectoria generada se utilizan como entrada de referencia para otros dos controladores PID. El primer PID controla el elevador, que afecta a la orientación del cabeceo. Mientras que, la salida del segundo PID se multiplica por una ganancia K para controlar el alerón, que afecta a la orientación del alabeo, pero también hace que el avión gire y, por tanto, afecta a la orientación de la guiñada. La acción de control también se aplica al timón, lo que afecta ligeramente a la orientación de la guiñada y se utiliza para compensar pequeños errores al seguir la guiñada de referencia ψ .

Como se explica en la sección 2.3.2, una clooide 3D (*C3D*) puede ser aproximada por una curva RB. El principal inconveniente de utilizar *C3D* es que estas curvas se calculan utilizando las integrales de Fresnel, lo que, para un paso de discretización pequeño y distancias largas (límite superior de integración), significa un alto consumo de tiempo computacional. Además, otro problema es que el resultado final depende de la selección del paso de discretización. Así, para idénticos valores de σ_κ y σ_τ , la trayectoria generada y la orientación final serán diferentes dependiendo de dicho paso de integración.

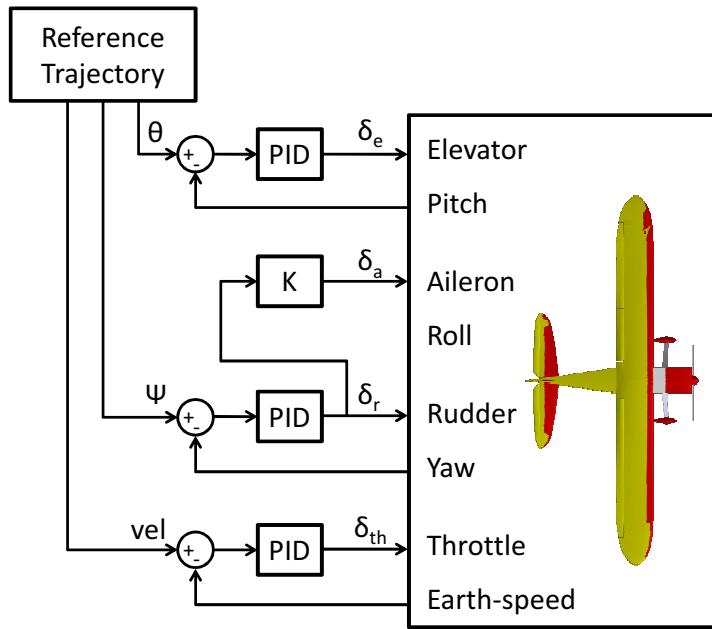


Figura 2.5: Esquema utilizado para el control del modelo de UAV Kadett.

Por el contrario, una RB que está parametrizada por n Puntos de control y pesos puede, en consecuencia, ser discretizada utilizando cualquier paso, produciendo siempre la misma curva. Otro inconveniente de utilizar directamente las curvas *C3D* es que, para resolver el problema de planificación, se utiliza algoritmos de optimización, con el objetivo de encontrar los valores $\sigma_{\kappa_0}^*$ y $\sigma_{\tau_0}^*$ y, a partir de ello, alcanzar un determinado vector tangente final T^* , para un determinado paso de integración. Este proceso consume mucho tiempo y no es apropiado para la planificación y el control en tiempo real de los UAV.

Para las simulaciones realizadas en esta sección, se ha aproximado una *C3D*, a través de una RB de orden $n = 13$. Tras un estudio en el que se utilizó diferentes órdenes ($n = \{5, 7, 9, 11, 13, 15, 17\}$), que no se muestra en favor de la brevedad, se encontró que $n = 13$ fue la mejor opción teniendo en cuenta la precisión y el tiempo de cálculo. Para este caso particular, se ha generado una *C3D* utilizando 10^4 puntos discretos. La

brusquedad de la curvatura se fija en $\sigma_\kappa = 5,54 \cdot 10^{-5} [\text{rad/m}^2]$, la brusquedad de la torsión es $\sigma_\tau = 6,12 \cdot 10^{-5} [\text{rad/m}^2]$, la longitud total es $l = 180 \text{ m}$, con ángulos finales $\theta = -\pi/8 \text{ rad}$ y $\psi = \pi/4 \text{ rad}$. El tiempo medio de cálculo para este caso particular es $t_c = 15,97 [\text{ms}]$, mientras que el error de aproximación integrado a lo largo de la trayectoria es $e = 5,84 \cdot 10^{-6} [\text{m}]$.

Para este análisis se ha utilizado un ordenador con procesador Intel Core i7-6700HQ 2,60GHz y memoria DDR4 de 16GiB. La Figura 7 muestra la C3D (azul) y la RB aproximada (rojo discontinuo), así como los \mathbf{P}_i (círculos verdes).

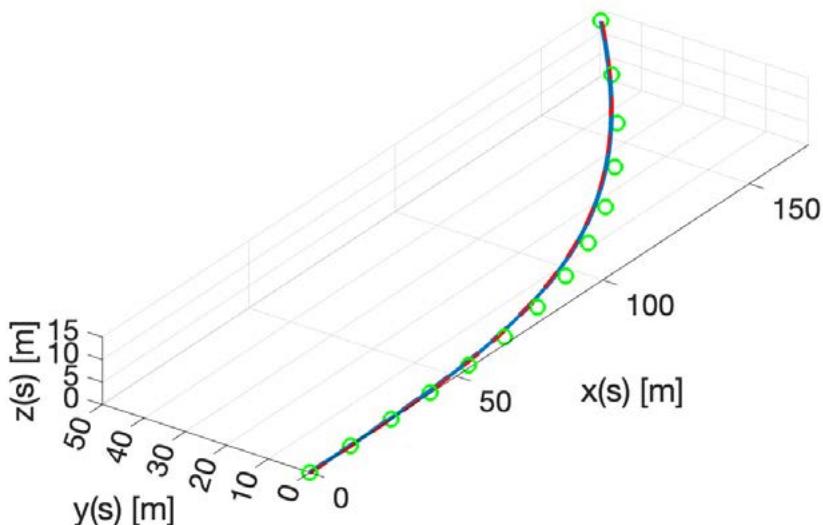


Figura 2.6: Aproximación de C3D con RBC: orientación final: $\theta^* = -\pi/8$ y $\psi^* = \pi/4$.

Para los experimentos se ha establecido que, la velocidad de referencia es constante en $vel = 18 [\text{m/s}]$, con un período de muestreo de $T_s = 20 [\text{ms}]$ para controlar la aeronave. Dado que la longitud de la trayectoria de referencia se ha fijado en $l = 180 [\text{m}]$, entonces, el tiempo de simulación es de $t = 10 [\text{s}]$.

Con el objetivo de realizar una descripción gráfica del seguimiento de la trayectoria por parte del UAV, en la Figura 8 se puede apreciar que el UAV (rojo) sigue muy de cerca la trayectoria de la referencia (azul), incluso cuando el UAV solo realiza el seguimiento de las referencias de cabeceo y guiñada.

Por otro lado, la Figura 9 muestra los errores de seguimiento para algunas variables. Se puede observar que el error de seguimiento de los ángulos de cabeceo y guiñada es lento. Esto se debe, principalmente, a que el RB es una aproximación de una C3D, que posee la propiedad de suavidad que permite un seguimiento fácil. En cuanto a la velocidad, el error es muy bajo a lo largo de casi toda la trayectoria. Sin embargo, en la última parte (alrededor de $l = 180 [\text{m}]$), debido a que el ángulo de cabeceo es alto,

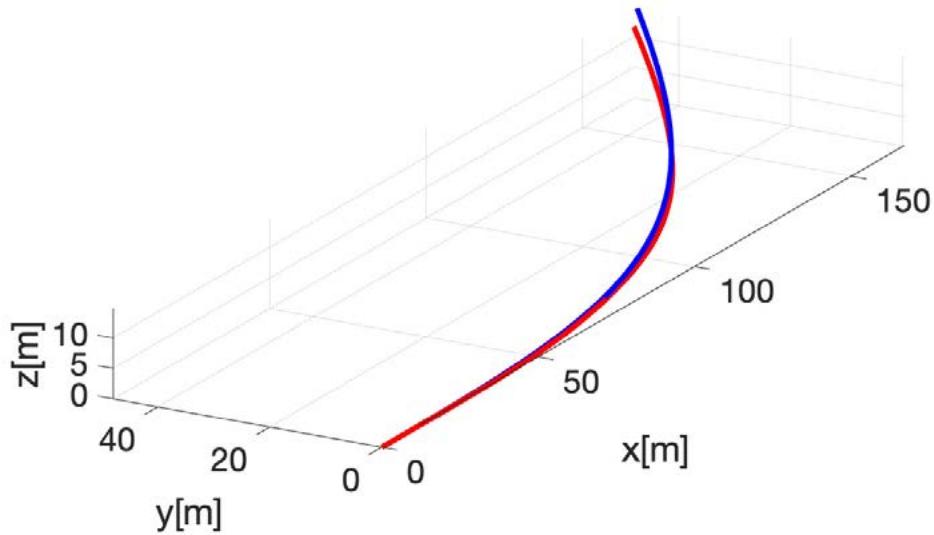


Figura 2.7: Trayectoria trazada por el UAV (rojo discontinuo) siguiendo una $C3D$ aproximada por un RB (azul) con una longitud $l = 180$ m y orientación final $\theta^* = -\pi/8$ rad y $\psi^* = \pi/4$ rad.

el UAV tiene algunas dificultades para seguir la velocidad de referencia y el error e_v aumenta. Como consecuencia, la distancia e_d entre el vehículo y la trayectoria también aumenta considerablemente en la segunda mitad del experimento.

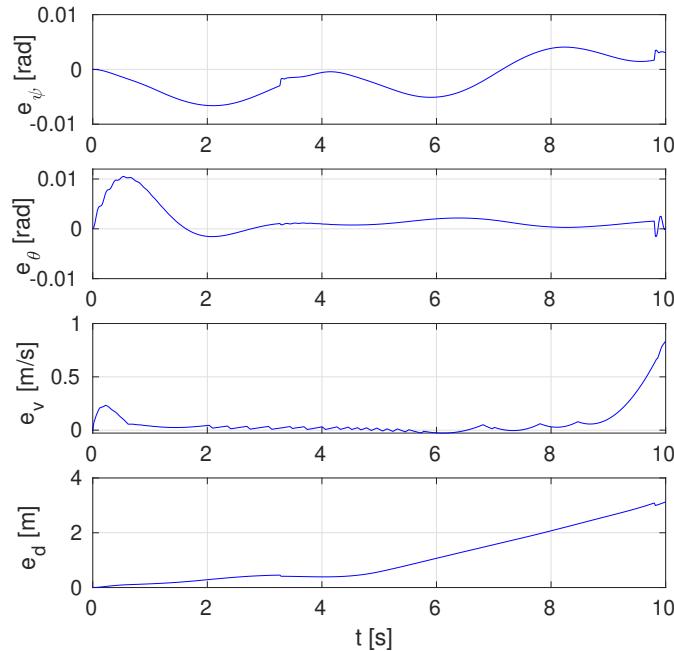


Figura 2.8: Error de seguimiento de los ángulos de guiñada y cabeceo, velocidad y distancia de posición a la trayectoria de referencia.

Finalmente, la Figura 10 presenta una secuencia de capturas de pantalla de la simulación en diferentes instantes de tiempo $t = \{0, 2, 4, 6, 8, 10\}$ [s]. Se puede observar que el UAV comienza en $t = 0$ [s] con altura $h = 0$, alabeo $\varphi = 0$, cabeceo $\theta = 0$ y

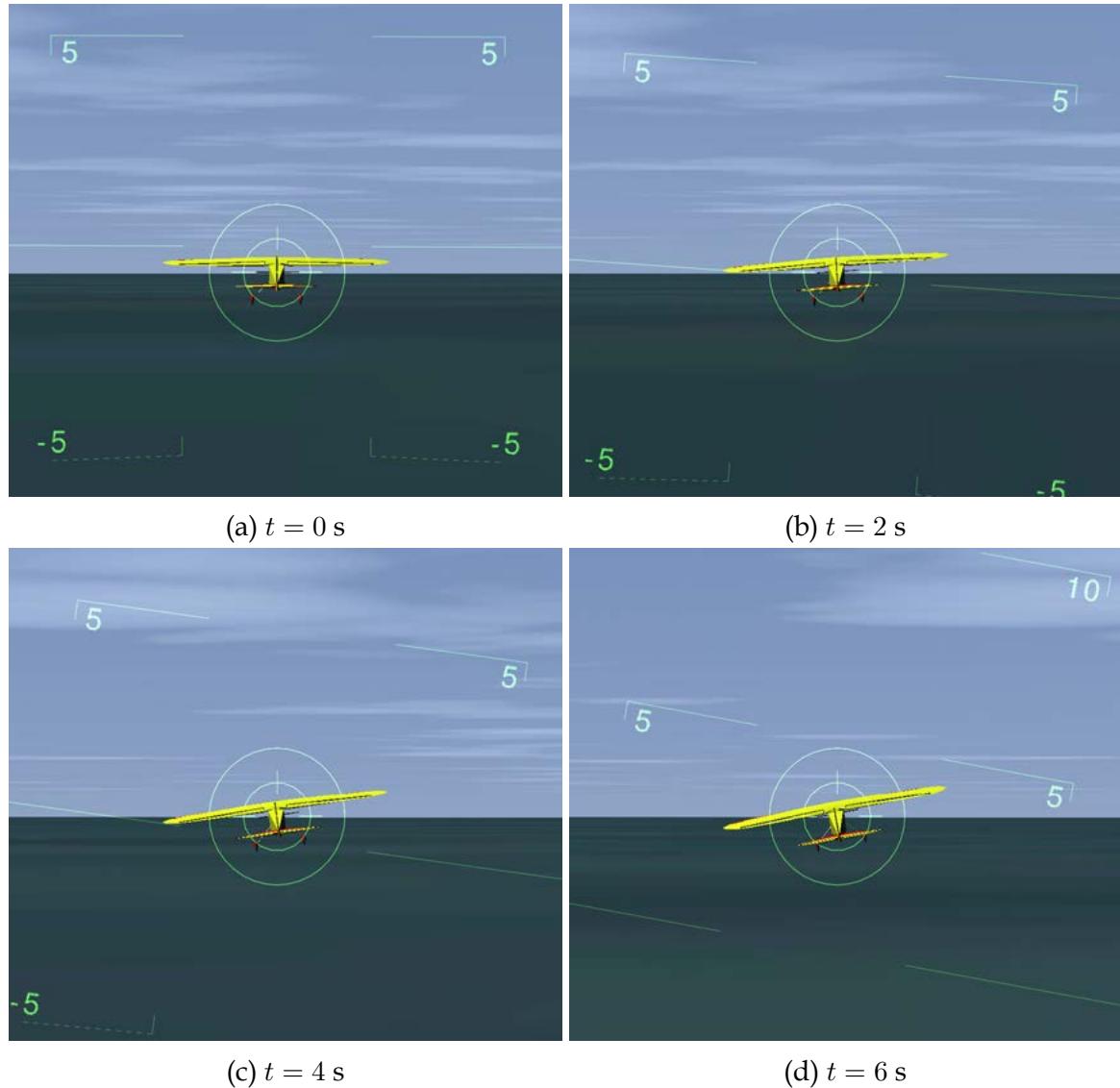


Figura 2.9: Capturas de pantalla de la simulación en FlightGear.

guiñada $\psi = 0$. A continuación, comienza el giro en alabeo, lo que hace que el avión gire hacia la izquierda, mientras aumenta progresivamente su ángulo de cabeceo y, en consecuencia su altura ($t = \{2 - 8\}\text{s}$). La última imagen muestra el UAV en la última configuración $t = 10\text{s}$, con cabeceo $\theta = -\pi/8 \text{ rad}$ y guiñada $\psi = \pi/4 \text{ rad}$.

2.5 Conclusiones

Se ha presentado un planificador de trayectorias suaves para aviones de ala fija. El esquema de control se divide en dos etapas: planificador local y control cinemático. En la etapa de planificación, las clotoides 3D se aproximan mediante curvas racionales de Bézier para generar trayectorias suaves en el espacio euclíadiano 3D. A continuación, con el objetivo de seguir las referencias de velocidad y orientación, se utiliza la trayectoria como referencia para pilotar el UAV de forma autónoma.

El uso de RBC de orden 13 para aproximar clotoides 3D ha demostrado ser eficiente computacionalmente (alrededor de 16 ms para calcular 10^4 puntos discretos) y con alta precisión (error de aproximación de alrededor de $e = 6 \cdot 10^{-6}$ s a lo largo de una trayectoria con longitud 180s).

Se ha realizado pruebas de simulación de vuelo utilizando el simulador FlightGear y MATLAB. Tras analizar los resultados, se observa algunas ventajas interesantes en el uso de clotoides 3D para la planificación de trayectorias. Un vehículo no-holonómico que sigue estas trayectorias suaves es capaz de seguir fácilmente la referencia de los ángulos de cabeceo y guiñada. En este sentido, aunque no se tenga en cuenta la referencia de posición, los resultados han demostrado que un UAV no-holonómico es capaz de trazar una trayectoria cercana a la diseñada por el planificador de trayectorias.

03

Planificación de trayectorias 3D limitadas por restricciones de maniobrabilidad en UAVs

RESUMEN

Este capítulo presenta una propuesta para la generación de trayectorias suaves tridimensionales 3D, navegables por vehículos aéreos no tripulados de características no-holonómicas (es decir, aviones de ala fija), a partir de la construcción de curvas continuas. El estudio realiza una breve inclusión de maniobrabilidad real de UAV de ala fija, donde el comportamiento dinámico y cinemático introduce restricciones en entornos reales. Finalmente, se incluye resultados de simulación de vuelo para clarificar la teoría presentada.

3.1 Introducción

Un reto fundamental para los algoritmos de planificación de vuelo consiste en incorporar estrategias activas de evitación de obstáculos, que garanticen una navegación continua, fluida y segura. Este reto implica la consideración de diversos aspectos, como las aceleraciones tangenciales, las derivadas parciales en sus componentes dimensionales, la odometría y el ruido [68, 69]. Actualmente, este campo de trabajo evidencia una actividad de investigación activa y relevante.

La literatura sobre planificadores de trayectoria y evasión de obstáculos, que además garanticen un seguimiento fluido y seguro, ya sea en entornos continuos o discretos, es amplia. Por este motivo, para el desarrollo de este capítulo se ha seleccionado dos de los enfoques más relevantes de la literatura para su comparación. Por un lado, se analiza la metodología rapidly-exploring random tree (RRT), que realiza el mapeo en el espacio continuo [70, 71, 72]. Por otro lado, se considera la metodología modified adaptive cell decomposition (MACD) [73] basado en técnicas de discretización del espacio [74, 75].

El enfoque general de las principales metodologías mencionadas consiste en describir la trayectoria óptima como un conjunto de puntos discretos. Ahora bien, este conjunto de puntos multidimensionales no resulta adecuado para un vehículo aéreo no tripulado (UAV) con características no-holonómicas (es decir, UAV de ala fija) [76]. Es importante destacar que los UAV con esta estructura particular realizan un vuelo continuo a una velocidad definida. Por lo tanto, sus movimientos están limitados por su capacidad de maniobrabilidad. En este sentido, cualquier trayectoria de vuelo calculada debe satisfacer las restricciones propias del modelo cinemático del UAV. Un enfoque interesante de estudio aborda la búsqueda de trayectorias que no solo rodeen el obstáculo por sus lados, sino que también lo evadan por arriba o por debajo. Con el objetivo para explotar todas las capacidades de maniobrabilidad de los UAV, este capítulo presenta un enfoque general de trabajo en el espacio euclíadiano 3D.

En resumen, el capítulo introduce un método para la creación de trayectorias suaves 3D, a partir de la construcción de curvas compuestas por segmentos de líneas rectas y segmentos curvos. Para la construcción de estas curvas se ha considerado las ca-

racterísticas cinemáticas del UAV de ala fija [77]. Finalmente, se presenta una breve transición completa entre la curva y la trayectoria navegable por un UAV.

