



Universidad  
Rey Juan Carlos

GRADO EN INGENIERÍA AEROESPACIAL EN  
AERONAVEGACIÓN

Curso Académico 2021/2022

Trabajo Fin de Grado

ESTUDIO DE LA VIABILIDAD DE TÉCNICAS DE  
REGRESIÓN DISPERSA (*SPARSE*) EN  
MODELOS DE TRAYECTORIAS DE AERONAVES

Autor : Meritxell Díaz Coque

Tutor : Pablo Solano López



*Dedicado a  
mi mejor amiga (mi madre)  
y a mi pareja*



# Agradecimientos

En primer lugar quiero dar las gracias por el apoyo incondicional de toda una vida a mi mejor amiga, mi madre. Gracias a ella estudié esta carrera, no solo porque sin ella no sabría que existe este doble grado, sino porque me ha ayudado a superar los momentos más difíciles en el camino. Con su ayuda siempre logro todo lo que me propongo.

A mi gato, Pepe, por hacerme ver que es bueno desconectar de vez en cuando y por estar literalmente en todo momento a mi lado mientras realizaba este proyecto.

A mi pareja, Víctor, por apoyarme cada día desde que nos conocimos. Por estar en todo momento cuando las cosas van bien y no tan bien, por ser esa balsa de tranquilidad que siempre me escucha y, en concreto en este proyecto, por aconsejarme sobre cómo mejorar los casos prácticos y ayudarme durante varias tardes a leer esta memoria para que esté todo perfecto. Gracias a sus consejos he aprendido mucho.

También quiero dar las gracias a la familia de mi pareja, en especial a sus padres. Me han tratado desde el primer momento como una más de la familia y me han acogido en su casa en esos huecos de varias horas de mi horario.

Por último, a mi tutor Pablo, por su tiempo y paciencia en responderme dudas y soluciones cuando me atascaba.



# Resumen

Comprender la dinámica del mundo que nos rodea es una tarea compleja para la cual a veces no disponemos de las herramientas adecuadas, haciendo necesario buscar otros caminos. En concreto, es muy relevante conocer la evolución temporal y las relaciones entre las variables de interés de sistemas complejos y dinámicos, como la propagación de un virus, los impulsos eléctricos del cerebro o el comportamiento de un cohete al ejecutar la reentrada en la atmósfera.

Por ello, en este proyecto se analizarán las técnicas de regresión dispersa para la predicción de modelos del vuelo de una aeronave mediante datos de sus trayectorias. Para ello, se utilizará el algoritmo SINDy a través de su implementación en Python en el paquete PySINDy.

Se ha planteado un estudio de complejidad incremental: se parte de modelos de juguete (*toy models*) como son las ecuaciones dinámicas que generan atractores extraños. Posteriormente, se aplican estas herramientas sobre casos simplificados de la dinámica de la aeronave. Estos casos, aunque de carácter más académico que realista, permiten representar el movimiento principal de una aeronave y por lo tanto analizar la respuesta de estos algoritmos.

El estudio se centrará en determinar la precisión de los modelos matemáticos obtenidos por el algoritmo de regresión dispersa ante las ecuaciones dinámicas de una aeronave en determinadas configuraciones de vuelo. Para ello, se definen unas métricas para medir el error de las trayectorias simuladas y se varían las entradas a dicho algoritmo para obtener así la viabilidad de su aplicación en trayectorias de este tipo.



# Summary

Understanding the dynamics of the world around us is a complex task for which sometimes we do not have the appropriate tools, making it necessary to look for other techniques. In particular, it is very important to know the temporal evolution and the relationships between the variables of interest of complex dynamic systems, such as the propagation of a virus, the electrical impulses of the brain or the behavior of a rocket when re-entering the atmosphere.

Therefore, this project will analyze sparse regression techniques for the prediction of aircraft flight models using data from their trajectories. For this purpose, the SINDy algorithm will be used through its implementation in Python in the PySINDy package.

An study with incremental complexity has been proposed: we start from toy models such as the dynamic equations that generate strange attractors. Subsequently, these tools are applied to simplified cases of aircraft dynamics. These cases, although more academic than realistic, allow to represent the main motion of an aircraft and therefore to analyze the results of these algorithms.