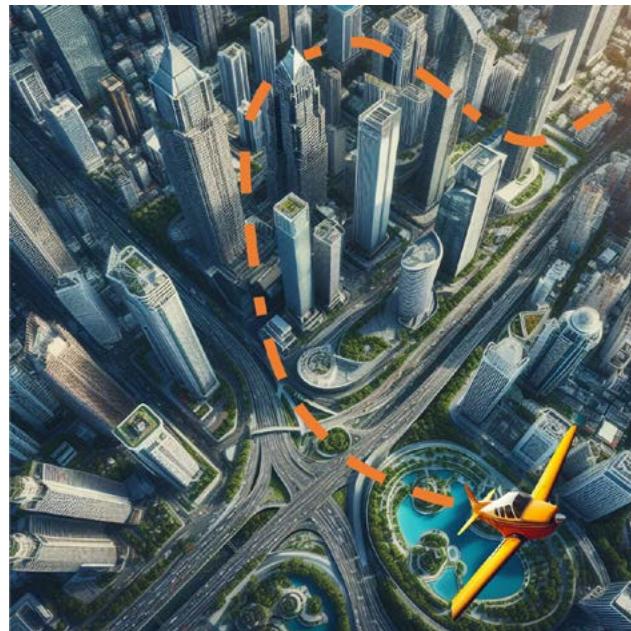


Figura 3.1: Descripción del problema de planificación de trayectoria suave 3D para un UAV de ala fija.

El capítulo se organiza de la siguiente forma: en la sección 3.2, se define el modelo UAV, además del enfoque de curvatura; en la sección 3.3 se presenta un nuevo enfoque para la construcción de trayectorias suaves; en la sección 3.4, se analiza los resultados y datos estadísticos obtenidos, a través de simulaciones de vuelo; por último, se señala las principales conclusiones en la sección 3.5.

3.2 Descripción del modelo UAV

Antes de realizar una definición matemática formal de un UAV, es importante entender de forma sencilla su concepto. Más allá del significado de sus siglas, que ya fueron definidas en el capítulo introductorio, es importante destacar que un UAV es una aeronave que puede volar sin piloto, tripulación ni pasajeros a bordo. Se controla de forma remota o funciona de forma autónoma, a través de una computadora a bordo, que le proporciona diferentes habilidades, entre las que se destaca la posibilidad de seguimiento de rutas programadas o respondiendo a sensores. Finalmente, un UAV integra a un operador humano que controla el UAV desde una estación de control terrestre.

3.2.1 Sistema no-holonómico UAV

Aunque los conceptos de sistemas holonómicos y no-holonómicos parezcan complicados, en realidad no lo son. En referencia directa a los vehículos móviles, un sistema holonómico se define como la capacidad de un vehículo de realizar giros bruscos, como un vehículo terrestre omnidireccional (ODV) que puede girar sobre sí mismo, como lo haría un helicóptero al realizar maniobras similares en el aire. En cambio, el vehículo no-holonómico presenta una restricción de movimiento, pues no puede moverse en cualquier dirección sin realizar un desplazamiento, así como sucede con un automóvil que, para girar a un lado, necesita estar en desplazamiento. Otro ejemplo es el avión de alas, que requiere estar en vuelo para poder realizar un giro en cualquier dirección.

Entonces, se puede asumir a \mathfrak{B} como un UAV no-holonómico con masa \mathfrak{M} e inercia \mathfrak{J} , que se desplaza en el espacio euclíadiano \mathbb{R}^3 , y cuyo espacio de estados debido al desplazamiento se define de la siguiente manera:

$$\begin{aligned} q_R(t) = & (X_R(t), Y_R(t), Z_R(t), \\ & F_R(t), G_R(t), V_R(t), W_R(t))^T \in \mathbb{R}^7 \end{aligned} \quad (3.1)$$

donde, $X_R(t)$, $Y_R(t)$ y $Z_R(t)$ representan el sistema de conversión de Tait-Bryan [78, 79] con origen común. En este caso, se puede eliminar la dependencia del tiempo y asumir condiciones ideales de vuelo estacionario para el UAV. Además de la fuerza total $F_R(t) = 0$, el momento total $G_R(t) = 0$, la velocidad lineal y la velocidad angular son constantes $V_R(t) = W_R(t) = k$. Entonces, se puede definir una configuración reducida de la siguiente manera:

$$\dot{q}_R = [\phi_R, \theta_R, \psi_R]^T \quad (3.2)$$

donde, $\psi_R(t)$ es el ángulo de guiñada, $\theta(t)$ es el ángulo de cabeceo y $\phi_R(t)$ es el ángulo de alabeo del UAV.

Si se asume las refracciones en un modelo geométrico de aproximación a la tierra plana [80], entonces se puede considerar que el sistema local del UAV permanece

alineado con un sistema cartesiano objetivo $q_T = [X_T, Y_T, Z_T]^T$. En este caso, las ecuaciones del sistema coinciden con los puntos de cada modelo. Por lo tanto, existe una conexión en todo momento entre los puntos generados por la posición del UAV \dot{q}_R y otra configuración derivada equivalente q_T .

Por otra parte, la relación de las coordenadas cartesianas sobre conjuntos abiertos $U = \{(r, \theta, \varphi) | r < 0, 0 < \theta < \pi, 0 \leq \varphi < 2\pi\}$ y $V = \{(x, y, z) | x^2 + y^2 + z^2 > 0\}$, muestra una correspondencia unívoca $F' : V \rightarrow U$ entre las coordenadas cartesianas y esféricas, que alcanzan singularidad al extenderse al eje z , con $x^2 + y^2 = 0$, donde, φ no está definido. Además, φ no es continua en (x, y, z) , con $x = 0$.

Por lo tanto, la función inversa $F^{-1}1$ entre los mismos conjuntos abiertos se puede describir los siguientes términos:

$$\begin{aligned} x &= r \sin \theta \cos \varphi \\ y &= r \sin \theta \sin \varphi \\ z &= r \cos \theta \end{aligned} \tag{3.3}$$

donde, la matriz jacobiana se define como:

$$\|J\| = r^2 \sin \theta \tag{3.4}$$

3.2.2 Radio de Curvatura

La magnitud que mide el cambio de dirección de la curva del vector tangente (definida como curvatura), frente a un objeto diferenciable incrustado en el espacio euclídeo [81], se define a partir de la ecuación paramétrica de la circunferencia $g : \mathbb{R} \rightarrow \mathbb{R}^n$, que asume el mismo valor de λ y satisface $g'(t) = \lambda'(t)$, siendo $g''(t) = \lambda''(t)$ en cada t fijo. Entonces, el radio no depende de la posición de $\lambda(t)$, sino únicamente de la velocidad $\lambda'(t)$ y de la aceleración $\lambda''(t)$. Por tanto, la ecuación paramétrica $\lambda(t) : \mathbb{R} \rightarrow \mathbb{R}^n$ define la curvatura ρ en dependencia de los escalares $|\lambda'(t)|^2$, $|\lambda'(t)|^2$ y $\lambda' \cdot \lambda'(t)$

Se parte de la ecuación paramétrica general de la circunferencia $g(u) = A \cos(h(u)) + B \sin(h(u)) + C$, donde, $C \in \mathbb{R}^n$ es el centro, $A, B \in \mathbb{R}$ son vectores perpendiculares de módulo ρ . Entonces, $A \cdot A = B \cdot B = \rho^2 \wedge A \cdot B = 0$, $h : \mathbb{R} \rightarrow \mathbb{R}$, es una función do-

blemente diferenciable en t . Por lo tanto, el sistema de derivadas se puede describir de la siguiente manera:

$$\begin{aligned}|g'|^2 &= \rho^2(h')^2 \\ g' \cdot g'' &= \rho^2 h' h'' \\ |g''|^2 &= \rho^2((h')^4) + (h'')^2\end{aligned}\tag{3.5}$$

donde, el sistema derivado en relación con λ se define como:

$$\begin{aligned}|\lambda'^2(t)| &= \rho^2 h'^2(t) \\ \rho'(t) \cdot \lambda''(t) &= \rho^2 h'(t) h''(t) \\ |\lambda''^2(t)| &= \rho(h'^4(t) + h''^2(t))\end{aligned}\tag{3.6}$$

Entonces, el sistema resultante en $\rho, h'(t)$ y $h''(t)$ se define como:

$$\rho = \frac{|\lambda|^3}{\sqrt{|\lambda'|^2 |\lambda''|^2 - (\lambda' \cdot \lambda'')^2}}\tag{3.7}$$

donde, ρ es la magnitud geométrica del radio de curvatura.

3.3 Definición de curvas

Diversos planificadores de trayectorias [82], devuelven un conjunto de vértices y nodos en su resultado, que se definen como puntos de control \mathbf{P}_i . Por lo tanto, la trayectoria 3D es definida a partir de un conjunto \mathbf{P}_i , establecido en un orden específico, y expresarsado como una secuencia de interpolación discreta.

$$\mathbf{P}_i = v_i = f(t_i) \rightarrow \mathbb{R}\tag{3.8}$$

donde, $f(t_i)$ es un conjunto de puntos 3D e $i = 1, \dots, n$ (siendo, n el número total de puntos) es un conjunto de nodos conocidos a partir de la planificación de trayectoria. Por lo tanto, se puede definir un conjunto de k sub-intervalos entre $i = 1$ e $i = n$

particionado en $[a, b]$, siendo:

$$\begin{aligned} [a, b] &= [t_1, t_2] \cup [t_2, t_3] \cup \dots \cup [t_{n-2}, t_{n-1}] \cup [t_{n-1}, t_n] \\ a &= t_1 \leq t_2 \leq \dots \leq t_{n-1} \leq t_n = b \end{aligned} \quad (3.9)$$

Por tanto, se puede expresar una curva $s(t)$ con n polinomios a trozos, tal que:

$$s(t) = \begin{cases} s_1(t), & t \in [t_0, t_1] \\ s_2(t), & t \in [t_1, t_2] \\ \vdots \\ s_n(t), & t \in [t_{n-1}, t_n] \end{cases}$$

$$\begin{aligned} s_j(t) &= q_k t^k + q_{k-1} t^{k-1} + \dots + q_1 t^1 + q_0 t^0 \\ j &= 1, 2, \dots, n \end{aligned} \quad (3.10)$$

donde, q_k , representa coeficientes constantes, k es el grado del polinomio $s_j(t)$. Entonces, $s(t)$ representa la función de interpolación spline [77, 83] de grado k para la secuencia discreta $\mathbf{P}_i = v_i = f(t_i)$.

Por otra parte, dentro de la interpolación polinómica, un efecto común es el conocido fenómeno de oscilación de Runge [84]. En este sentido, las funciones de interpolación por curvas splines [77] minimizan la rugosidad sometida a restricciones, además, de su extrapolación en varias dimensiones.

3.3.1 Algoritmo de De Boor

El algoritmo De Boor es un método utilizado para evaluar curvas B-spline que, además, son una generalización de las curvas de Bézier. En realidad, realizan un promedio ponderado que se ajusta paso a paso, acercándose al punto correcto en la curva..

El algoritmo de De Boor [85] ofrece una estabilidad numérica para evaluar curvas spline $s(x)$ en el punto x . De Boor se construye a partir de una suma de funciones B-spline $B_{i,p}(x)$ multiplicadas por los \mathbf{P}_i . Las B-splines de orden $p+1$, son funciones

polinómicas de orden p conectadas a trozos y definidas sobre una malla de nodos $t_0, \dots, t_i, \dots, t_m$. Por otro lado, es importante resaltar que el algoritmo de De Boor utiliza $O(p^2) + O(p)$ operaciones [86] para evaluar la curva spline en forma B-spline.

$$s(x) = \sum_i c_i B_{i,p}(x) \quad (3.11)$$

Los polinomios B-spline son positivos en un dominio finito y nulos en el resto de dominios, de ahí que:

$$B_{i,0} := \begin{cases} 1 & \text{if } t_i \leq x \leq t_{i+1} \\ 0 & \text{de lo contrario} \end{cases} \quad (3.12)$$

$$\begin{aligned} B_{i,p}(x) = & \frac{x - t_i}{t_{i+p} - t_i} B_{i,p-1}(x) + \\ & \frac{t_{i+p+1} - x}{t_{i+p+1} - t_i} B_{i+1,p-1}(x) \end{aligned} \quad (3.13)$$

El algoritmo no calcula las funciones B-spline $B_{i,p}(x)$ directamente. En su lugar, evalúa $s(x)$ a través de una ecuación iterativa equivalente. Entonces, se define un conjunto de \mathbf{P}_i como $d_{i,r}$ con $d_{i,0}$ para $i = k-p, \dots, k$, y para $r = 1, \dots, p$ que se aplica en la siguiente ecuación:

$$\begin{aligned} d_{i,r} = & (1 - \alpha_{i,r})d_{i-1,r-1} + \alpha_{i,r}d_{i,r-1}; \\ i = & k-p+r, \dots, k \\ \alpha_{i,r} = & \frac{x - t_i}{t_{i+1+p-r} - t_i} \end{aligned} \quad (3.14)$$

Las iteraciones se completan cuando $s(x) = d_{k,p}$, lo que significa que $d_{k,p}$ es el resultado deseado. Esto permite evaluar diferentes grados de splines.

La Figura 12 muestra un ejemplo de ejecución de curvas spline con diferente orden. Para este ejemplo $\mathbf{P}_i = 4$ (denotados como puntos azules), de modo que la curva se construye desde p_1 hasta p_4 . Entonces, la trayectoria final se ve en la línea roja, mientras que las cajas verdes denotan los obstáculos. Las spline calculan la trayectoria suave a partir de los \mathbf{P}_i ; sin embargo, en ningún momento se considera la restricción

de radio curvatura, como se describe en la ecuación (3.7).

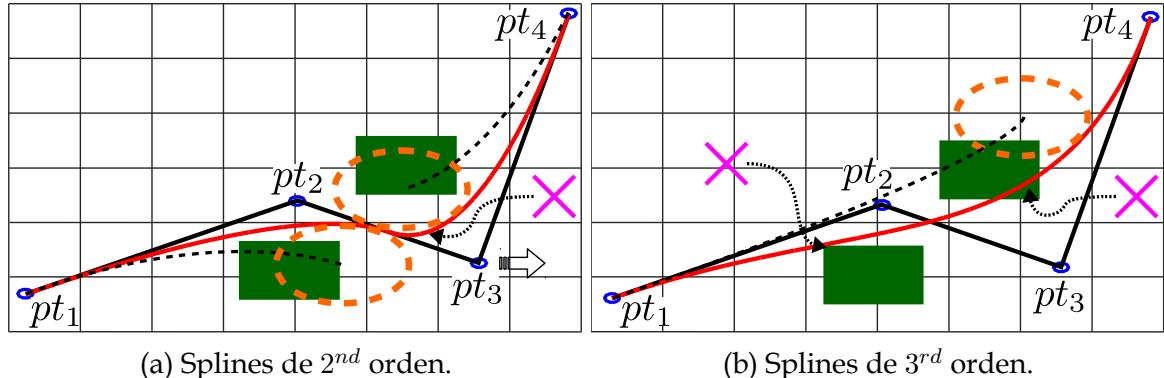


Figura 3.2: Curva spline construida con 4 puntos de control.

La Figura 12a muestra una spline de 2nd orden, donde, ρ se muestra fuera de sus límites (cruz magenta). Este problema se puede resolver moviendo los P_i , con el objetivo de mantener una trayectoria libre de colisiones. En [87] el problema se resuelve añadiendo P_i (es decir, nuevos puntos de control) en zonas críticas entre (p_1, p_2) , (p_2, p_3) y (p_3, p_4) .

Por otro lado, la Figura 12b muestra un ejemplo de curva spline de orden 3rd, donde, se satisfacen los límites ρ . No obstante, la trayectoria cambia y se producen 2 colisiones (cruz magenta).

Con el objetivo de evitar estas posibles colisiones, se propone una solución que se enfoca en la modificación de la curvatura en los puntos críticos, es decir, en los lugares cercanos de intersección de las líneas rectas que unen pares de P_i . Este enfoque considera las limitaciones de movimiento del UAV y propone una variación de la curva 3D a partir de semicircunferencias.

3.3.2 Aproximación de curvatura con semicircunferencias

Si se considera $P_i = nP$, entonces, existe un conjunto de ecuaciones de una recta $nL = nP - 1$, y un conjunto de ángulos entre las rectas $n\theta = nL - 1$. Por lo tanto,

cada ecuación de recta rL se determina de la siguiente manera:

$$rL_{i=2}^{nL} = \begin{pmatrix} x & y & 1 \\ P_{(i)}x & P_{(i)}y & 1 \\ P_{(i-1)}x & P_{(i-1)}y & 1 \end{pmatrix} \quad (3.15)$$

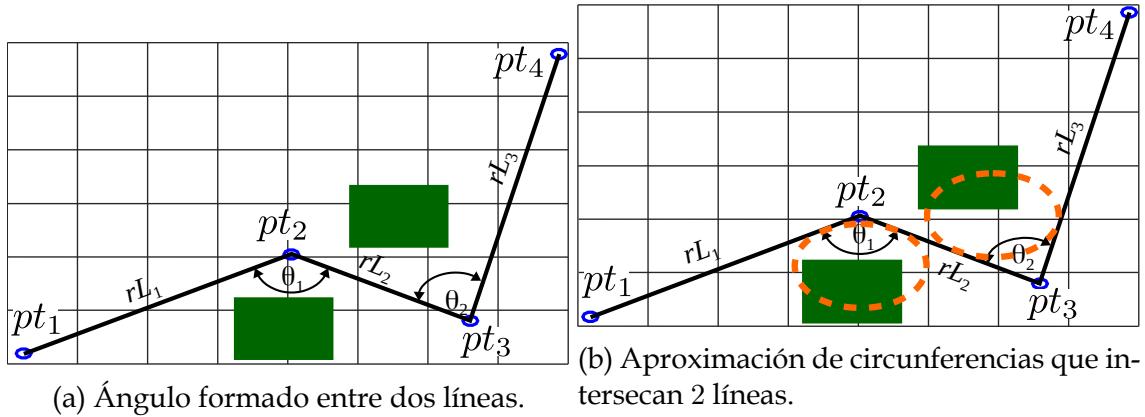
mientras, que el conjunto de ángulos θ se define como:

$$\begin{aligned} \tan \theta_{i=2}^{n\theta} &= \left| \frac{m_{(i)} - m_{(i-1)}}{1 + m_{(i)} \cdot m_{(i-1)}} \right| \\ m_{i=2}^{n\theta} &= \frac{P_{(i)}y - P_{(i-1)}y}{P_{(i)}x - P_{(i-1)}x} \end{aligned} \quad (3.16)$$

El objetivo es encontrar una forma para localizar una ecuación de circunferencia cE , definida entre cada par de líneas, siendo:

$$cE_{i=1}^{nL-1} = (x - a)^2 + (y - b)^2 - r^2 \quad (3.17)$$

manteniendo los P_i de la planificación de la trayectoria y añadiendo una curva suave con límites ρ entre los puntos críticos, como se describe en la Figura 13.



(a) Ángulo formado entre dos líneas.

(b) Aproximación de circunferencias que intersecan 2 líneas.

Figura 3.3: Aproximación de curva semi-circular con 4 puntos de control.

La metodología se muestra en la Figura 14, y se detalla, a continuación.

- 1) *Ubicar 2 puntos relevantes.-* Se ubica un punto P_A en dirección rL_1 a distancia de radio de curvatura ρ . Despues, se ubica otro punto P_B en dirección rL_2 a la misma distancia de ρ .

- 2) *Líneas perpendiculares*.- Se contruye una recta perpendicular $R_A \perp rL_1$ a partir del punto P_A , y después otra recta $R_B \perp rL_2$ a partir de P_B .
- 3) *Lugar de intersección*.- Las rectas R_A y R_B se intersecan en algún lugar dentro del plano coor-denado (x, y) , punto denotado como e .
- 4) *Determinar el centro de la circunferencia*.- Para completar el proceso propuesto, se han considerado 3 diferentes escenarios para definir el punto central.
- a) El caso ideal es cuando el ángulo entre $rL_1 = rL_2 = 90^\circ$, de esta forma el punto de intersección entre R_A y R_B determina el nuevo centro de la circunferencia, como muestra en la Figura 14a. Una nueva intersección entre esta ecuación de circunferencia y las rectas rL_1 y rL_2 determina la nueva traza parcial con un ρ adecuado.
 - b) Este caso se muestra en la Figura 14b. Si el ángulo θ formado entre rL_1 y rL_2 se incrementa, entonces, la distancia $d_{(c, R_A \wedge R_B)}$ a la intersección $R_A \wedge R_B$ se aleja describiendo una curva exponencial, tal que:
- $$f(x) = \frac{(p_1 \cdot x^2 + p_2 \cdot x + p_3)}{(x^3 + q_1 \cdot x^2 + q_2 \cdot x + q_3)} \quad (3.18)$$
- c) Al igual que en el caso b, existe un ángulo θ formado entre rL_1 y rL_2 , que disminuye. Esta acción provoca un alejamiento del centro de la circunferencia hasta la intersección de R_A y R_B ; este cambio de distancia es exponencial y se ha resuelto con la ecuación (3.18).

Hasta el momento, se ha definido la nueva circunferencia. El siguiente paso es determinar los puntos de intersección entre las rectas rL_1 , rL_2 y la circunferencia cE . La recta definida en la ecuación (3.15) puede expresarse como un vector $\overrightarrow{rL_{i=2}^{nL}} = \vec{d} + \gamma\hat{n}$, donde, d es el vector de posición de un punto respecto a una recta, \hat{n} es un vector unitario en la dirección de la recta, y λ es un parámetro que se desliza por los límites de la recta. Por otra parte, una circunferencia vectorial expresada por la ecuación (3.17) como $(\overrightarrow{cE_{i=1}^{nL-1}} - \vec{c})^2 = r^2$, donde, \vec{c} es

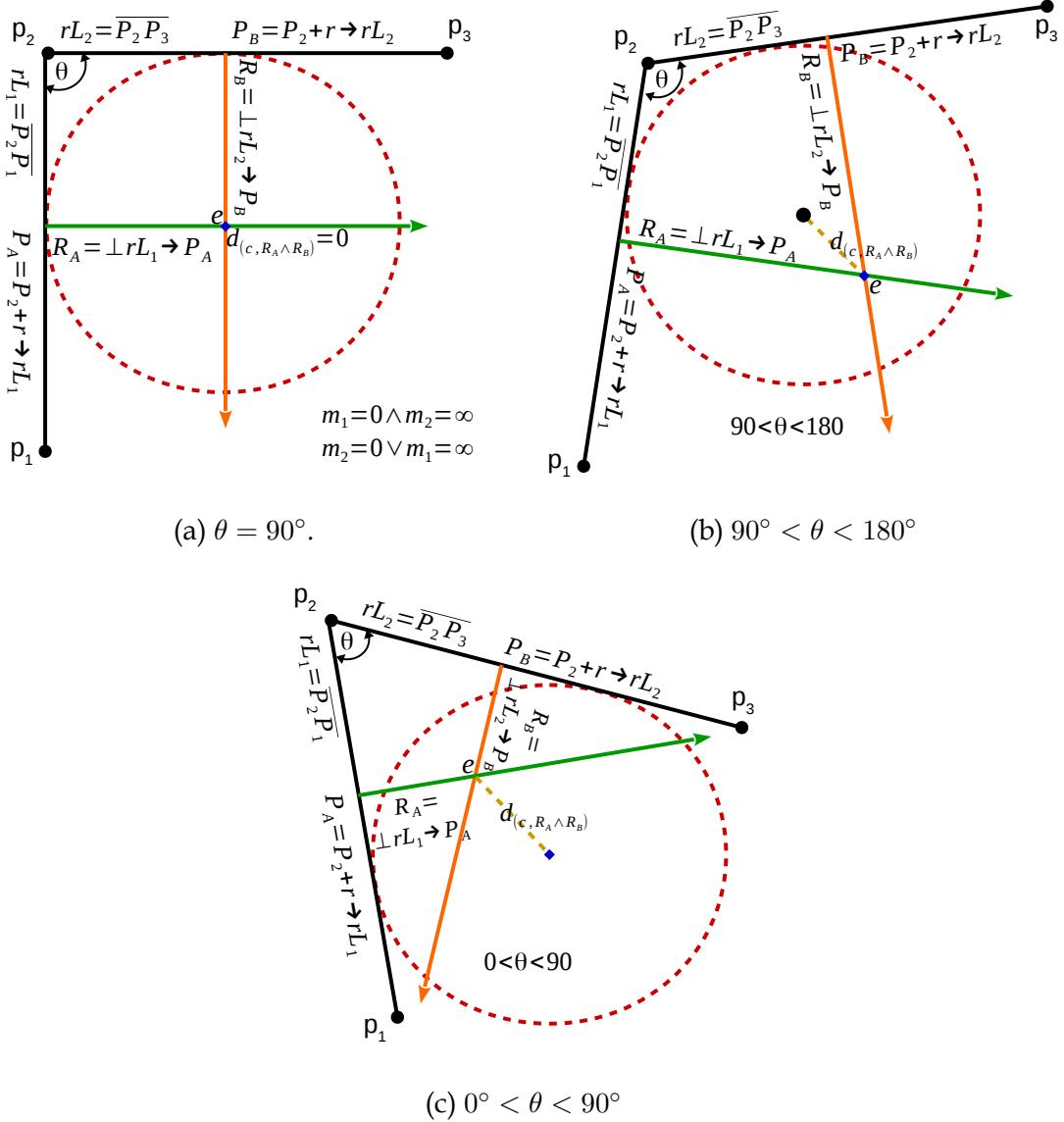


Figura 3.4: Aproximación de la curva semi-circular.

el vector posición del centro de la circunferencia, y r es el radio. Por lo tanto, la intersección de los dos sistemas se conecta, a partir de:

$$\left(\vec{d} + \gamma \hat{n} - c \right)^2 = r^2 \quad (3.19)$$

Cuadro 3.1: Condiciones del entorno de vuelo de prueba para el UAV.

<i>cInit</i>	[39.4207, -0.4231, 270]	[° dec.]
<i>cGoal</i>	[39.4457, -0.4337, 180]	[° dec.]
<i>Dimensiones UAV</i>	[2.4, 17.45, 0.42]	[m]
<i>radios de curvatura</i>	33	[rad]
<i>velocidad de vuelo</i>	18.16	[m/s]
<i>cabeceo</i>	± 21	[° dec.]
<i>alabeo</i>	± 21	[° dec.]

Por lo tanto, las posibles soluciones se definen como:

$$\begin{aligned}\vec{x}_\pm &= \vec{d} + (A \pm \sqrt{A^2 - B}) \cdot \hat{n} \\ A &= (\vec{c} - \vec{d}) \cdot \hat{n} \\ B &= (d^2 + c^2 - r^2 - 2 \vec{d} \cdot \vec{c})\end{aligned}\tag{3.20}$$

donde, B determina los posibles puntos de intersección entre las rectas (rL_1, rL_2) y la nueva circunferencia cE .

El proceso completo describe la curva en un plano (x, y) ; sin embargo, la curva contenida en el plano (x, z) se completa al replicar el proceso descrito, con la condición única de longitud idéntica en el plano (x, y) .

3.4 Experimentos y resultados

La tabla 3.1 describe las condiciones del entorno 3D, donde *cInit* representa el punto de partida y *cGoal* es el punto de meta del UAV, descritos en términos de latitud y longitud. El tamaño completo del entorno en metros es [1180, 2789, 300][m]. Por otro lado, es importante destacar que los parámetros de altitud son relevantes para asegurar un vuelo por completo 3D.

A partir del entorno descrito, se ha realizado una modificado, ubicando diversos obstáculos estáticos en diferentes lugares. La tabla 3.2 muestra la ubicación especificada de cada obstáculo. La columna [*obs*] muestra el número de obstáculos en la escena, la columna [*Ubicación*] muestra la ubicación geodésica de cada obstáculo y la

columna [Dimensiones] muestra las características dimensionales de los obstáculos.

Cuadro 3.2: Obstáculos, características y ubicaciones.

#	obs	Ubicación			Dimensiones
		lat. $^{\circ}$ dec	lon. $^{\circ}$ dec	alt.[m]	(x, y, z)[m]
1	1	39.4332	-0.4284	0	[13, 26, 30]
2	1	39.4362	-0.4269	150	[11, 22, 35]
	2	39.4302	-0.4314	150	[22, 22, 30]
	1	39.4359	-0.4257	100	[6, 6, 20]
3	2	39.4293	-0.4323	200	[6, 6, 20]
	3	39.4287	-0.4266	130	[13, 13, 30]
	4	39.4332	-0.4284	120	[20, 20, 60]

3.4.1 Planificación de trayectorias

A partir del entorno descrito, se ha realizado una modificación, ubicando diversos obstáculos estáticos en diferentes lugares. La tabla 3.3,muestra la ubicación especificada de cada obstáculo.La columna [obs] muestra el número de obstáculos en la escena, la columna [Ubicación] muestra la ubicación geodésica de cada obstáculo y la columna [Dimensiones] muestra las características dimensionales de los obstáculos.

Cuadro 3.3: Resultados de los Planificadores.

	escena	Iteraciones	Distancia [km]	Vértices	P _i
RRT	1	260	3.663	250	89
	2	368	3.684	350	90
	3	192	3.585	189	87
MACD	1	1016	3.369	378	19
	2	616	3.259	336	13
	3	808	3.298	414	21

Entonces, a partir de los resultados producidos por el planificador MACD, en la Figura 15 se muestra los resultados de la trayectoria en la *escena 2*, Específicamente, el literal a muestra una vista 3D. Las vistas laterales se muestran en los literales b

y c. De esta forma, este conjunto de resultados se utiliza para evaluar las curvas analizadas en la sección 3.3.

Asumiendo velocidades y aceleraciones constantes, la variación de ρ se produce a partir del ángulo resultante del cálculo de $cAct$ y $cNext$, donde, $cAct$ es el punto teórico real del UAV en la curva y $cNext$ está determinado por el punto de la curva. En esta instancia, se ha aplicado el algoritmo De Boore para evaluar las splines de diferentes órdenes, complementando una comparación con la curva semicircular descrita en la sección 3.3.2.

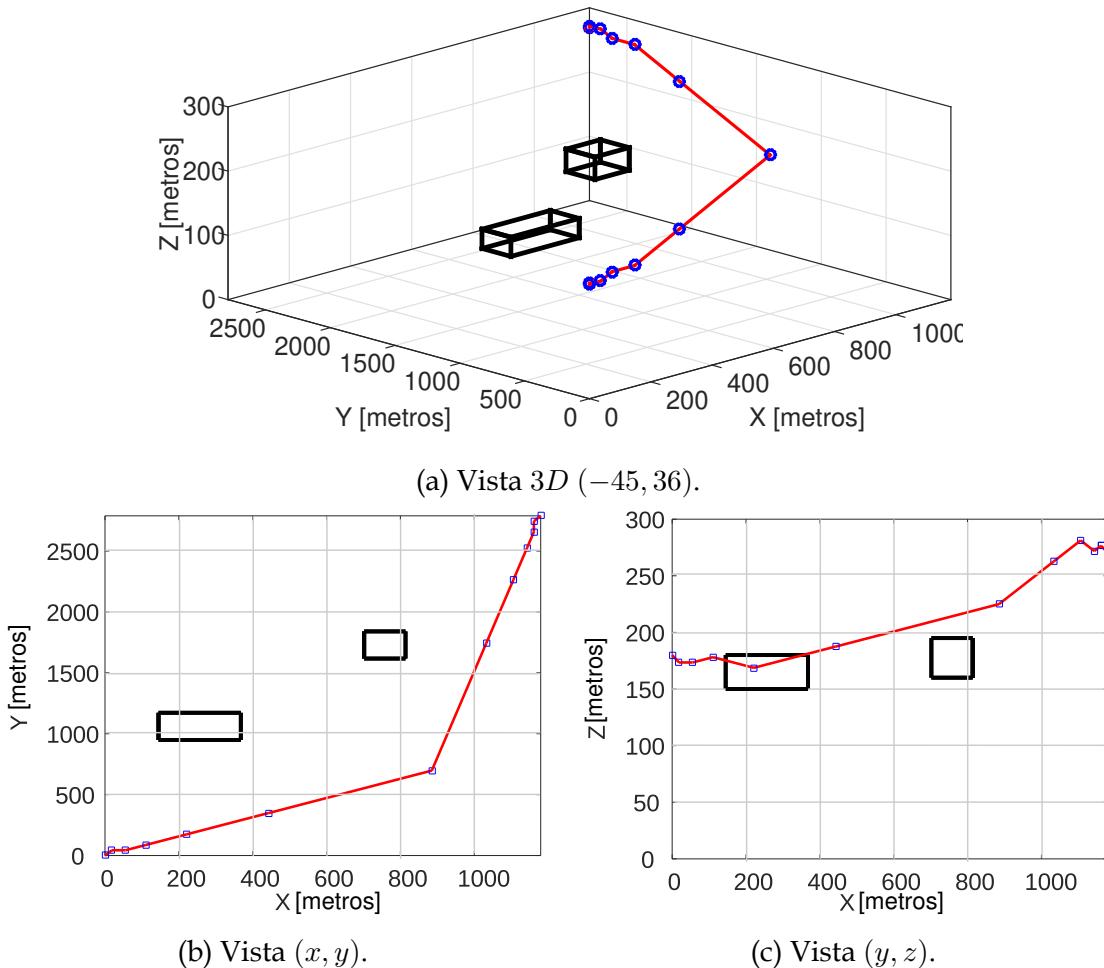


Figura 3.5: Resultados del Planificador de trayectorias MACD.

La Figura 16 muestra las curvas finales construidas mediante la metodología descrita. Donde, la Figura 16a y 16b muestran las vistas laterales de los planos (x, y) y

(x, z) . Los puntos rojos son los \mathbf{P}_i , la línea azul es la curva semicircular, las líneas amarilla, verde y naranja describen el algoritmo de De Boore de orden diferente. Por otro lado, la figura 16c muestra una ampliación de la parte superior derecha de la figura 16b, y visualiza los puntos de control p_1, p_2, p_3, p_4 y p_5 . Finalmente, la circunferencia negra punteada muestra el máximo ρ realizable por UAV en vuelo. De esta forma, se evidencia que para los polinomios de De Boore de orden 3 y 4, se generan aproximaciones en p_2 que superan la capacidad de maniobra del UAV.

Con el objetivo de comprobar la eficacia del método descrito, así como los resultados teóricos, se ha utilizado la herramienta Matlab/Simulink, y el simulador de vuelo FlightGear [88].

Por otro lado, durante el vuelo se han recogido 186,121 muestras provenientes del vuelo. Los resultados en variaciones de velocidades y aceleraciones tras finalizar la trayectoria suave se muestran en los literales a y b de la Figura 17.

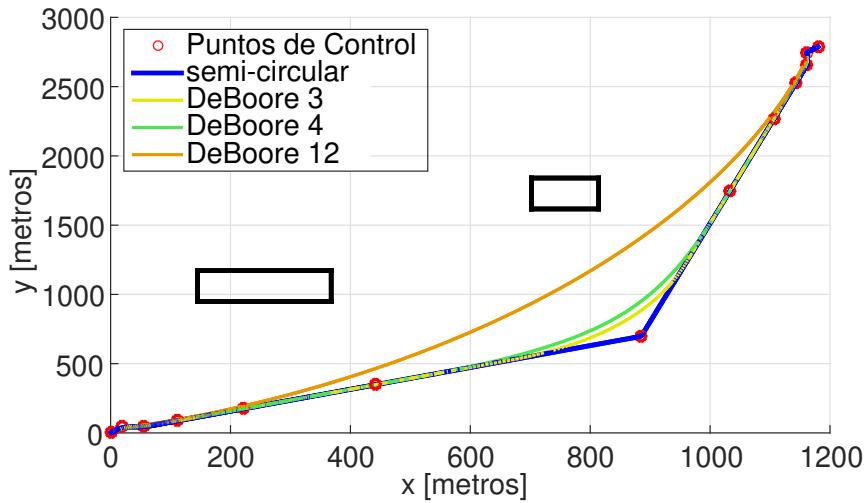
Las velocidades y aceleraciones mantienen su valor medio, lo que significa que las maniobras ejecutadas por el UAV permanecen dentro de su rango de restricción de movilidad. Por ejemplo, en velocidad de avance horizontal [*Vel.x*], la media es cercana a 18[m/s], lo que significa que el UAV mantiene un vuelo continuo.

3.5 Conclusiones

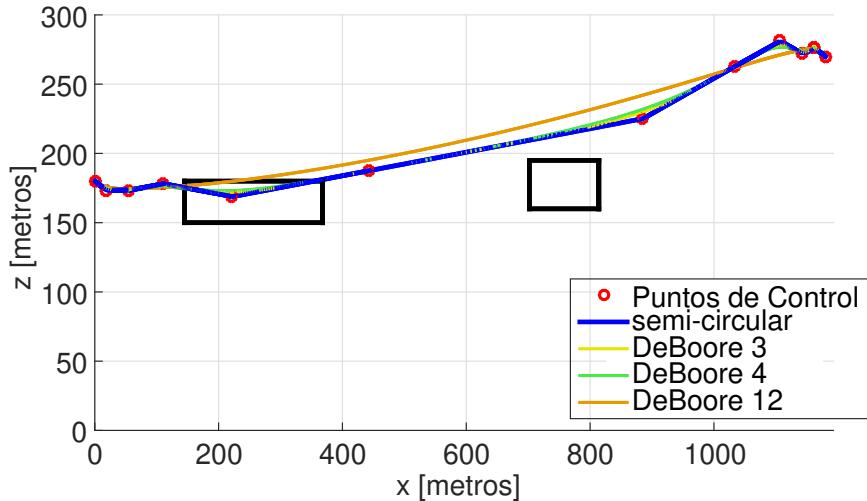
En este capítulo se ha desarrollado una propuesta de construcción de un planificador de trayectorias suaves para UAV de ala fija, en un espacio euclíadiano 3D. Así, el principal aporte ha sido la inclusión de restricciones y límites de curvatura en la construcción de las trayectorias.

Dentro del campo de la planificación de trayectorias, una reconstrucción discreta del entorno ofrece una forma directa de llegar desde *cInit* hasta *cGoal*. Por lo tanto, si los \mathbf{P}_i producidos por los planificadores arrojan la distancia adecuada, entonces, el UAV puede seguir la trayectoria suave final.

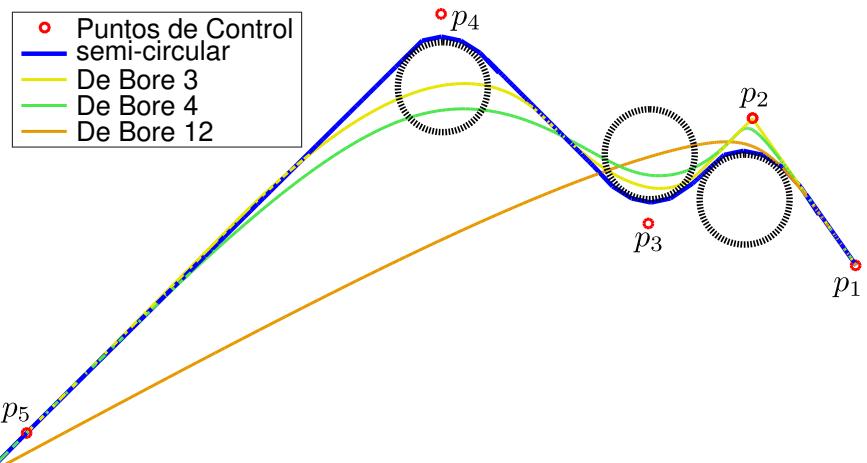
En resumen, antes de construir curvas que sirvan como transición hacia las trayectorias suaves, es importante tomar en cuenta diferentes características como restricciones de maniobrabilidad del UAV. Este enfoque brinda apoyo en la trazabilidad, veloci-



(a) Curvas, vista desde el plano (x, y).



(b) Curvas, vista desde el plano (x, z).



(c) Curvas, ampliación en la vista (x, z).

Figura 3.6: Construcción de curvas, desde los P1, en la escena 2.