The study will focus on determining the accuracy of the mathematical models obtained by the sparse regression algorithm to the dynamic equations of an aircraft in certain flight configurations. For this purpose, the error of the simulated trajectories and the error in the coefficients of the equation are measured. Finally, the inputs to the algorithm are varied to obtain the feasibility of its application in this type of trajectories.



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	3
1.1.1. Big Data y ciencia de datos . . . . .	4
1.2. Objetivos . . . . .	6
1.3. Planificación temporal . . . . .	7
1.4. Estructura de la memoria . . . . .	8
<b>2. Marco teórico</b>	<b>11</b>
2.1. Sistemas dinámicos . . . . .	11
2.1.1. Atractores . . . . .	15
2.2. Ecuaciones de la mecánica de vuelo . . . . .	18
2.2.1. Sistema de referencia y ecuaciones del movimiento . . . . .	20
<b>3. Sparse Identification of Nonlinear Dynamics (SINDy)</b>	<b>25</b>
3.1. Aplicaciones sobre atractores . . . . .	31
<b>4. Arquitectura general</b>	<b>39</b>
<b>5. Resultados</b>	<b>45</b>
5.1. Caso A: Planeador en descenso uniforme . . . . .	47
5.2. Caso B: Vuelo rectilíneo simétrico para velocidades grandes . . . . .	49
5.3. Caso C: Vuelo rectilíneo simétrico para velocidades pequeñas . . . . .	53
5.3.1. Caso C-1: Vuelo rectilíneo simétrico para velocidades pequeñas . . .	53
5.3.2. Caso C-2: Aproximación en serie de Taylor de un vuelo rectilíneo simétrico para velocidades pequeñas . . . . .	57

5.4. Caso D: Lazo ideal en el plano vertical . . . . .	62
5.4.1. Caso D-1: Lazo ideal en el plano vertical para velocidad y ángulo de asiento de la velocidad variables . . . . .	63
5.4.2. Caso D-2: Lazo ideal en el plano vertical para empuje y ángulo de asiento de la velocidad variables . . . . .	70
5.5. Análisis general . . . . .	74
<b>6. Conclusiones</b>	<b>79</b>
6.1. Consecución de objetivos . . . . .	81
6.2. Aplicación de lo aprendido . . . . .	82
6.3. Lecciones aprendidas . . . . .	82
6.4. Trabajos futuros . . . . .	83

# Índice de figuras

1.1. Sistema dinámico formado por dos carros y un sistema de muelles y amortiguadores viscosos . . . . .	2
1.2. Diagrama de Gantt con la planificación temporal del proyecto . . . . .	7
2.1. Representación del sistema del péndulo simple . . . . .	14
2.2. Representación tridimensional del atractor de Lorenz . . . . .	17
2.3. Evolución con el tiempo de la coordenada x en el atractor de Lorenz . . . . .	18
2.4. Atractores de Rössler, Thomas y <i>Three-Scroll Unified Chaotic System</i> . . . . .	19
2.5. Relación entre el sistema horizonte local y viento . . . . .	21
2.6. Representación de las tres vistas de la aeronave junto a los ejes viento asociados . . . . .	22
3.1. Diagrama del funcionamiento de SINDy . . . . .	28
3.2. Representación del atractor de Lorenz . . . . .	32
3.3. Representación del atractor de Rössler . . . . .	33
3.4. Representación del atractor de Thomas . . . . .	34
3.5. Representación del atractor de <i>Three-Scroll Unified Chaotic System</i> . . . . .	35
3.6. Error cuadrático medio en los coeficientes de la ecuación para los atractores de Lorenz, Rössler, Thomas y <i>Three-Scroll Unified Chaotic System</i> . . . . .	38
4.1. Tiempo de ejecución para cada uno de los casos desarrollados sobre aeronaves	43
5.1. Caso A. Representación de las variables que intervienen en la trayectoria de un planeador . . . . .	47