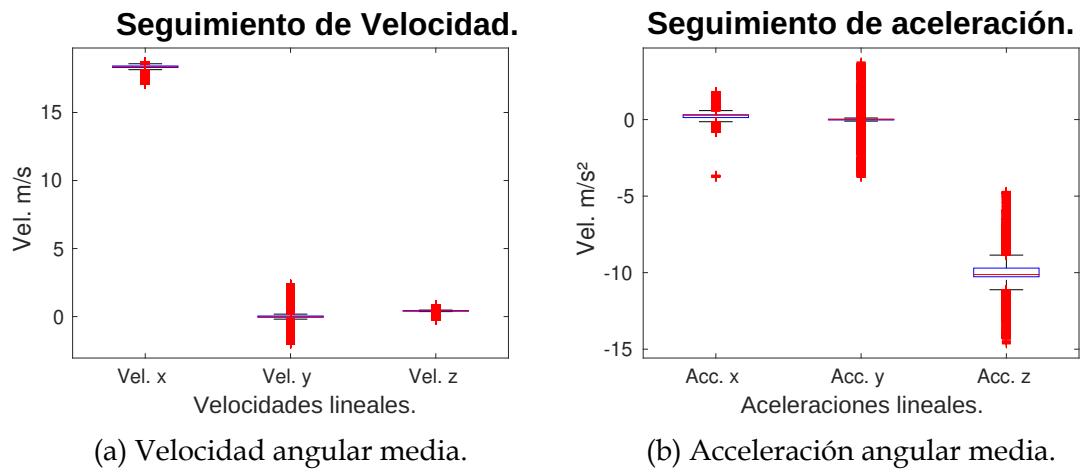


Figura 3.7: Ratio UAV de velocidades y aceleraciones en vuelo.

dades constantes y un vuelo continuo, al momento de su implementación y pruebas, lo que significa la posibilidad de incremento en el tiempo de vuelo del UAV.

04

Optimización multiobjetivo para la construcción de Trayectorias suaves 3D

RESUMEN

Este capítulo propone una arquitectura para la construcción de trayectorias de vuelo suave 3D para vehículos aéreos no tripulados (UAV) de ala fija. El objetivo es determinar la trayectoria de vuelo factible, minimizando el esfuerzo de giro, partiendo de un conjunto de puntos de control en el espacio 3D. A partir de los puntos de control definidos y las restricciones de movimiento del UAV, se ha construido una trayectoria que combina un conjunto de segmentos rectilíneos y segmentos de curvas esféricas. Esta metodología conlleva la posibilidad de infinitas soluciones para la construcción de la trayectoria final. Por este motivo, implica también un problema de optimización multi-objetivo (MOP) que logre maximizar de forma independiente cada uno de los radios de giro de la trayectoria. Finalmente, los resultados se contrastan a través de simulaciones por medio de MATLAB, Simulink y FlightGear.

4.1 Introducción

La rama tecnológica de los UAV experimenta un desarrollo constante y vertiginoso en diversos campos, especialmente en lo relacionado con nuevas técnicas de navegación y guiado. Esta evolución continua responde a nuevos desafíos que plantean las aplicaciones reales [89, 90, 91].

Es importante destacar que el problema más común en la determinación de trayectorias 3D de vuelo suave es la consideración de las restricciones intrínsecas del UAV. Por lo tanto, la no inclusión de las restricciones cinemáticas y/o dinámicas del UAV al momento de abordar el problema de planificación de trayectorias puede generar soluciones no viables que imposibiliten que el UAV complete una trayectoria de forma satisfactoria. Sin embargo, incluir en el diseño todas las restricciones del UAV en la fase de cálculo puede causar problemas de optimización muy complejos, sin solución única y con costes computacionales muy elevados.

Este capítulo se centra en la generación de trayectorias suaves, navegables por UAV de ala fija. Debido a las limitaciones no-holonómicas de los UAV de ala fija, el objetivo es crear una curva tridimensional suave desde un punto inicial hasta un punto objetivo, a través del espacio euclíadiano 3D con o sin obstáculos. Para lograr este objetivo, es esencial definir una trayectoria factible que minimice el esfuerzo de giro del vuelo y la distancia recorrida.

El conjunto de \mathbf{P}_i que definen el espacio libre de colisiones se calcula utilizando planificadores de trayectorias específicos, ya sean basados en el muestreo del entorno continuo o en el discreto. Algunos ejemplos de estas metodologías son el árbol aleatorio de exploración rápida (RRT) [92, 93, 94, 95]; las hojas de ruta probabilísticas (PRM) [96, 97, 98, 99, 100]; planificadores heurísticos (Algoritmos genéticos GA) [101, 102]; inteligencia de enjambre [103, 104, 105, 106]; lógica difusa [107, 108]; diagramas de Voronoi [109, 110, 111]; potencial artificial [112, 113, 114, 115]; o la descomposición celular adaptativa modificada con recompensa recursiva (RR-MACD) [116].

Las técnicas mencionadas, abordan el problema estándar de la planificación de trayectorias, construyendo trayectorias a trozos, ya sea en 2D o 3D. Estos métodos pueden proporcionar trayectorias óptimas o casi óptimas; sin embargo, no pueden garantizar la suavidad y la continuidad en la trayectoria, lo que podría dificultar el

guiado del UAV a través de las trayectorias. Además, estas técnicas no incorporan de forma directa las diferentes restricciones operativas del UAV y del entorno. Por lo tanto, este capítulo propone una metodología para definir trayectorias viables, suaves y navegables por UAV no-holonómicos, en las que se incluye las restricciones cinemáticas operativas del sistema.

Como punto de partida se determina un conjunto de puntos 3D libres de colisión S_{free} , a partir de una planificación de trayectoria 3D, con los que se construye un conjunto ordenado de rectas que definen una primera trayectoria. Dicha trayectoria es posteriormente suavizada, con el objetivo de incorporar la viabilidad y las restricciones del UAV. Las limitaciones del UAV se centran en su capacidad de giro horizontal y vertical. Por lo tanto, para que el UAV complete una secuencia de giros a una velocidad definida, debe determinar su radio de giro mínimo R_p . Si el radio de giro es demasiado pequeño, el UAV perderá la trayectoria; sin embargo, si el radio de giro aumenta, el UAV podrá realizar maniobras con menos esfuerzo.

El objetivo es mantener los resultados del planificador 3D y, al mismo tiempo, generar un conjunto finito de posibles curvas 3D que optimicen una curva 3D aproximada. Para ello, se plantea un problema de optimización multiobjetivo (MOP) [117]. Este planteamiento devuelve como resultado un conjunto de trayectorias que satisfacen las restricciones del UAV, expresadas como soluciones dominantes en un frente de Pareto de n -dimensiones [118]. Por último, se aplica criterios de selección para determinar la respuesta deseada desde el punto de vista de la curvatura κ y la torsión τ de la curva 3D. Para verificar la funcionalidad de la metodología, se compara los resultados de las curvas construidas después de la optimización de la curva 3D con una metodología conocida de aproximación tipo Bézier.

Este capítulo ha sido estructurado de la siguiente forma: en la sección 4.2.1 se presenta un breve resumen de los conceptos MOP; en la sección 4.3 se presenta la formulación del problema; la sección 4.4 detalla la metodología completa para que resuelve el problema, mientras que, la sección 4.5 detalla los experimentos y resultados de la planificación de trayectorias suaves 3D; por último, las conclusiones se presentan en la sección 4.6.

4.2 Preliminares

4.2.1 Optimización multiobjetivo

El concepto de optimización multiobjetivo es de sencillo entendimiento, pues su propósito principal es mejorar varios objetivos al mismo tiempo, considerando que optimizar varios objetivos en simultáneo es un problema de difícil alcance. Por lo tanto, la optimización multiobjetivo intenta alcanzar un compromiso entre mejorar ciertos criterios sin dañar demasiado a los demás. Por ejemplo, si se busca minimizar costos y maximizar calidad, en este problema, vale señalar que es inevitable sacrificar algo de calidad sin disminuir costos o viceversa.

Entonces, el problema de optimización (OP) intenta determinar una solución que represente el valor óptimo (mínimo o máximo) de una función, como $f : X \rightarrow \mathbb{R}$, donde, X es un vector de decisión factible, tal que $\min(f(x)) : x \in X$. Sin embargo, para problemas donde es necesaria la optimización simultánea de más de un objetivo, es decir, optimización multiobjetivo (MOP), la función tiene forma $f : x \rightarrow \mathbb{R}^k$, donde, $k \geq 2$ es el número de objetivos. Por lo tanto, el vector de valores de la función objetivo se puede definir como $f : X \rightarrow \mathbb{R}^k$, $f(x) = (f_1(x), \dots, f_k(x))^T$.

Sin embargo, no suele haber una única X que genere un óptimo que satisfaga simultáneamente cada uno de los k objetivos, debido al conflicto entre los objetivos. Entonces, el compromiso es encontrar una situación en la que todos los objetivos se encuentren satisfactoriamente dentro de unos parámetros aceptables. La solución MOP conduce a puntos en los que cualquier mejora en un objetivo provoca la degradación de cualquier otro objetivo (uno o varios). Así, estos puntos se representan como un frente de Pareto [118], donde, todos los puntos del frente son igualmente óptimos.

Por lo tanto, como se expresa en [117] el MOP puede expresarse como:

$$\min \mathbf{J}(\theta) = \min_{\theta \in D} [J_1(\theta), J_2(\theta), \dots, J_m(\theta)] \quad (4.1)$$

sujeto a:

$$\begin{aligned} g(\theta) &\leq 0 \\ h(\theta) &= 0 \\ \theta_{il} &\leq \theta_i \leq \theta_{iu}, i = [1, \dots, n], \end{aligned} \tag{4.2}$$

donde, $\theta \in \mathbb{R}^n$ es el vector de decisión, D es el espacio de decisión; $\mathbf{J}(\theta) \in \mathbb{R}^m$ es el vector objetivo; $g(\theta)$ y $h(\theta)$ son los vectores de restricción; y, por último, θ_{il} es el límite superior y θ_{iu} es el límite inferior del espacio de decisión. En consecuencia, no existe un único modelo óptimo; de hecho, hay un conjunto de soluciones óptimas con diferentes compensaciones entre objetivos, entre las cuales ninguna es mejor que las demás. Utilizando la definición de dominancia, el conjunto Pareto Θ_P es el conjunto de cada solución no dominada.

De esta forma, se define la dominancia de Pareto en el caso de que una solución θ^1 domine a otra solución θ^2 ; es decir, $(\theta^1 \prec \theta^2)$, si:

$$\forall i \in B, J_i(\theta^1) \leq J_i(\theta^2) \wedge \exists k \in B : J_k(\theta^1) < J_k(\theta^2), \tag{4.3}$$

donde, $J_i(\theta)$, $i \in B := [1 \dots m]$ son los objetivos a optimizar. Por lo tanto, el conjunto óptimo de Pareto Θ_P se define como:

$$\begin{aligned} \Theta_P &= \theta \in D | \nexists \tilde{\theta} \in D : \tilde{\theta} \prec \theta \\ J(\Theta_p) &= \{J(\theta) | \theta \in \Theta_p\}, \end{aligned} \tag{4.4}$$

donde, Θ_p y $J(\Theta_p)$ son soluciones MOP. Sin embargo, en la mayoría de los casos son inalcanzables porque Θ_P normalmente incluye soluciones infinitas. Por tanto, un conjunto finito de Θ_P^* a partir de Θ_P y otro conjunto finito de $J(\Theta_p^*)$ a partir de $J(\Theta_p)$ representan soluciones satisfactorias. A partir de $J(\Theta_p^*)$, el tomador de decisiones (DM) selecciona una solución, de acuerdo con las preferencias establecidas.

Por ejemplo, un cierto punto del frente de Pareto que está cerca del punto ideal (también llamado punto utópico) \mathbf{J}^{ideal} , definido como:

$$\mathbf{J}^{ideal} = \{J_{1 min}(\theta), \dots, J_{m min}(\theta)\}. \tag{4.5}$$

De ahí que una metodología apropiada para caracterizar una MOP es el conocido como algoritmo evolutivo multiobjetivo elitista (ϵ -MOGA) [117], que realiza una aproximación distribuida al frente de Pareto. El objetivo de ϵ -MOGA es encontrar una convergencia distribuida inteligente hacia un conjunto de ϵ -Pareto; es decir, determinar $\theta_{P_\epsilon}^*$ a lo largo del frente de Pareto $\mathbf{J}(\Theta_P)$. El espacio objetivo se divide en un número fijo de cajas. Por tanto, para cada dimensión $i \in B$, se crean celdas n_box_i de ancho ϵ_i , donde:

$$\begin{aligned}\epsilon_i &= (J_i^{max} - J_i^{min}) / n_box_i \\ J_i^{max} &= \max_{\theta \in \Theta_{P_\epsilon}^*} J_i(\theta), \quad J_i^{min} = \min_{\theta \in \Theta_{P_\epsilon}^*} J_i(\theta).\end{aligned}\tag{4.6}$$

Cada *box* puede estar ocupada por una única solución; por lo tanto, esta cuadrícula produce una distribución inteligente y preserva la diversidad de $\mathbf{J}(\Theta_{P_\epsilon}^*)$.

4.3 Definición del problema

Se asume un espacio de trabajo $W = R^3$, en el que es posible definir un conjunto de obstáculos estáticos o dinámicos como box terrestres o aéreas de diferentes dimensiones y ubicaciones (véase la Figura 18). En este espacio, el UAV en vuelo recibe datos de su estación de control (referentes a las condiciones ambientales) y a partir de estos se realiza los cálculos necesarios para determinar la mejor trayectoria suave 3D. Los datos relevantes incluyen el conjunto de puntos de control de vuelo 3D, ordenados y libres de colisiones $\rho = [P_1, \dots, P_5]$. Además, las capacidades intrínsecas de maniobra están determinadas por un radio de giro R_p (que determina las limitaciones de giro vertical y horizontal) definido por su velocidad de vuelo.

El objetivo es partir de ρ_{init} y alcanzar ρ_{goal} de tal forma que el UAV se aproxime a la trayectoria directa marcada por la secuencia ordenada ρ . Por tanto, $\rho = P_i(x_i, y_i, z_i)$, donde, $(i = 1, \dots, n)$ y n es el conjunto total de espacios libres de colisión S_{free} , puede expresarse como una secuencia de interpolación discreta $\rho = f(t_i) \rightarrow \mathbb{R}$, donde, $f(t_i)$ es un conjunto de nodos en el espacio 3D. En consecuencia, es posible establecer un conjunto de subintervalos $(n - 1)$ entre $i = 1$ y $i = n$ partitionados en $[a, b]$, definidos

como:

$$\begin{aligned} [a, b] &= [t_1, t_2] \cup [t_2, t_3] \cup \cdots \cup [t_{n-2}, t_{n-1}] \cup [t_{n-1}, t_n] \\ a = t_1 &\leq t_2 \leq \cdots \leq t_{n-1} \leq t_n = b \end{aligned} \quad (4.7)$$

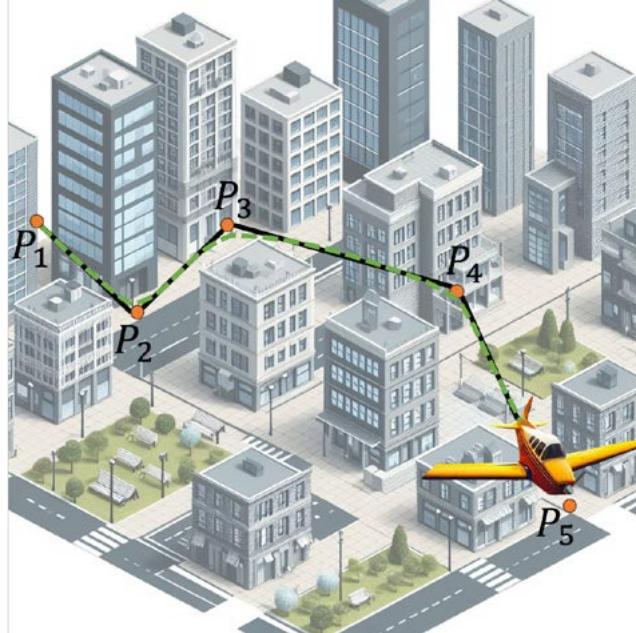


Figura 4.1: Perspectiva del problema de vuelo en 3D para un UAV de ala fija. El conjunto de puntos libres de colisión P_i se representa mediante puntos naranja; la línea negra describe la trayectoria discreta construida a partir de la planificación de trayectorias 3D; la línea verde discontinua representa la nueva trayectoria suave optimizada, seguible por el UAV.

Una unión lineal entre pares de puntos resulta entonces en $\mathbf{L} : [a, b] \rightarrow (x, y, z)$ y puede expresarse como un conjunto de rectas que marcan una trayectoria de vuelo directo $\mathbf{L}(t)$ dividido en $(n - 1)$ trozos.

$$\mathbf{L}(t) = \begin{cases} L_1(t) : t \in [t_1, t_2] \\ L_2(t) : t \in [t_2, t_3] \\ \vdots \\ L_n(t) : t \in [t_{n-1}, t_n] \end{cases} \quad (4.8)$$

$$\mathbf{L}(t) = L_1(t) + L_2(t) + \cdots + L_n(t).$$

Por tanto, $\mathbf{L}(t)$ es una función de interpolación lineal para la secuencia discreta $\rho = f(t_i)$. Asimismo, entre los puntos ρ existe un subconjunto de $(n - 1)$ rectas que unen el inicio y el final de la trayectoria a lo largo del espacio de vuelo sin colisiones.

Sin embargo, un UAV no-holonómico no puede realizar todos los tipos de maniobra definidas por $\mathbf{L}(t)$. En general, es deseable realizar maniobras con un radio de giro

elevado. Por lo tanto, el enfoque presentado construye una trayectoria suave a partir de ρ , que intenta evitar maniobras inadecuadas utilizando valores bajos de κ y τ , incluidos dentro de los límites del giro de vuelo del UAV, al tiempo que se acerca a la trayectoria $L(t)$.

En la Figura 18, el conjunto de puntos libres de colisión P_i se representa mediante puntos naranja, la línea negra, denotada como $L(t)$ muestra la trayectoria directa entre los puntos libres de colisión del entorno, y la línea verde punteada describe la curva suave 3D, definida como $C(t)$. La construcción de esta curva $C(t)$ se realiza uniendo un conjunto de segmentos que pueden ser de dos tipos: curvas esféricas S (definidas a partir de una esfera de radio R_p) o rectas L . Así, cada segmento S está definido por tres puntos continuos de ρ ; mientras el segmento S tiene dos puntos de tangencia, uno por cada par de rectas adyacentes $L(t)$ formadas por el conjunto actual de tres puntos de ρ .

Por lo tanto, cada segmento S puede tener infinitas soluciones, con cada radio R_p , dando lugar a diferentes puntos de tangencia en las rectas $L(t)$. En consecuencia, para cada segmento S se puede definir un conjunto infinito de esferas, que se enlanzan a través de los segmentos L correspondientes o de otro segmento S . Obviamente, este planteamiento del problema sugiere la existencia de infinitas combinaciones para los segmentos S y L . La forma de abordar esta problemática ha sido mediante el planteamiento de un MOP.

4.4 Metodología

En esta sección se describe la metodología propuesta para la generación de trayectorias suaves en 3D. El método se divide en dos partes, primero se detalla cómo se obtuvieron los segmentos S y luego se describe la unión con los segmentos L .

4.4.1 Definición del segmento esférico

Si se asume una trayectoria discreta definida como el conjunto de puntos libres de colisión $\rho = [P_1, \dots, P_n]$, a partir de la Figura 19 (puntos rojos), entonces, este conjunto de puntos se define como: $P_i(x_i, y_i, z_i) : i = \{1, \dots, n\}$, donde $p_{init} = P_i(x_i, y_i, z_i) :$

$$i = \{1\} \text{ y } p_{goal} = \mathbf{P}_i(x_i, y_i, z_i) : i = \{n\}.$$

El literal b de la Figura 19b muestra una esfera osculante (oS) [119] definida con un valor mínimo Rp , situada entre el conjunto de los 3 primeros \mathbf{P}_i y tangente a las rectas $L(t)$ formadas entre el mismo conjunto de \mathbf{P}_i . Por tanto, teniendo en cuenta el número de puntos libres de colisión ρ , el conjunto de esferas es igual a $G_i : i = \{1, \dots, n-2\}$, como se muestra en el literal c de la figura 19c (vista ortogonal).

El literal b de figura 19b muestra el primer $G_i : i = 1$ situado entre los tres primeros \mathbf{P}_i . Por lo tanto, es posible definir un plano $\pi_i : i = 1$ entre los mismos puntos \mathbf{P}_i , que tienen un ángulo en relación con la ubicación del conjunto actual de \mathbf{P}_i , como puede verse en la Figura 20a y 20b. La importancia de la definición de este plano viene dada por el hecho de que dentro del mismo está contenido el centro de G_i con radio Rp .

De esta forma, existe una curva S_i auto contenida (como una serie de puntos a lo largo del espacio euclídeo) sobre la superficie de la esfera y tangente a $L(t)$ con t_2 y t_3 en el plano π_i , como se muestra en la Figura 20. Por tanto, el segmento de curva S_i (línea negra) se define como:

$$\begin{aligned} S_i(t) &= [S_x, S_y, S_z] \\ \left. \begin{array}{l} S_{i_x} = x_0 + Rp * \sin(\psi) * \cos(\varphi) \\ S_{i_y} = y_0 + Rp * \sin(\psi) * \sin(\varphi) \\ S_{i_z} = z_0 + Rp * \cos(\psi) \end{array} \right\} & \varphi_1 \geq \varphi \geq \varphi_2 \quad (4.9) \\ & \wedge \\ & \left. \begin{array}{l} \psi_1 \leq \psi \leq \psi_2. \end{array} \right\} \end{aligned}$$

donde, x_0 , y_0 y z_0 representan el centro de G_i . La curva S_i realiza un recorrido horizontal y vertical debido a los rangos angulares de ψ y φ , lo que implica variaciones en los valores de κ y τ (éstos tienen una conexión directa con Rp y la longitud de arco de S_i).

En consecuencia, si el valor de Rp crece, S_i también crece, mientras que κ y τ disminuyen.

Es importante remarcar que si el plano π_i es paralelo al plano horizontal (x, y) del entorno, entonces $\tau = 0$, lo que implica que los movimientos del UAV sean horizontales. Del mismo modo, si π_i es paralelo al plano vertical (x, z) del entorno, entonces $\kappa = 0$.

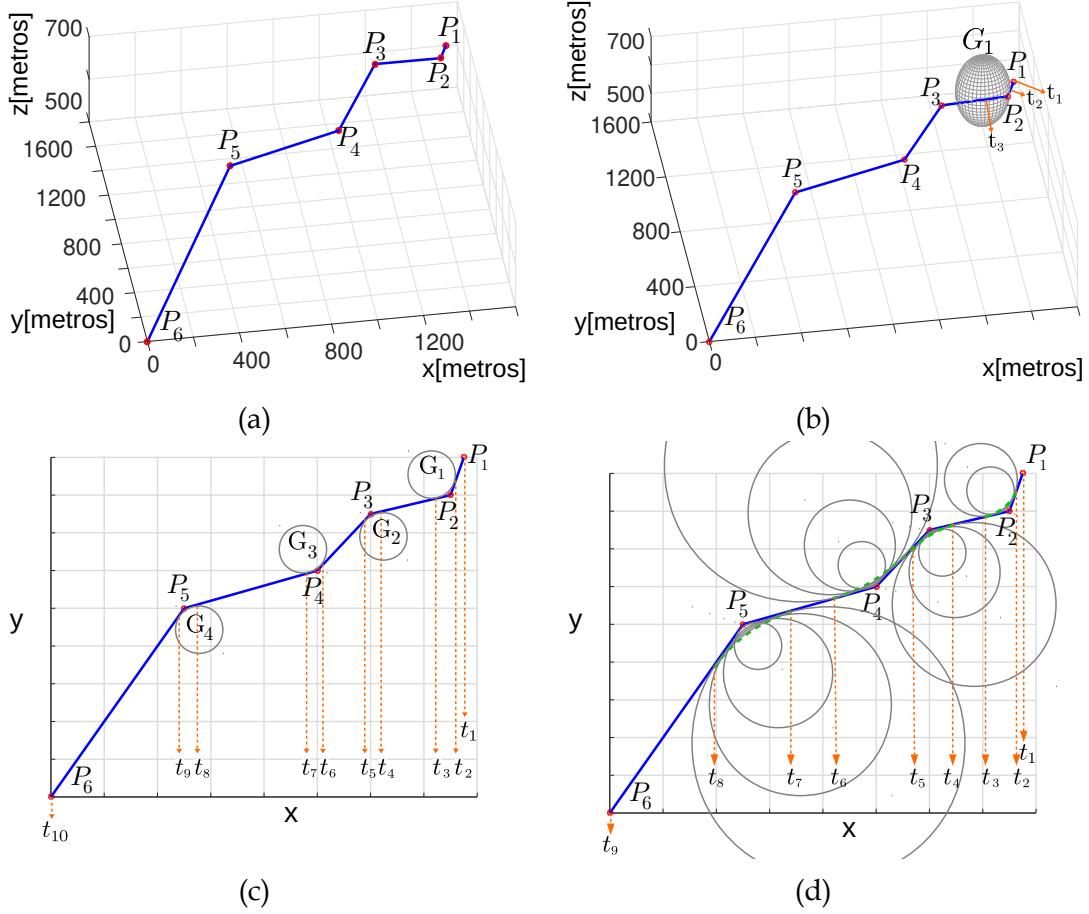


Figura 4.2: Problema de planificación de trayectorias suave. Los puntos rojos representan ρ , la línea azul es la trayectoria formada por rectas $L(t)$. (a) Puntos libres de colisión ρ . (b) Definición de esfera con una relación del mínimo R_p . (c) Conjunto de G_i sobre ρ . (d) Ejemplo de trayectoria suave óptima, con κ y τ optimizados en líneas verdes discontinuas.

Sin embargo, antes de aplicar la ecuación (4.9), es necesario determinar la situación de los puntos (x_0, y_0, z_0) , de modo que G_i sea tangente en un punto de su superficie con $L(t)$, como se muestra en la Figura 19b en los puntos $(t_2$ y $t_3)$. No obstante, se debe considerar que existe un ángulo entre cada par de $L(t)$, y esto hace que G_i se acerque o se aleje de las rectas y sus puntos tangentes. El análisis geométrico aplicado para llegar a una solución óptima se detalla a continuación.

En primer lugar, una dirección vectorial en el espacio puede definirse como $\vec{v} = p - q : p \wedge q \in \mathbb{R}^3$. Por tanto, partiendo de los datos conocidos $\rho = [P_1, \dots, P_n]$,

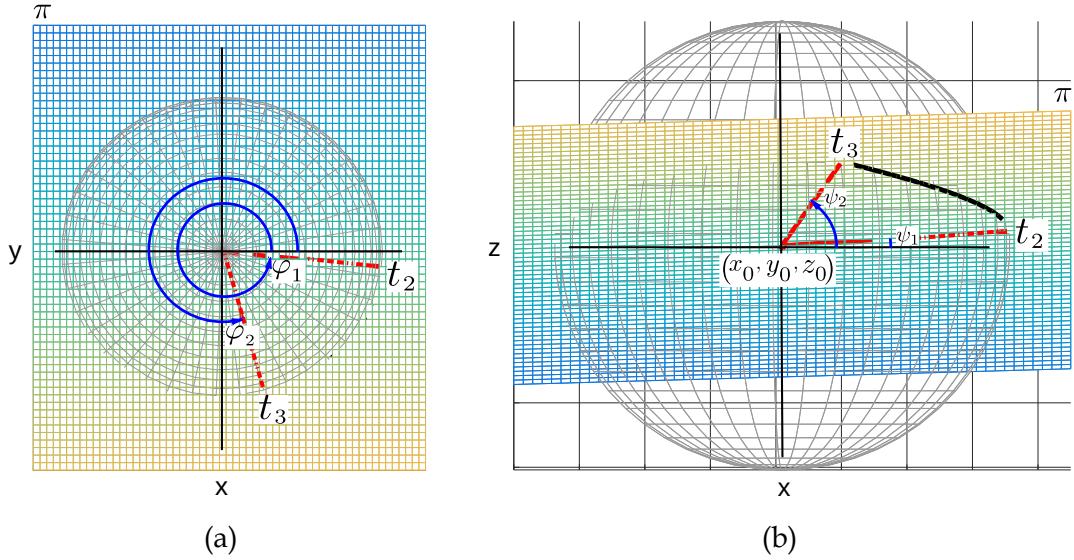


Figura 4.3: Trayectoria esférica suave, la línea negra muestra la curva en un segmento S_i a lo largo del plano π_i . Las líneas rojas muestran la unión desde el centro de coordenadas de la esfera hasta los puntos de intersección entre la esfera y el plano π_i resultante de la semicurva esférica. (a) Vista perpendicular al plano horizontal (x, y). (b) Vista perpendicular al plano vertical (x, z).

tomando como ejemplo la Figura 19, donde se asume que los puntos iniciales libres de colisión se definen como $(\mathbf{P}_i : \{i = 1, \dots, 3\})$, determinan un primer conjunto de dos vectores como:

$$\left. \begin{array}{l} \vec{u}_i = p - q : p = \mathbf{P}_{(i+1)}, q = \mathbf{P}_{(i)} \\ \vec{v}_i = p - q : p = \mathbf{P}_{(i+1)}, q = \mathbf{P}_{(i+2)} \end{array} \right\}, i = 1 \quad (4.10)$$

De la misma forma, un vector perpendicular desde \vec{u}_i hasta \vec{v}_i , denotado como $\vec{\eta}$, define el vector normal, tal que:

$$\vec{\eta} = \vec{u}_i \times \vec{v}_i. \quad (4.11)$$

En consecuencia, la ecuación paramétrica del plano π_i que contiene tres puntos se define como:

$$\begin{aligned}\pi_i = & \left\{ \begin{bmatrix} (x - p_x) \\ (y - p_y) \\ (z - p_z) \end{bmatrix} * [\vec{\eta}] \right\} \\ & : \begin{bmatrix} p_x = \mathbf{P}_{(i+1)_x} \\ p_y = \mathbf{P}_{(i+1)_y} \\ p_z = \mathbf{P}_{(i+1)_z} \end{bmatrix}\end{aligned}\quad (4.12)$$

Del mismo modo, la distancia euclídea definida entre dos puntos p y q viene dada por:

$$d(p, q) = \sqrt{\sum (p - q)^2} \quad (4.13)$$

Por lo tanto, se pueden expresar dos distancias como $du_j : p = \mathbf{P}_{(i+1)}, q = \mathbf{P}_{(i)}$ y $dv_j : p = \mathbf{P}_{(i+1)}, q = \mathbf{P}_{(i+2)}$. Por último, el ángulo entre \vec{v}_i y \vec{u}_i se define por:

$$\angle(\vec{u}_i, \vec{v}_i) = \phi_i = \tan^{-1} \frac{\|\vec{u}_i \times \vec{v}_i\|}{\vec{u}_i \cdot \vec{v}_i}. \quad (4.14)$$

Por lo tanto, con la ecuación (4.14) y conocido el valor de Rp , los puntos tangenciales a las rectas $\mathbf{L}(t)$ pueden localizarse a una distancia definida como:

$$\sigma_i = \frac{Rp}{\phi_i/2}. \quad (4.15)$$

De este modo, dos puntos espaciales definidos como pUi_i y pUg_i , situados en la dirección del vector \vec{u}_i proporcionan $\mathbf{P}_i = \mathbf{P}_{i+1}$, y $\mathbf{P}_g = \mathbf{P}_i$ y $d(p, q) = du_i$; por tanto:

$$\begin{aligned}\gamma &= \sigma_i / d(p, q) \\ pUi_j &= (\mathbf{P}_i - \mathbf{P}_g) * \gamma + \mathbf{P}_i \\ pUg_j &= -(\mathbf{P}_i - \mathbf{P}_g) * \gamma + \mathbf{P}_i\end{aligned}\quad (4.16)$$

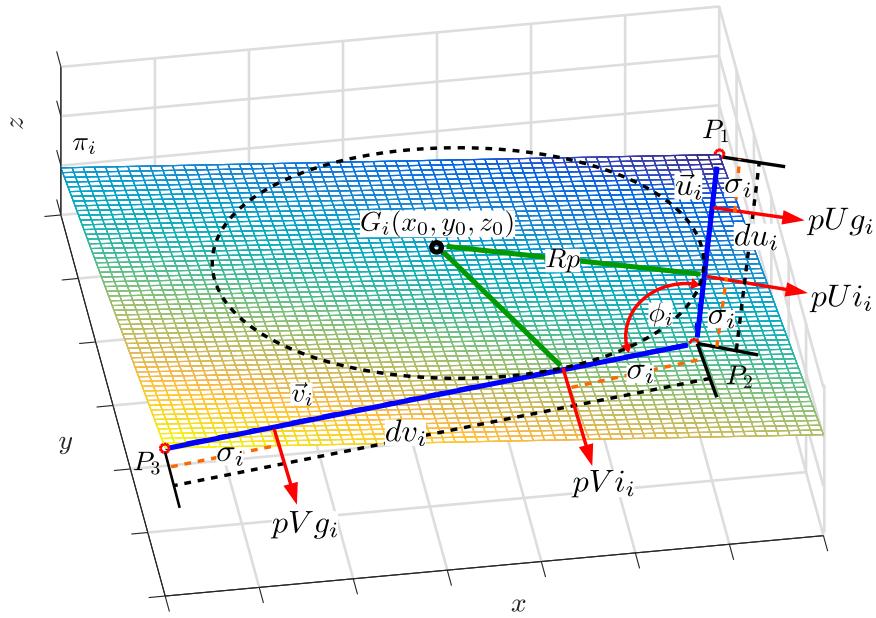
Del mismo modo, se pueden definir dos puntos pVi_i y pVg_i en la dirección vectorial \vec{v}_i , siempre que $\mathbf{P}_i = \mathbf{P}_{i+1}$, $Pg = \mathbf{P}_{i+2}$ y $d(p, q) = dv_i$, según la ecuación (4.16). De modo que, la bisectriz perpendicular de pUi_i y pVi_i en el plano π_i determina el centro

de la esfera (x_0, y_0, z_0) . El literal a de la Figura 21a muestra la aplicación de las ecuaciones (4.10), (4.11), que pueden repetirse a lo largo de los sucesivos puntos libres de colisión ρ . Entre los centros de las esferas (x_0, y_0, z_0) y los puntos de intersección con $\mathbf{L}(t)$ se encuentran los ángulos de desplazamiento φ y ψ de los segmentos S_i , como puede verse en la Figura 20, donde $pUi_i = t_2$ y $pVi_i = t_3$.

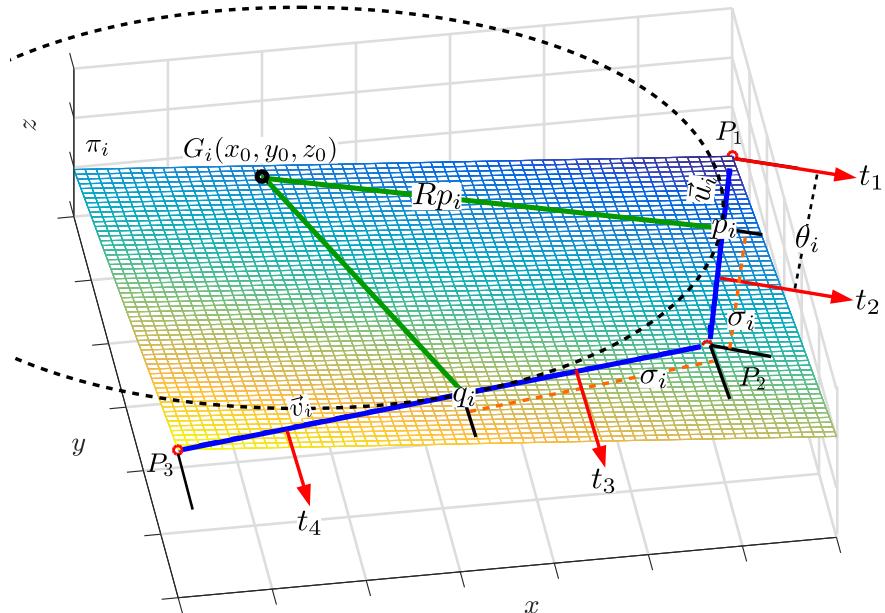
Es importante remarcar que, independientemente de la condición angular producida por el par de rectas $\mathbf{L}(t)$ denotado en la ecuación (4.14), el ángulo formado entre los puntos de intersección sobre la esfera G_i , vista desde su centro hacia la componente vertical u horizontal, no supera en ningún caso 90° ; es decir, $(0^\circ < \varphi < 90^\circ)$ y $(0^\circ < \psi < 90^\circ)$.

El proceso descrito muestra el análisis geométrico para la localización del conjunto de esferas G_i definidas con radio constante Rp , como puede verse en la Figura 19c. Además, existe un conjunto de cuatro segmentos S y otro conjunto de cinco segmentos L , siendo los segmentos S los comprendidos por los intervalos $[t_2, t_3]$, $[t_4, t_5]$, $[t_6, t_7]$ y $[t_8, t_9]$, mientras que los segmentos L están incluidos en los intervalos $[t_1, t_2]$, $[t_3, t_4]$, $[t_5, t_6]$, $[t_7, t_8]$ y $[t_9, t_{10}]$. El objetivo es aumentar el radio Rp en cada segmento, de modo que se minimicen los valores de κ y τ a lo largo de la curva, aumentando el radio Rp_i en cada G_i .

La solución adoptada consiste en desplazar el punto de intersección de cada esfera G_i en la dirección del segmento adyacente $\mathbf{L}(t)$. En consecuencia, $G_i : i = 1$, se aproxima de forma simétrica a los intervalos t_1 y t_4 , entonces, $G_i : i = 2$ realiza la correspondiente aproximación a los intervalos t_3 y t_6 , y así de forma sucesiva. Por tanto, en la Figura 21b, los segmentos $\mathbf{L}(t)$ pueden verse adyacentes a $G_i : i = 1$, denotados como $[t_1 \equiv P_1, t_2 \equiv pUi_j, t_3 \equiv pVi_j, t_4 \equiv pVg_j]$.



(a)



(b)

Figura 4.4: El círculo negro discontinuo muestra la esfera osculante (oS); la línea verde es el radio de giro R_{p_i} . (a) Situación de G_i con radio de giro mí-nimo. (b) Situación de G_i con radio de giro superior desplazado dentro de los límites $[t_1, t_2]$, definido por el valor de θ_i .

Así, entre los intervalos adyacentes $[t_1, t_2]$ se define un vector $\vec{u}_i = t_2 - t_1$, asociado a este vector se establece un punto espacial p_i definido por la ecuación paramétrica:

La solución adoptada consiste en desplazar el punto de intersección de cada esfera G_i en la dirección del segmento adyacente $L(t)$. En consecuencia, $G_i : i = 1$, se aproxima de forma simétrica a los intervalos t_1 y t_4 , entonces, $G_i : i = 2$ realiza la correspondiente aproximación a los intervalos t_3 y t_6 , y así de forma sucesiva. Por tanto, en el literal b de la Figura 21b, los segmentos $L(t)$ pueden verse adyacentes a $G_i : i = 1$, denotados como $[t_1 \equiv P_1, t_2 \equiv pUi_j, t_3 \equiv pVi_j, t_4 \equiv pVg_j]$.

Así, entre los intervalos adyacentes $[t_1, t_2]$ se define un vector $\vec{u}_i = t_2 - t_1$, asociado a este vector se establece un punto espacial p_i definido por la ecuación paramétrica:

$$\left. \begin{array}{l} p_{i_x} = t_{2x} + \theta_i * \vec{u}_{i_x} \\ p_{i_y} = t_{2y} + \theta_i * \vec{u}_{i_y} \\ p_{i_z} = t_{2z} + \theta_i * \vec{u}_{i_z} \end{array} \right\}, 0 \leq \theta_i \leq 1 \quad (4.17)$$

donde, θ_i define el punto espacial p_i a lo largo de \vec{u}_i y dentro de los intervalos $[t_1, t_2]$. Por lo tanto, el valor de la distancia σ_i desde P_{i+1} hasta p_i se define según la ecuación en (4.13), siendo, $p = P_{i+1}$ y $q = p_{i+2}$. Se define un punto del espacio simbólico q_i entre los intervalos $[t_3, t_4]$ con dirección $\vec{v}_i = t_3 - t_4$ a la misma distancia σ_i . Entonces, σ_i también posee un ángulo ϕ_i , y según la ecuación (4.14), es posible definir un nuevo Rp_i a partir de la ecuación (4.15), la que tiene un valor de radio mayor. Por último, la bisectriz perpendicular de p_i y q_i en el plano π_i determina el centro de $G_i(x_0, y_0, z_0)$ (ver Figura 21b). Por lo tanto, el centro de G_i define la ecuación (4.9) en el segmento S_i y sobre este segmento se determinan los valores inferiores de κ y τ según las ecuaciones (2.6) y (2.10).