5.2. Caso A. Representación del error cuadrático medio de la trayectoria frente al número de trayectorias . . . . .	48
5.3. Caso B. Representación de la velocidad de la trayectoria frente al tiempo .	50
5.4. Caso B. Representación del error cuadrático de la trayectoria frente al tiempo con y sin ruido . . . . .	51
5.5. Caso B. Representación del error cuadrático medio de la trayectoria y de los coeficientes . . . . .	52
5.6. Caso C-1. Representación de la velocidad de la trayectoria frente al tiempo	54
5.7. Caso C-1. Representación del error cuadrático de la trayectoria frente al tiempo . . . . .	55
5.8. Caso C-1. Representación del error cuadrático medio de la trayectoria frente a la potencia de ruido . . . . .	56
5.9. Caso C-1. Representación del error cuadrático medio de la trayectoria y de los coeficientes de la ecuación (5.20) . . . . .	57
5.10. Caso C-1. Representación del error cuadrático medio de los coeficientes de la ecuación (5.20) frente a la potencia de ruido . . . . .	58
5.11. Caso C-2. Representación de la velocidad de la trayectoria frente al tiempo para una aproximación en serie de Taylor de orden 2 y 3 . . . . .	60
5.12. Caso C-2. Representación del error cuadrático medio de la trayectoria frente al número de trayectorias . . . . .	61
5.13. Caso C-2. Representación del error cuadrático medio de la trayectoria al variar la duración temporal de la trayectoria de validación . . . . .	61
5.14. Caso C-2. Representación del error cuadrático medio de la trayectoria variando parámetros del vector de tiempo . . . . .	62
5.15. Caso D-1. Representación de la velocidad y ángulo de asiento de la velocidad ( $\gamma$ ) de la trayectoria frente al tiempo . . . . .	65
5.16. Caso D-1. Representación del error cuadrático medio de la trayectoria frente al número de trayectorias . . . . .	66
5.17. Caso D-1. Representación del error cuadrático medio de la trayectoria al variar la duración temporal de la trayectoria de validación . . . . .	67

5.18. Caso D-1. Representación del error cuadrático medio de la trayectoria y de los coeficientes del sistema de ecuaciones (5.44) . . . . .	68
5.19. Caso D-1. Representación del error cuadrático medio los coeficientes del sistema de ecuaciones (5.44) frente a la potencia de ruido . . . . .	69
5.20. Caso D-2. Representación del empuje y ángulo de asiento de la velocidad ( $\gamma$ ) de la trayectoria frente al tiempo . . . . .	72
5.21. Caso D-2. Representación del error cuadrático medio de la trayectoria frente al número de trayectorias . . . . .	73
5.22. Caso D-2. Representación del error cuadrático medio de la trayectoria frente a la potencia de ruido . . . . .	74
5.23. Error cuadrático medio en los coeficientes para los casos analizados . . . . .	75



# Índice de tablas

1.1. Objetivos generales y específicos del proyecto . . . . .	6
1.2. Estructura de la memoria . . . . .	9
3.1. Tabla comparativa entre las ecuaciones obtenidas por SINDy y las teóricas para el caso de atractores extraños . . . . .	37
5.1. Índice de los casos sobre aeronaves desarrollados en el proyecto . . . . .	46
5.2. Tabla comparativa entre las ecuaciones obtenidas por SINDy y las teóricas para las trayectorias de aeronaves . . . . .	77



# Índice de listas de código

3.1. Regresión dispersa en Matlab . . . . .	31
4.1. Pseudocódigo para la generación de las trayectorias . . . . .	40
4.2. Pseudocódigo para el entrenamiento del modelo y validación de las ecuaciones obtenidas por SINDy . . . . .	41



# Capítulo 1

## Introducción

Vivimos en un mundo en constante expansión en el que se generan grandes cantidades de información. Por ello, es especialmente importante saber procesar esos datos para resolver problemas de naturaleza muy diversa.

Gracias a nuestra comprensión de la naturaleza ha sido posible el desarrollo de la tecnología en un amplio rango de campos, que incluyen la aviación, motores de combustión, satélites o las telecomunicaciones. Campos que, gracias a sus respectivas aportaciones, hacen posible mejorar nuestra calidad de vida.

En el contexto de la ingeniería aeroespacial en general y de este proyecto en particular, los avances nos han permitido eliminar barreras en cuanto al tiempo que nos lleva transportar personas y mercancías, promoviendo un mundo globalizado y en constante desarrollo.

Algunos de estos avances son el desarrollo de modelos que describan el comportamiento de una aeronave en vuelo dadas unas condiciones determinadas. Ello nos permite conocer toda su trayectoria a lo largo del tiempo, lo que nos facilita optimizarla (previo al despegue) y minimizar la intervención humana mientras se encuentra en vuelo, ya que es posible llevar a cabo una evaluación en la toma de decisiones que provoquen cambios en el comportamiento del sistema (por ejemplo, decisiones del piloto en cuanto al guiado de la aeronave) y las consecuencias que acarrearán estas previo al vuelo de dicha aeronave.