Definición del problema multiobjetivo (MOP)

Dada la ecuación (4.17), es importante señalar que cualquier valor de θ_i entre 0 y 1, define un punto espacial entre el intervalo $[t_1, t_2]$. Del mismo modo, es importante resaltar que dentro de los límites de θ_i , existe un número infinito de puntos espaciales con un número infinito de radios Rp_i y su correspondiente número infinito de G_i , con los que se construye los correspondientes segmentos S_i .

Por este motivo, para obtener una solución óptima, se ha resuelto el MOP mediante algoritmos evolutivos basados en el concepto de ϵ -dominancia [120, 121]. Para lo que se ha definido variables de decisión, condiciones iniciales del proceso, restricciones del MOP y el vector índice a optimizar para representar el frente de Pareto. Entonces, si se supone que el número de esferas G_i es igual a m , y el número de objetivos para cada G_i igual a dos, entonces, $J^{ideal}(\theta) = [J_1(\theta), J_2(\theta), \dots, J_{2*m}(\theta)]$ representa el vector de objetivos, donde, J_i denota el i^{esimo} objetivo. En consecuencia, $J_i^A = \min(\kappa(\theta_i), J_i^B) = \min(\tau(\theta_i)) \in G_i : [i = 1, \dots, m]$, donde, J_i^A y J_i^B dependen del vector de variables de decisión θ . Asumiendo D como un espacio de decisión dentro de un subconjunto \mathbb{R}^D , siendo, θ el vector de variables de decisión compuesto por un conjunto de θ_i para todo $i \in 1 < i < m$, con θ_i es $[0, 1]^D$. En consecuencia, el problema MOP puede enunciarse como:

$$\min_{\theta \in D} [J_i^A(\theta), J_i^B(\theta)]_{1 \times (2*m)}, \quad \forall i \in 1 \leq i \leq m. \quad (4.18)$$

donde:

$$\begin{aligned} J_i^A &= \frac{\|S'_i(t) \times S''_i(t)\|}{\|S'_i(t)\|^3} \\ J_i^B &= \frac{S'_i(t) \cdot [S''_i(t) \times S'''_i(t)]}{\|S'_i(t) \times S''_i(t)\|^2} \\ \theta &= [\theta_i]_{1 \times m}, \quad \forall i \in 1 \leq i \leq m \end{aligned}$$

sujeto a:

$$S_i(t) = \begin{cases} S_{i_x} = x_0 + Rp_i * \sin(\psi) * \cos(\varphi) \\ S_{i_y} = y_0 + Rp_i * \sin(\psi) * \sin(\varphi) , \\ S_{i_z} = z_0 + Rp_i * \cos(\psi) \end{cases}$$

desde ec. (4.9)

$$Rp_i = \sigma_i * (\phi_i/2), \text{ desde ec. (4.15)}$$

$$\sigma_i = \sqrt{\sum (p_i - P_{i+1})^2}, \text{ desde ec. (4.13)}$$

$$p_i = \begin{cases} p_{i_x} = t_{2_x} + \theta_i * \vec{u}_{i_x} \\ p_{i_y} = t_{2_y} + \theta_i * \vec{u}_{i_y} , \text{ desde ec. (4.17)} \\ p_{i_z} = t_{2_z} + \theta_i * \vec{u}_{i_z} \end{cases}$$

$$\theta_i \in [0, 1]^D$$

En resumen, el objetivo es encontrar una curva suave 3D óptima que minimice κ y τ en cada una de las posibles S_i . Es importante mencionar que las esferas adyacentes G_i pueden crecer una dentro de la otra, hasta que un máximo de $q_i \in \vec{v}_i \equiv p_{i+1} \in \vec{u}_i$, lo que implica una disminución del conjunto total de segmentos, como describe la Figura 19d, donde la línea verde discontinua muestra el conjunto de segmentos S_i pertenecientes a G_i .

Un ejemplo de reconstrucción según la respuesta Θ_P^* puede verse en el literal b de Figura 20b, donde la reconstrucción S se realiza en cuatro segmentos, definidos por las fronteras $[t_2, t_3]$, $[t_4, t_5]$, $[t_5, t_6]$ y $[t_7, t_8]$. Los segmentos S pertenecientes a $C(t)$ se definen según la ecuación (4.9).

En contraste, y con referencia a la Figura 19d, los segmentos L están definidos por el resto de los límites, siendo tales límites $[t_1, t_2]$, $[t_3, t_4]$, $[t_6, t_7]$ y $[t_8, t_9]$.

4.4.2 Definición de segmentos de línea recta

Una trayectoria de tipo línea recta L puede describirse mediante dos puntos en el espacio euclídeo. El literal d de la Figura 19d muestra un ejemplo de un segmento

\mathbf{L} definido por los puntos $[t_1, t_2]$, donde, la dirección de la línea está dada por la trayectoria de vuelo del UAV. Por tanto, \vec{u} (ver la Figura 22) es un vector unitario que apunta en la dirección de la orientación deseada, y con d definido como la distancia entre t_1 y t_2 según la ecuación (4.13). Por tanto, los segmentos \mathbf{L} se describirán, en general, como:

$$\begin{aligned} \mathbf{L}(t) &= \{r \in \mathbb{R}^3 : r = (t_1 - t_2) * \gamma + t_1\} \\ &\rightarrow 0 \leq \gamma \leq d \end{aligned} \quad (4.19)$$

Finalmente, la interpolación de S y \mathbf{L} construye una curva suave 3D final en el plano (x, y, z) .

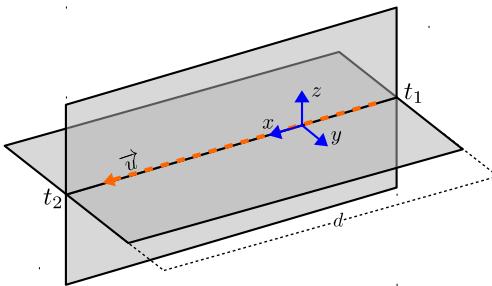


Figura 4.5: Segmento de línea recta.

4.5 Experimentos y resultados

MATLAB/Simulink y el simulador de vuelo flightGear han sido las herramientas de apoyo para la presentación de los resultados de simulación por ordenador. En este sentido, se ha realizado el análisis de cinco escenarios en el espacio 3D, tomados a partir de la metodología propuesta en [122]. La metodología RR-MACD ofrece dos conjuntos de resultados basados en las restricciones definidas. Los resultados presentados en [122] se muestran de forma resumida en la Tabla 4.1, donde la primera columna muestra el número de escenario. La segunda columna muestra el RR-MACD con cuatro restricciones y el RR-MACD con 10 restricciones; en cambio, en la tercera columna se muestran las condiciones para resolver el problema de planificación de trayectorias. Los puntos de control 3D reflejados en la Tabla 4.1, $\rho_x(F) \approx \rho$,

son los puntos de partida para la construcción de curvas suaves 3D, además, del análisis del método descrito en este capítulo.

Por último, es importante señalar que los algoritmos se han ejecutado en una CPU Intel(R) Core(TM) i7 – 4790 a 3,60 GHz (Fabricante: Gigabyte Technology Co., Ltd., Modelo: B85M – D3H) con 8Gb de RAM y S.O. Ubuntu Linux 16,04 LTS. Los algoritmos se programaron en MATLAB versión 9,4,0,813654 (R2018a).

Cuadro 4.1: Resultados de la planificación de trayectorias 3D. El número de espacios colisión libre se definen como S_{free} , mientras que el número de nodos 3D discretos se denotan por $\rho_x(F)$ [122].

Env.	4 Restricciones		10 Restricciones	
	RR-MACD		RR-MACD	
	# S_{free}	# $\rho_x(F)$	# S_{free}	# $\rho_x(F)$
# 1	115	18	202	27
# 2	27	8	35	10
# 3	19	6	16	7
# 4	11	6	51	10
# 5	19	7	35	10

Es importante mencionar que las características del UAV asumidas en los experimentos han sido tomadas de [123], un estudio sobre el UAV de ala fija kadett 2400, representado por seis estados ($x, y, z, \phi, \theta, \psi$) donde los tres primeros estados definen el vector de posición del sistema de coordenadas global del UAV, situado en el origen de su centro de gravedad. Los tres últimos son los ángulos de Euler de balanceo, cabeceo y guiñada, respectivamente, que definen la orientación del UAV.

Por último, las simulaciones han mostrado que el UAV mantiene un vuelo continuo a una velocidad constante de 18[m/s], dentro de un radio de curvatura mínimo establecido como $R_p = 33[m]$, lo que muestra un comportamiento suave y sin maniobras que puedan poner en peligro la integridad del avión.

Es importante recordar que, debido a que en el ejemplo específico de análisis el número de puntos libres de colisión $\rho_x(F) = [P_1, \dots, P_n]$ es superior a cinco, es necesario un método de visualización adecuado para la toma de decisiones en la solución final. Por ello, se ha utilizado el método de representación gráfica denominado diagrama de niveles [124], que consiste en representar cada objetivo y cada parámetro

de diseño en diagramas separados, sincronizados con su eje y. La sincronización se realiza con distancia normalizada de cada punto del frente de Pareto hasta punto ideal.

4.5.1 Ejemplo aplicativo

Para representar los resultados visuales y numéricos de la Tabla 4.1, a continuación, se detalla los resultados para el entorno #3 provenientes de con RR-MACD con cuatro restricciones. Como en este ejemplo, el número de P_i es igual a 6, entonces, el número de criterios de decisión de la Ecuación (4.18) $\rightarrow m = 4$. Por lo tanto, existen cuatro valores de κ y cuatro valores de τ ; es decir, $\Theta_p^* = (J_1(\theta_1) = \kappa_1, J_3(\theta_2) = \kappa_2, J_5(\theta_3) = \kappa_3, J_7(\theta_4) = \kappa_4, J_2(\theta_1) = \tau_1, J_4(\theta_2) = \tau_2, J_6(\theta_3) = \tau_3 \text{ y } J_8(\theta_4) = \tau_4)$, como puede verse en la Figura 23.

Es importante destacar que la interpolación de los segmentos S y L construye un conjunto de curvas suaves 3D, y todas representan posibles soluciones. Por lo tanto, es necesario abordar una etapa de decisión (DM) que seleccione una de ellas; es decir, un punto en el frente de Pareto. En este trabajo se ha utilizado los criterios de selección basados en la menor distancia hasta el punto ideal. Las Figuras 23 y 25 muestran en rojo, el punto seleccionado de $J(\Theta_p^*)$ y Θ_p^* , que ha sido seleccionado utilizando la norma ∞ .

La Figura 24a muestra la construcción de la curva 3D suave $C(t)$, mientras que la Figura 24b muestra la mejor optimización en términos de κ y τ , manteniendo una media matemática baja para estas variables geométricas. Sin embargo, en algunos casos particulares se detecta un aumento debido al cambio de dirección del vuelo. La curva generada por Bézier $B(t)$ se muestra como una línea amarilla, además puede verse una ruta más directa entre el punto de $cInit$ y $cGoal$. Sin embargo, esta curva se acerca al obstáculo del fondo. Para solucionar esto, diferentes autores proponen modificar los puntos de control ρ , o añaden nuevos puntos dentro de los puntos inicialmente definidos.

La Figura 26 muestra un conjunto de cuatro ejemplos adicionales a partir de la Tabla 4.1, donde se representa la funcionalidad del algoritmo. Es importante señalar que el número de ρ fue diferente en cada experimento, así como las altitudes, lo que ga-

rantizó la planificación 3D. También es importante mencionar que el primer entorno mostrado en el literal a de la Figura 26a posee características dimensionales de vuelo menores, por lo que el radio de giro en este ejemplo se fijó en $R_p = 3[\text{m}]$ con una velocidad media de vuelo de 1,7[m/s].

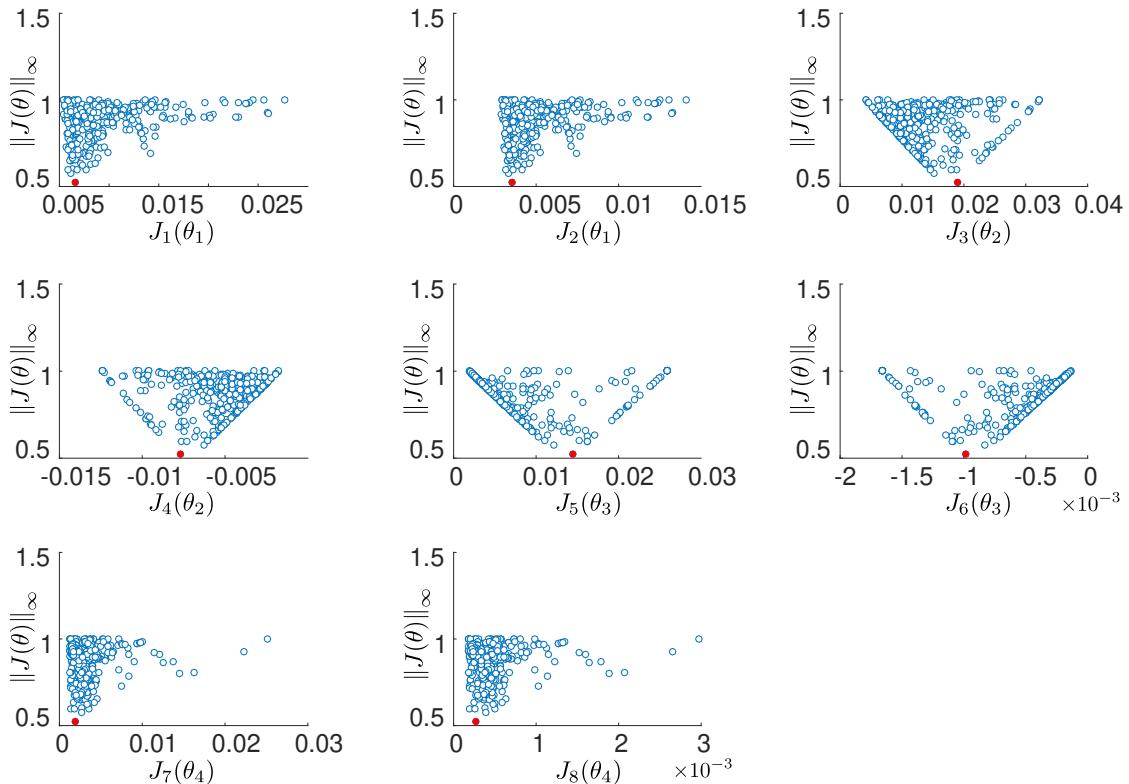


Figura 4.6: Representación del frente de Pareto utilizando norma- ∞ . Los subíndices pares J representan los valores κ en cada S_i , mientras que los subíndices impares J representan los valores τ en S_i . Los J^{ideal} se muestran como círculos rojo sólido.

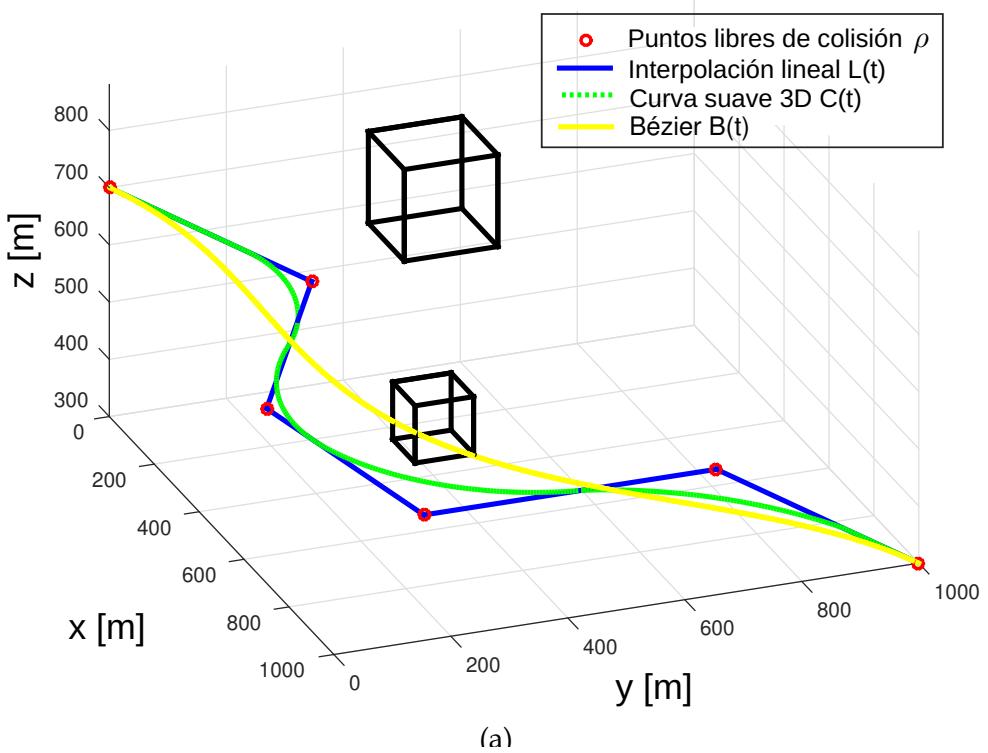
Con el objetivo de realizar una descripción de los diferentes grupos de trayectorias construidas por $L(t)$, $C(t)$ o $B(t)$, en la Figura 26, y en las Tablas 4.2 y 4.3 se muestra los resultados del vuelo desde el punto $cInit$ hasta $cGoal$ en términos de distancias, a partir de uno de los resultados de cada entorno establecido en la Tabla 4.1. Se puede observar que el conjunto de mayores distancias, correspondientes a la trayectoria en forma de línea recta marcada por $L(t)$, $C(t)$ reduce la distancia en $L(t)$. Dado que $B(t)$ hace una aproximación (como expresión matemática) entre el conjunto de ρ de cada entorno, entonces, se contruye trayectorias más corta. La columna “EAA Error (metros)” muestra el error absoluto aproximado $EAA = \frac{1}{n} \sum_{i=1}^n |A - B|$, donde $A = L(t)$ y $B = C(t) \wedge B = B(t)$. Por lo tanto, los resultados de la columna “EAA Error (metros)” muestra una mayor aproximación a $C(t)$, lo que se traduce en una mejor evasión de obstáculos.

Del mismo modo, la Tabla 4.4 muestra un conjunto de resultados de los cinco entornos analizados. Los promedios de κ y τ generados a lo largo de cada curva suave muestran que $B(t)$ supera a $C(t)$. Sin embargo, en el primer entorno se produce una colisión provocada por la curva $B(t)$.

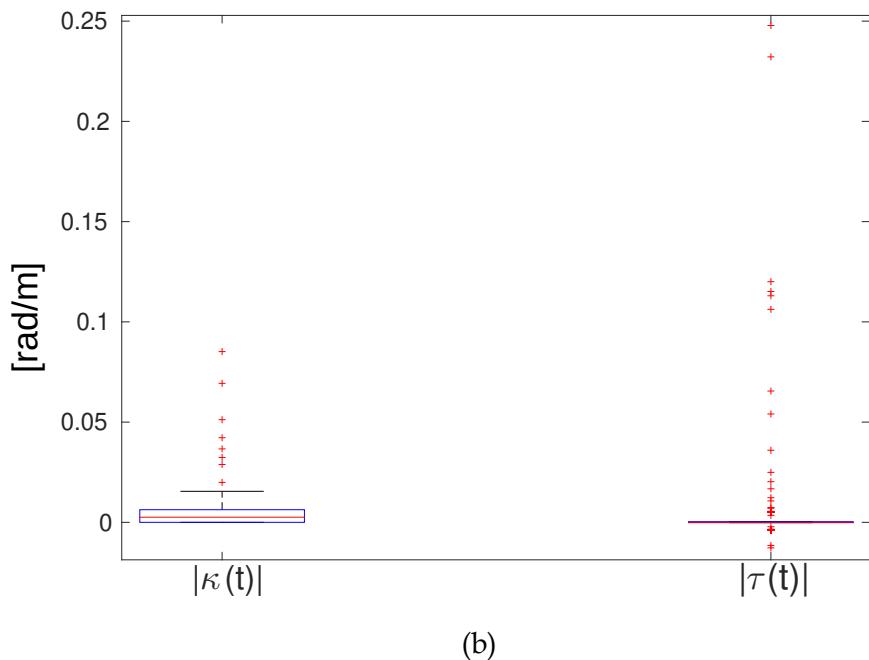
Cuadro 4.2: Distancias de vuelo. Muestra la distancia en me-tros en los puntos de colisión libre, $cInit$ y $cGoal$ marcados con ρ .

	Distancia de vuelo [m]		
	$L(t)$	$C(t)$	$B(t)$
#1	182.929355	174.002834	148.911388
#2	1728.757868	1610.781941	1453.060601
#3	1863.391222	1721.505017	1526.055284
#4	1936.078758	1860.263202	1772.944453
#5	1873.814514	1839.965587	1743.723244

Finalmente, los resultados producidos por la simu-lación de vuelo del UAV Kaddet 2400 realizada por medio de Matlab/Simulink/FlightGear, sobre el entorno #3 se muestran en la Figura 27. Las coordenadas geodésicas de la Figura 27a se expresan en grados decimales. En este ejemplo, el vuelo comienza con una altitud de 500,4 [m], y tras las maniobras realizadas por el UAV, alcanzan una nueva altitud de 603,1 [m]. El literal b de la Figura 27b muestra el modelo del UAV en vuelo.



(a)



(b)

Figura 4.7: Ejemplo de un entorno 3D con obstáculos. (a) Reconstrucción de trayectorias 3D. (b) Medias geométricas de las variables κ y τ de la trayectoria final.

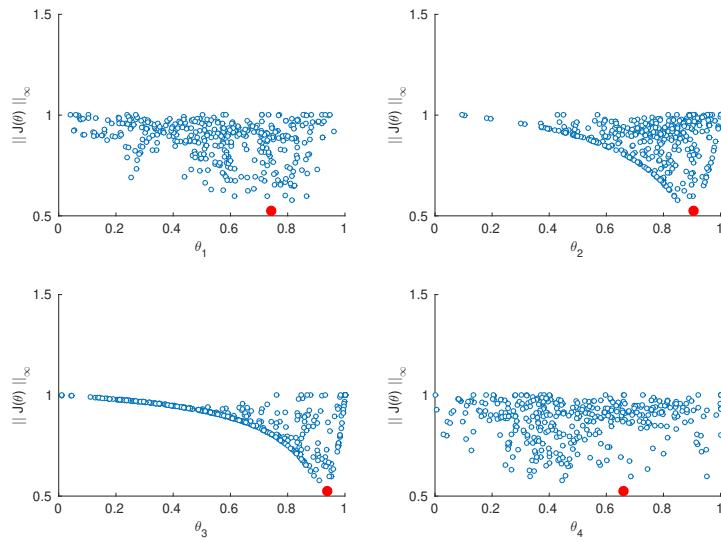


Figura 4.8: Representación de los parámetros óptimos de Pareto. Los objetivos más cercanos a J^{ideal} se muestran en círculos rojos sólidos.

Cuadro 4.3: Error AEA [m]. Muestra el error medio en metros a lo largo de las trayectorias.

Env.	EAA Error [m]	
	$L(t)$ vs $C(t)$	$L(t)$ vs $B(t)$
#1	0.622684	3.248545
#2	17.234613	41.453691
#3	14.600159	56.678212
#4	9.871725	36.617234
#5	9.891240	36.614752

Cuadro 4.4: Resultados promedio de κ y τ a lo largo de las curvas $C(t)$ y $B(t)$. La columna “Colisión” de la trayectoria contra un obstáculo se muestra como positiva (x), o negativa (o).

	<i>Curva</i>	κ	τ	<i>Colisión</i>
#1	$C(t)$	0.157961	0.185973	o
	$B(t)$	0.019513	0.092539	x
#2	$C(t)$	0.007138	0.159732	o
	$B(t)$	0.001082	0.006652	o
#3	$C(t)$	0.004556	0.185806	o
	$B(t)$	0.001068	0.004442	o
#4	$C(t)$	0.003445	0.574121	o
	$B(t)$	0.000812	0.003332	o
#5	$C(t)$	0.004515	0.135183	o
	$B(t)$	0.000643	0.004253	o

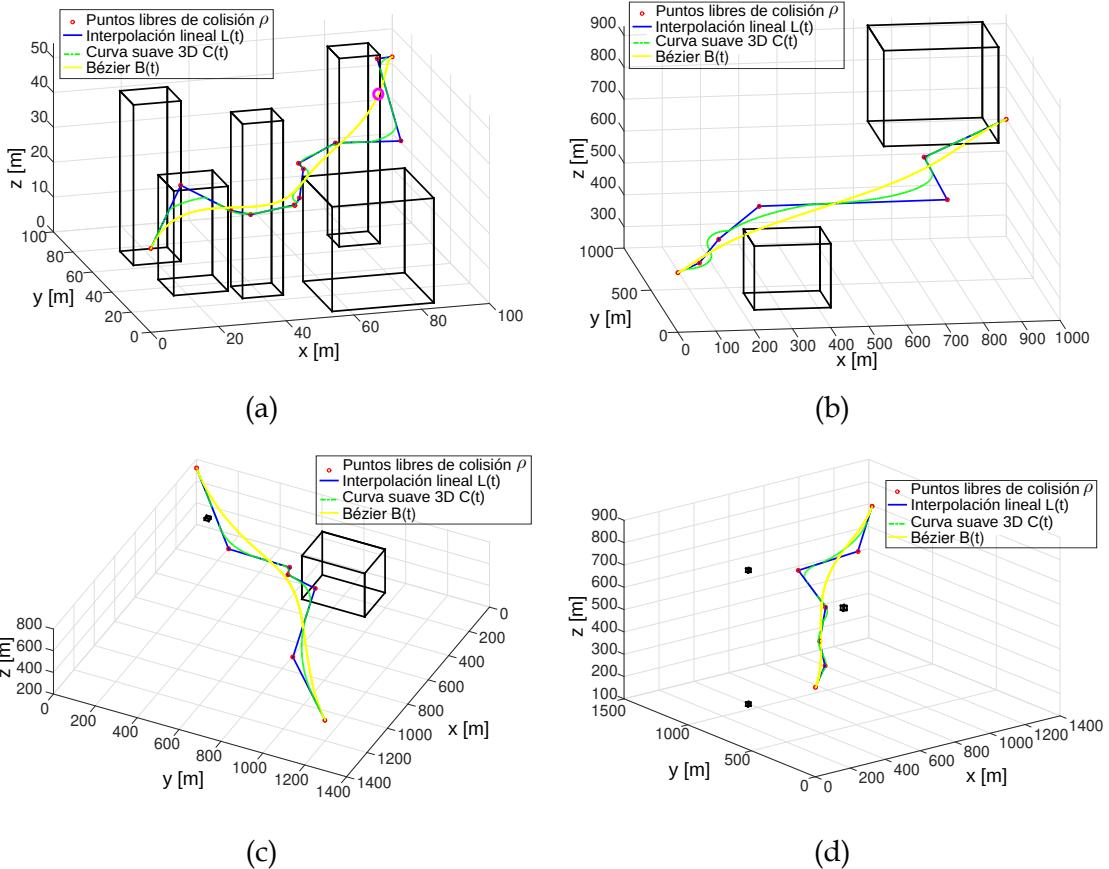


Figura 4.9: Experimentos adicionales de entornos 3D. (a) (Tabla 4.1 Entorno #1.) Representa un entorno desestructurado con diferentes edificios, donde se aprecia una colisión entre $B(t)$ y un edificio (colisión marcada como una circunferencia de color magenta). (b) (Tabla 4.1 Entorno #2.) Entorno 3D con dos obstáculos de diferentes tamaños. (c) (Tabla 4.1 Entorno #4.) Entorno 3D con dos obstáculos de diferentes tamaños. (d) (Tabla 4.1 Entorno #5.) Entorno 3D con tres pequeños obstáculos aéreos.

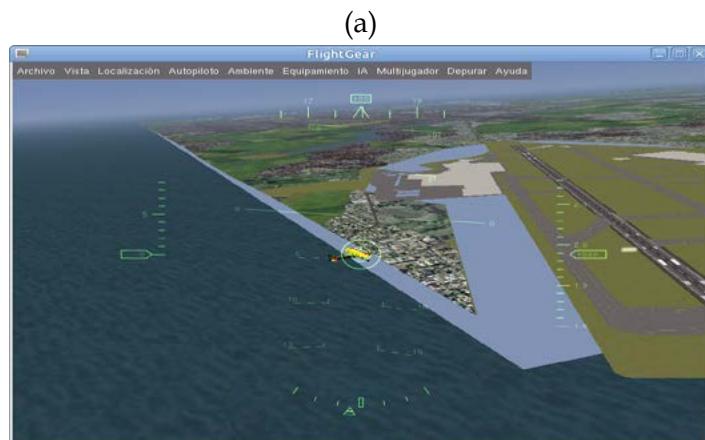
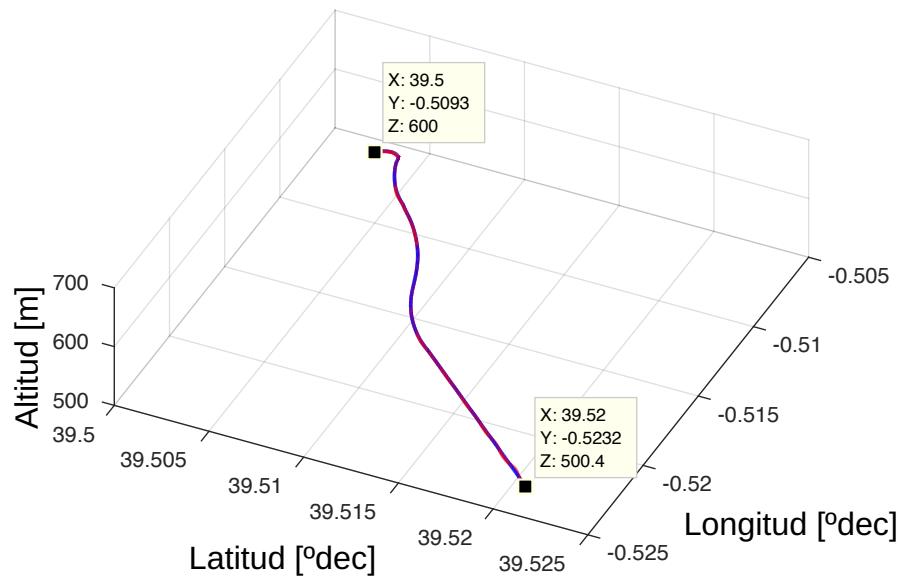


Figura 4.10: Simulación de vuelo. (a) Vuelo del UAV, la línea azul es la trayectoria calculada a partir del proceso descrito y la línea roja es la trayectoria real del UAV. (b) Vista desde la perspectiva de vuelo del simulador.

4.6 Conclusiones y trabajos futuros

En este trabajo se describe un enfoque para la generación de trayectorias suaves continuas 3D, construidas a partir de las limitaciones operativas del vehículo aéreo no tripulado de ala fija.

El capítulo realiza una descripción de la construcción de trayectorias suaves mediante la definición de dos tipos de segmentos. El primer tipo se define como segmentos esféricos S , que garantizan un perfil continuo y de curvatura mínima. El segundo tipo se define como segmentos L y se trata de aquellos que se conectan generalmente con S .

Para obtener los valores numéricos de los parámetros de la trayectoria, se ha planteado la resolución de un MOP, dado que el problema tiene infinitas soluciones factibles. En la resolución del MOP, durante la etapa de DM, es esencial seleccionar el punto deseado del conjunto Pareto de soluciones óptimas.

Es importante recordar que con métodos como las curvas clásicas Bézier o B-splines, se puede definir el número de muestras a lo largo de la trayectoria. Sin embargo, la distancia medida entre un punto y el siguiente no es la misma, de modo que la diferencia puede ser grande. Este tipo de curvas son útiles en entornos relativamente sencillos con pocos obstáculos; sin embargo, a medida que crece el número de obstáculos, los puntos de control aumentan debido a la planificación de la trayectoria. En consecuencia, la construcción de la curva puede provocar colisiones.

El capítulo ha considerado las restricciones cinemáticas del UAV. En este sentido, una consideración importante que puede mejorar la construcción de nuevas trayectorias es incrementar las variables matemáticas como el consumo de energía o los datos incompletos en entornos dinámicos.





Bibliografía

- [1] N. S. J. Liang y S. Srigrarom. «Multi-camera multi-target drone tracking systems with trajectory-based target matching and reidentification». En: *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, jun. de 2021, págs. 1337-1344. doi: [10.1109/ICUAS51884.2021.9476845](https://doi.org/10.1109/ICUAS51884.2021.9476845).
- [2] B. Esakki et al. «Development of 3D-printed floating Quadrotor for collection of algae in remote water bodies». En: *Computers and Electronics in Agriculture* 164 (sep. de 2019), pág. 104891. doi: [10.1016/j.compag.2019.104891](https://doi.org/10.1016/j.compag.2019.104891).
- [3] P. H. Chung, D. M. Ma y J. K. Shiao. «Design, Manufacturing, and Flight Testing of an Experimental Flying Wing UAV». En: *Applied Sciences* 9.15 (2019), pág. 3043. doi: [10.3390/app9153043](https://doi.org/10.3390/app9153043).
- [4] F. Yuan et al. «SoftMP: Attentive feature pooling for joint local feature detection and description for place recognition in changing environments». En: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, mayo de 2021, págs. 5847-5853. doi: [10.1109/ICRA48506.2021.9562087](https://doi.org/10.1109/ICRA48506.2021.9562087).
- [5] M. Zhou et al. «Review of vertical take-off and landing fixed-wing UAV and its application prospect in precision agriculture». En: *International Journal of Precision Agricultural Aviation* 1.1 (2018), págs. 8-17. doi: [10.33440/j.ijpaa.20200304.130](https://doi.org/10.33440/j.ijpaa.20200304.130).
- [6] Luqman Hakim, Farhan Ahmed y Muhammad Zubair Khan. «Survey on path planning algorithms for UAVs: Optimality, completeness, and performance». En: *IEEE Access* 9 (2021), págs. 134935-134956. doi: [10.1109/ACCESS.2021.3116118](https://doi.org/10.1109/ACCESS.2021.3116118).

- [7] Y. Zhang et al. «Design and Control of an Ultra-Low-Cost Logistic Delivery Fixed-Wing UAV». En: *Applied Sciences* 14.11 (mayo de 2024), pág. 4358. issn: 2076-3417. doi: 10.3390/app14114358.
- [8] H. Gu et al. «Development and experimental verification of a hybrid vertical take-off and landing (VTOL) unmanned aerial vehicle (UAV)». En: *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, jun. de 2017, págs. 160-169. doi: 10.1109/ICUAS.2017.7991420.
- [9] Vishwesh Chipade y Dimitra Panagou. «A sampling-based algorithm for motion planning with temporal logic specifications». En: *2018 IEEE Conference on Decision and Control (CDC)*. 2018, págs. 3706-3713. doi: 10.1109/CDC.2018.8619664.
- [10] Tiago Oliveira, Marco Santos y Pedro Oliveira. «Multi-objective UAV path planning using evolutionary algorithms». En: *Applied Soft Computing* 107 (2021), pág. 107393. doi: 10.1016/j.asoc.2021.107393.
- [11] Steven M. LaValle. *Planning Algorithms*. Cambridge, UK: Cambridge University Press, 2006. isbn: 978-0521862059. url: <http://lavalle.pl/planning/>.
- [12] Zhenyu Li, Xuefeng Liu y Dong Sun. «An improved RRT algorithm for UAV path planning in 3D complex environments». En: *Journal of Intelligent and Robotic Systems* 77.2 (2015), págs. 369-384. doi: 10.1007/s10846-014-0130-7.
- [13] J. Lévine. «Flatness and Motion Planning». En: *Analysis and Control of Nonlinear Systems: A Flatness-based Approach*. Springer, 2009, págs. 181-191.
- [14] Euclid. *The Thirteen Books of Euclid's Elements*. Ed. por Thomas L. Heath. Reprint of the 1908 edition. Dover Publications, 1956.
- [15] D. Hilbert. *Foundations of Geometry*. Open Court, 1999.
- [16] Y. Bertot y F. Guilhot. «Formal Verification of Geometric Theorems Using Coq». En: *Mathematics in Computer Science* 2 (2008), págs. 1-20. doi: 10.1007/s11786-008-0051-9.
- [17] K. M. Lynch y F. C. Park. *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, 2017.

- [18] S. M. LaValle. «Rapidly-Exploring Random Trees: A New Tool for Path Planning». En: *The International Journal of Robotics Research* 20.4 (2001), págs. 286-303. doi: 10.1177/02783640122067653.
- [19] Roger W. Brockett. «Asymptotic stability and feedback stabilization». En: *Differential Geometric Control Theory* (1983), págs. 181-191.
- [20] Christos G. Cassandras y Stéphane Lafortune. *Introduction to Discrete Event Systems*. 2nd. Springer, 2008.
- [21] Rajeev Alur y David L. Dill. «A theory of timed automata». En: *Theoretical Computer Science*. Vol. 126. 2. 1995, págs. 183-235.
- [22] E. A. Lee y S. A. Seshia. «Foundations of Cyber-Physical Systems». En: *Proceedings of the IEEE* 104.5 (2016), págs. 899-911. doi: 10.1109/JPROC.2015.2511545.
- [23] Stephen A. Cook. «The complexity of theorem-proving procedures». En: *Proceedings of the Third Annual ACM Symposium on Theory of Computing* (1971), págs. 151-158. doi: 10.1145/800157.805047.
- [24] Michael R. Garey y David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- [25] Richard M. Karp. «Reducibility among combinatorial problems». En: *Complexity of Computer Computations*. Ed. por Raymond E. Miller y James W. Thatcher. Springer, 1972, págs. 85-103. doi: 10.1007/978-1-4684-2001-2_9.
- [26] Bernard Chazelle. «Approximation and decomposition of shapes». En: *Algorithmica* 2.1 (1987), págs. 133-162. doi: 10.1007/BF01840360.
- [27] Lydia E. Kavraki y Jean-Claude Latombe. «Probabilistic roadmaps for robot path planning». En: *Practical Motion Planning in Robotics: Current Approaches and Future Directions* (1998), págs. 33-53.
- [28] Mark de Berg et al. *Computational Geometry: Algorithms and Applications*. 3rd. Springer, 2008.
- [29] Jean-Claude Latombe. *Robot Motion Planning*. Springer, 1991.
- [30] Bernard Chazelle y David P. Dobkin. «Optimal convex decompositions». En: *Computational Geometry* 15.1 (1985), págs. 63-69.

- [31] Edsger W. Dijkstra. «A note on two problems in connexion with graphs». En: *Numerische Mathematik* 1 (1959), págs. 269-271. doi: 10.1007/BF01386390.
- [32] Lydia E. Kavraki et al. «Probabilistic roadmaps for path planning in high-dimensional configuration spaces». En: *IEEE Transactions on Robotics and Automation* 12.4 (1996), págs. 566-580. doi: 10.1109/70.508439.
- [33] Marius Kloetzer y Calin Belta. «A fully automated framework for control of linear systems from temporal logic specifications». En: *IEEE Transactions on Automatic Control* 53.1 (2008), págs. 287-297. doi: 10.1109/TAC.2007.914982.
- [34] Francesco Bullo y Andrew D. Lewis. *Geometric Control of Mechanical Systems*. Springer, 2004.
- [35] Nathan Ratliff et al. «CHOMP: Gradient optimization techniques for efficient motion planning». En: *IEEE International Conference on Robotics and Automation (ICRA)*. 2009, págs. 489-494. doi: 10.1109/ROBOT.2009.5152817.
- [36] Sertac Karaman y Emilio Frazzoli. «Sampling-based algorithms for optimal motion planning». En: *International Journal of Robotics Research*. Vol. 30. 7. 2011, págs. 846-894. doi: 10.1177/0278364911406761.
- [37] Mike Phillips y Maxim Likhachev. «SIPP: Safe interval path planning for dynamic environments». En: *IEEE International Conference on Robotics and Automation (ICRA)*. 2011, págs. 5628-5635. doi: 10.1109/ICRA.2011.5980471.
- [38] Nathan Ratliff et al. «Learning to search: Functional gradient techniques for imitation learning». En: *Autonomous Robots* 37.1 (2014), págs. 25-53. doi: 10.1007/s10514-014-9403-9.
- [39] Jonathan D. Gammell, Siddhartha S. Srinivasa y Timothy D. Barfoot. «Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic». En: *IEEE Transactions on Robotics* 34.4 (2018), págs. 997-1012. doi: 10.1109/TR.2018.2830331.
- [40] Anthony M. Bloch. *Nonholonomic Mechanics and Control*. Springer, 2003.
- [41] Carlos Canudas de Wit, Bruno Siciliano y Georges Bastin. «Theory of Robot Control». En: *Springer* (1993).

- [42] Florent Lamiraux y Jean-Paul Laumond. «Motion planning for a car-like robot: A case study». En: *IEEE International Conference on Robotics and Automation (ICRA)*. 1999, págs. 583-588. doi: 10.1109/ROBOT.1999.770034.
- [43] Z Kong C. Goerzen et al. «A Survey of Motion Planning Algorithms from the Perspective of Autonomous UAV Guidance». En: *Journal of Intelligent Robots and Systems* 57.1-4 (2010), págs. 65-100. issn: 09210296. doi: 10.1007/s10846-009-9383-1.
- [44] P.B. Sujit, Srikanth Saripalli y Joao Borges Sousa. «Unmanned Aerial Vehicle Path Following: A Survey and Analysis of Algorithms for Fixed-Wing Unmanned Aerial Vehicles». En: *IEEE Control Systems* 34.1 (2014), págs. 42-59. issn: 1066-033X. doi: 10.1109/MCS.2013.2287568.
- [45] Ee-May Kan et al. «Contour Based Path Planning with B-Spline Trajectory Generation for Unmanned Aerial Vehicles (UAVs) over Hostile Terrain». En: *Journal of Intelligent Learning Systems and Applications* 03.03 (2011), págs. 122-130. issn: 2150-8402. doi: 10.4236/jilsa.2011.33014.
- [46] Dongwon Jung y Panagiotis Tsotras. «On-Line Path Generation for Unmanned Aerial Vehicles Using B-Spline Path Templates». En: *Journal of Guidance, Control, and Dynamics* 36.6 (nov. de 2013), págs. 1642-1653. issn: 0731-5090. doi: 10.2514/1.60780.
- [47] Xiaoliang Wang et al. «Curvature continuous and bounded path planning for fixed-wing UAVs». En: *Sensors (Switzerland)* 17.9 (2017), págs. 1-21. issn: 1424-8220. doi: 10.3390/s170\92155.
- [48] D. S. Meek y D. J. Walton. «A note on finding clothoids». En: *Journal of Computational and Applied Mathematics* 171.1-2 (2004), págs. 433-453. issn: 03770427. doi: 10.1016/j.cam.2003.12.047.
- [49] H. Marzbani, Reza N. Jazar y M. Fard. *Better Road Design Using Clothoids*. Springer, Cham, 2015, págs. 25-40. doi: 10.1007/978-3-319-17999-5_3.
- [50] Vicent Girbés, Leopoldo Armesto y Josep Tornero. «Path Following Hybrid Control for Vehicle Stability Applied to Industrial Forklifts». En: *Robotics and Autonomous Systems* 62.6 (2014), págs. 910-922.

- [51] Hormoz Marzbani et al. *Better Road Design for Autonomous Vehicles Using Clothoids*. Springer, Cham, 2015, págs. 265-278. doi: 10.1007/978-3-319-19830-9_24.
- [52] Suhyeon Gim et al. «Clothoids Composition Method for Smooth Path Generation of Car-Like Vehicle Navigation». En: *Journal of Intelligent and Robotic Systems: Theory and Applications* 88.1 (2017), págs. 129-146. issn: 15730409. doi: 10.1007/s10846-017-0531-8.
- [53] Klaus D. Mielenz. «Computation of Fresnel Integrals II». En: *Journal of Research of the National Institute of Standards and Technology* 105.4 (2000), págs. 589-590. issn: 1044677X. doi: 10.6028/jres.105.049.
- [54] N Montés et al. «Real-time clothoid approximation by Rational Bezier curves». En: *IEEE Int. Conf. on Robotics and Automation*. 2008, págs. 2246-2251. doi: 10.1109/ROBOT.2008.4543548.
- [55] M Brezak e I Petrovic. «Real-time Approximation of Clothoids With Bounded Error for Path Planning Applications». En: *Robotics, IEEE Transactions on* 30.2 (2014), págs. 507-515. issn: 1552-3098. doi: 10.1109/TR0.2013.2283928.
- [56] Smita Narayan. «Approximating Cornu spirals by arc splines». En: *Journal of Computational and Applied Mathematics* 255 (), págs. 789-804. issn: 03770427. doi: 10.1016/j.cam.2013.06.038.
- [57] Yong Chen et al. «Accurate and Efficient Approximation of Clothoids Using Bézier Curves for Path Planning». En: *IEEE Transactions on Robotics* 33.5 (2017), págs. 1242-1247. issn: 1552 3098. doi: 10.1109/TR0.2017.2699670.
- [58] Thomas. Banchoff y Stephen (Stephen T.) Lovett. *Differential geometry of curves and surfaces*. A.K. Peters, 2010, pág. 331. isbn: 97815688 14568.
- [59] R Levien. *Technical Report UCB/EECS-2008-111, The Euler spiral: a mathematical history*. Inf. téc. EECS Department, University of California, Berkeley, 2008.
- [60] Gur Harary y Ayellet Tal. «3D Euler spirals for 3D curve completion». En: *Computational Geometry: Theory and Applications* 45.3 (abr. de 2012), págs. 115-126. issn: 09257721. doi: 10.1016/j.comgeo.2011.10.001.
- [61] William J Gordon y Richard F Riesenfeld. «Bernstein -Bézier methods for the computer-aided design of free-form curves and surfaces». En: *Journal of the ACM (JACM)* 21.2 (1974), págs. 293-310.

- [62] A Segovia et al. «Comparative study of the different methods of path generation for a mobile robot in a free environment». En: *Advanced Robotics, 1991.'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*. IEEE. 1991, págs. 1667-1670.
- [63] Nicolas Montes, Marta C Mora y Josep Tornero. «Trajectory generation based on rational bezier curves as clothoids». En: *Intelligent Vehicles Symposium, 2007 IEEE*. IEEE. 2007, págs. 505-510.
- [64] Widyawardana Adiprawita, Adang Suwandi Ahmad y Jaka Sembiring. «Development of ahrs (attitude and heading reference system) for autonomous uav (unmanned aerial vehicle)». En: (2012).
- [65] William H Press. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.
- [66] V Girbes, L Armesto y J Tornero. «On Generating Continuous-Curvature Paths for Line Following Problem with Curvature and Sharpness Constraints». En: *IEEE International Conference on Robotics and Automation*. Mayo de 2011, págs. 6156-6161.
- [67] J Velasco-Carrau et al. «Multi-objective optimization for wind estimation and aircraft model identification». En: *Journal of Guidance, Control, and Dynamics* 39.2 (2015), págs. 372-389.
- [68] K. H. Knuth, R. M. Powell y P. A. Reali. «Estimating flight characteristics of anomalous unidentified aerial vehicles». En: *Entropy* 21.10 (2019), pág. 939.
- [69] J. Tao, S. Guo e Y. Shang. «An Effective Strategy for Collision Avoidance of Multiple UAVs with Unknown Acceleration». En: *IEEE Access* 11 (2023), págs. 112600-112619.
- [70] Nathaniel Melchior y Reid Simmons. «Particle RRT for path planning». En: *Proceedings of the IEEE International Conference on Robotics and Automation*. 2007, págs. 1617-1624. doi: 10.1109/ROBOT.2007.363600.
- [71] Steven M. LaValle y James J. Kuffner. «Rapidly-Exploring Random Trees: A New Tool for Path Planning». En: *Computer Science Department Technical Report TR98-11, Iowa State University* (1998).

- [72] Steven M. LaValle y James J. Kuffner Jr. «Rapidly-exploring random trees: Progress and prospects». En: *Proceedings of the 4th International Workshop on Algorithmic Foundations of Robotics*. Cambridge: Cambridge University Press, 2000, págs. 293-308.
- [73] Wei Zhang, Cheng Li y Tao Wang. «Real-time trajectory planning for UAVs in dynamic environments». En: *Electronics* 8.3 (2019), pág. 306. doi: 10.3390/electronics8030306.
- [74] Gunilla Borgefors. «Distance transformations in digital images». En: *Computer Vision and Image Understanding* 64.3 (2006), págs. 368-376. doi: 10.1006/cviu.1996.0069.
- [75] Hanan Samet. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann, 2002.
- [76] Eric W. Justh y Perinkulam S. Krishnaprasad. «A simple control law for UAV formation flying». En: *Proceedings of the 41st IEEE Conference on Decision and Control*. Vol. 2. 2002, págs. 1534-1538. doi: 10.1109/CDC.2002.1184713.
- [77] Isaac J. Schoenberg. «Contributions to the Problem of Approximation of Equidistant Data by Analytic Functions». En: *Quarterly of Applied Mathematics* 4.1 (1988), págs. 45-99.
- [78] Eric W. Weisstein. *Euler Angles*. From MathWorld—A Wolfram Web Resource. <https://mathworld.wolfram.com/EulerAngles.html>. 2009.
- [79] James Diebel. *Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors*. Inf. téc. Technical Report. Stanford University, 2006.
- [80] Laurence C. Young. *Lectures on the Calculus of Variations and Optimal Control Theory*. American Mathematical Society, 2006.
- [81] Victor A. Toponogov. *Differential Geometry of Curves and Surfaces*. Springer, 2006.
- [82] L. Yang et al. «Survey of Robot 3D Path Planning Algorithms». En: *Journal of Control Science and Engineering* (2016), págs. 1-22. doi: 10.1155/2016/7426913.

- [83] I. J. Schoenberg. «Contributions to the problem of approximation of equidistant data by analytic functions. Part B. On the problem of osculatory interpolation. A second class of analytic approximation formulae». En: *Quarterly of Applied Mathematics* 4.2 (1946), págs. 112-141.
- [84] Carl Runge. «Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten». En: *Zeitschrift für Mathematik und Physik* 46 (1901), págs. 224-243.
- [85] Fumihiko Yamaguchi. «Curves and Surfaces in Computer Aided Geometric Design». En: *Springer Series in Geometry and Computing* (2012). doi: 10.1007/978-3-642-31057-4.
- [86] Ian Chivers y Jane Sleightholme. *Introduction to Programming with Fortran*. Springer, 2015. doi: 10.1007/978-3-319-20188-7.
- [87] Kwangjin Yang y Salah Sukkarieh. «3D smooth path planning for a UAV in cluttered environments». En: *IEEE International Conference on Robotics and Automation* (2008), págs. 3050-3055. doi: 10.1109/ROBOT.2008.4543641.
- [88] Curtis Perry y Jim Hall. *FlightGear: Open Source Flight Simulator*. Disponible en: <https://home.flightgear.org>. 2004.
- [89] Sebastian Münzer, Christoph Holscher y Michael Brosamle. «Planning of motion trajectories in complex environments». En: *Journal of Experimental Psychology: Human Perception and Performance* 32.3 (2006), págs. 663-679. doi: 10.1037/0096-1523.32.3.663.
- [90] José Morales, Santiago Martínez y Rafael Fierro. «UAV path planning based on RRT algorithms». En: *Journal of Intelligent and Robotic Systems* 69.1 (2013), págs. 273-293. doi: 10.1007/s10846-012-9763-9.
- [91] E. Krotkov y M. Heber. «Mapping and positioning for a prototype lunar rover». En: *Proceedings of 1995 IEEE International Conference on Robotics and Automation, Vol. 3*. IEEE, 1995, págs. 2913-2919. doi: 10.1109/ROBOT.1995.525697.
- [92] Imad M. Abbadi. «A framework for establishing trust in cloud computing». En: *International Journal of Cloud Computing and Services Science* 1.2 (2012), págs. 115-128.

- [93] Carlos Aguilar-Ibáñez y Ángel J. Muñoz-Vázquez. «Robust control for UAV trajectory tracking». En: *International Journal of Control, Automation and Systems* 14.4 (2016), págs. 995-1006. doi: 10.1007/s12555-015-0184-2.
- [94] Carlos Aguilar-Ibáñez, Ángel Jesús Muñoz-Vázquez y Vicente Parra-Vega. «A robust linear time-invariant control for quadrotor trajectory tracking». En: *IEEE Transactions on Control Systems Technology* 25.2 (2017), págs. 545-552. doi: 10.1109/TCST.2016.2561263.
- [95] Xin Yao y Yuhui Xu. «Evolutionary Programming Made Faster». En: *Proceedings of the IEEE Transactions on Evolutionary Computation*. Vol. 3. 2. 2015, págs. 82-102. doi: 10.1109/4235.771163.
- [96] Zheng Yan, Nicolas Jouandieu y Adeline A. Cherif. «A survey and analysis of multi-robot coordination». En: *International Journal of Advanced Robotic Systems* 10.399 (2013). doi: 10.5772/57313.
- [97] Siang Yeh y Kuo-Hua Lee. «Bezier curve based path planning for UAVs». En: *International Journal of Aerospace Engineering* 2012 (2012), págs. 1-8. doi: 10.1155/2012/635980.
- [98] John Denny y Balasubramanian Manohar. «Smooth trajectory generation for UAVs in complex environments». En: *IEEE International Conference on Robotics and Automation (ICRA)*. 2013, págs. 4939-4944. doi: 10.1109/ICRA.2013.6631261.
- [99] Shiyu Li, Yu Ding y Dong Sun. «3D smooth trajectory planning for unmanned aerial vehicles». En: *Journal of Intelligent and Robotic Systems* 73 (2014), págs. 323-340. doi: 10.1007/s10846-013-9913-5.
- [100] F. Ortiz, R. Flores y R. Lozano. «Path planning for fixed-wing UAVs using B-splines». En: *International Journal of Robust and Nonlinear Control* 25.4 (2015), págs. 562-580. doi: 10.1002/rnc.3147.
- [101] Peng Cheng, James Keller y Vijay Kumar. «Time-optimal UAV trajectory planning for 3D urban structure coverage». En: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS* (2008), págs. 2750-2757. issn: 2153-0858. doi: 10.1109/IROS.2008.4650988.

- [102] Y. Liang y L. Xu. «Global path planning for mobile robot based genetic algorithm and modified simulated annealing algorithm». En: *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*. ACM, 2009, págs. 303-308. doi: 10.1145/1543834.1543875.
- [103] J. Liu et al. «An improved ant colony algorithm for robot path planning». En: *Soft Computing* 21.19 (2017), págs. 5829-5839. doi: 10.1007/s00500-016-2161-7.
- [104] H. Cao et al. «Visualized trajectory planning of flexible redundant robotic arm using a novel hybrid algorithm». En: *Optik* 127.20 (2016), págs. 9974-9983. doi: 10.1016/j.ijleo.2016.07.078.
- [105] H. Duan y P. Qiao. «Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning». En: *International Journal of Intelligent Computing and Cybernetics* 7.1 (2014), págs. 24-37. doi: 10.1108/ijicc.2009.39802aaa.001.
- [106] Mohammad Alhanjouri y Ahmad Baroudi. «Genetic algorithm-based path planning for UAVs». En: *International Journal of Computer Applications* 40.16 (2011), págs. 37-43. doi: 10.5120/5152-7222.
- [107] Ahmed Bakdi, Nabil Mesbahi y Mohamed El Amine Lalami. «Optimal path planning for UAVs using genetic algorithms». En: *International Journal of Advanced Robotic Systems* 14.1 (2017). doi: 10.1177/1729881416684012.
- [108] Gyanendra Pandey, Rajiv Sonkar y Shivam Bharti. «Path planning for UAVs using artificial bee colony optimization». En: *International Conference on Communication and Signal Processing*. 2017, págs. 2083-2087. doi: 10.1109/ICCPSP.2017.8286704.
- [109] Dorina Thanou y Pascal Frossard. «Multi-objective path planning for aerial robotics». En: *IEEE Transactions on Robotics* 30.2 (2014), págs. 354-366. doi: 10.1109/TRO.2013.2293886.
- [110] Q. Yaohong, Y. Zhang e Y. Zhang. «Optimal flight path planning for UAVs in 3-D threat environment». En: *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, mayo de 2014, págs. 149-155. doi: 10.1109/ICUAS.2014.6842250.

- [111] Zongying Fang, Shaoping Wang y Zhe Chen. «Smooth trajectory generation for UAVs via optimal control». En: *Aerospace Science and Technology* 66 (2017), págs. 1-12. doi: 10.1016/j.ast.2017.02.015.
- [112] Dwi Khuswendi, Widyawardana Adiprawita y Andreas Wibowo. «3D path planning for UAV using artificial potential fields». En: *Proceedings of the International Conference on Electrical Engineering and Informatics*. 2011, págs. 1-6. doi: 10.1109/ICEEI.2011.6021761.
- [113] X. Chen y J. Zhang. «The Three-Dimension Path Planning of UAV Based on Improved Artificial Potential Field in Dynamic Environment». En: *2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics*, Vol. 2. IEEE, ago. de 2013, págs. 144-147. doi: 10.1109/IHMSC.2013.181.
- [114] D. M. Rivera, F. Augusto Preto y R. Ramirez. «Trajectory Planning for UAVs in 3D Environments Using a Moving Band in Potential Sigmoid Fields». En: *2012 Brazilian Robotics Symposium and Latin American Robotics Symposium*. IEEE, oct. de 2012, págs. 115-119. doi: 10.1109/SBR-LARS.2012.26.
- [115] L. Lifen et al. «Path planning for UAVs based on improved artificial potential field method through changing the repulsive potential function». En: *2016 IEEE Chinese Guidance, Navigation and Control Conference*. IEEE, 2016, págs. 2011-2015.
- [116] F. Samaniego et al. «Comparative Study of 3-Dimensional Path Planning Methods Constrained by the Maneuverability of Unmanned Aerial Vehicles». En: *2018 7th International Conference on Systems and Control*. ICSC, 2018, págs. 13-20. doi: 10.1109/ICoSC.2018.8587810.
- [117] V. Herrero, J. Mora y R. García. «Trajectory generation for UAVs in 3D environments». En: *IEEE Aerospace Conference* (2007), págs. 1-10. doi: 10.1109/AERO.2007.352788.
- [118] Jin Yaochu y Zhang Mengjie. *Evolutionary Computation in Combinatorial Optimization*. Springer, 2008. isbn: 978-3540782852.
- [119] E. Abbena, S. Salamon y A. Gray. *Modern Differential Geometry of Curves and Surfaces with Mathematica*. Chapman y Hall/CRC, 2017.

- [120] M. Laumanns et al. «Combining Convergence and Diversity in Evolutionary Multiobjective Optimization». En: *Evolutionary Computation* 10.3 (sep. de 2002), págs. 263-282. doi: 10.1162/106365602760234108.
- [121] C. A. Coello Coello, G. Toscano Pulido y E. Mezura Montes. «Current and Future Research Trends in Evolutionary Multiobjective Optimization». En: *Information Processing with Evolutionary Algorithms*. Springer-Verlag, 2005, págs. 213-231. doi: 10.1007/1-84628-117-2_15.
- [122] F. Samaniego et al. «Recursive Rewarding Modified Adaptive Cell Decomposition (RR-MACD): A Dynamic Path Planning Algorithm for UAVs». En: *Electronics* 8.3 (2019), pág. 306.
- [123] J. Velasco-Carrau et al. «Multi-Objective Optimization for Wind Estimation and Aircraft Model Identification». En: *Journal of Guidance, Control, and Dynamics* 39.2 (feb. de 2016), págs. 372-389. doi: 10.2514/1.G001294.
- [124] Xavier Blasco et al. «A new graphical visualization of N-dimensional Pareto front for decision-making in multiobjective optimization». En: *Information Sciences* 178.20 (2008), págs. 3908-3924. doi: 10.1016/j.ins.2008.05.018.

cedia

El sello editorial de la Corporación Ecuatoriana para el Desarrollo de la Investigación y la Academia - CEDIA, nace con la finalidad de apoyar a la creación y la publicación de resultados, investigaciones y procesos académicos, que fomenten el desarrollo de la ciencia y la innovación a nivel nacional e internacional.

cedia | Editorial