Algunos de estos modelos son modelos dinámicos, los cuales se basan en ecuaciones diferenciales para describir el comportamiento de sistemas físicos. Un ejemplo sería el de un carro sobre el que actúa la fuerza de un muelle, figura 1.1, este caso es analizado en

profundidad en la sección 2.1.

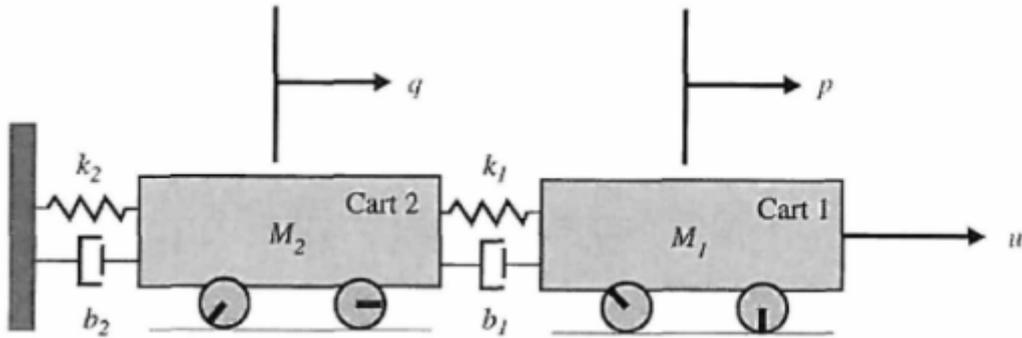


Figura 1.1: Sistema dinámico formado por dos carros y un sistema de muelles y amortiguadores viscosos. Tomado de [44, diapositiva 25].

Gracias a estos modelos dinámicos es posible entender las variables que intervienen en él, las relaciones entre ellas y los posibles fallos y limitaciones del sistema previo a su puesta en marcha. Sin dichos modelos, en muchos casos no sería posible visualizar estas dependencias. De esta forma es posible establecer un sistema de control que controle la operación del sistema.

Debido a ello, en el campo de la ingeniería aeroespacial aparece el área de mecánica del vuelo (MV). Puede definirse como la ciencia que se encarga de estudiar el movimiento de vehículos voladores, siendo un vehículo cualquier objeto artificial formado por un conjunto de sistemas deformables ligados entre sí [5, prólogo].

La MV nos permite modelar el comportamiento de una aeronave en vuelo ante determinadas condiciones, logrando un doble objetivo: conocer su posición, velocidad y actitud en cada instante de tiempo y aplicar un bucle de control para poder llevar a cabo las maniobras necesarias.

El objetivo de este proyecto es analizar la viabilidad de la técnica de regresión dispersa propuesta en [7] para la recuperación de las ecuaciones dinámicas que modelan el vuelo de aeronave. Asimismo, se pretende dar una serie de indicaciones para mejorar los resultados obtenidos utilizando esta técnica.

En lo que resta de capítulo se analizará la motivación subyacente para el desarrollo de este proyecto, la importancia de utilizar datos cuya relevancia y procesado sean los adecuados, se analizan los objetivos perseguidos, se adjunta la planificación llevada a cabo

y se detalla la estructura de la memoria.

## 1.1. Motivación

El modelado matemático de sistemas físicos es una tarea compleja pero a la vez muy importante, pues gracias a ello somos capaces de entender matemáticamente el mundo que nos rodea, permitiendo establecer un control sobre ellos para obtener el resultado que deseamos.

La generación de estos modelos manualmente (conociendo las características físicas de la aeronave escribir las ecuaciones que gobiernen su movimiento) es una tarea compleja, ya que requiere realizar determinadas asunciones y simplificaciones, pues por muy complejo que sea el modelo, en todos ellos se realizan simplificaciones de la realidad.

En contraposición, obtener los datos de posición, velocidad y/o aceleración lineal y angular de una aeronave para después adquirir el modelo dinámico implícito en ellos, apenas requiere tiempo ni intervención humana siendo este modelo mucho más fiel a la realidad. Además, mediante un algoritmo o técnica adecuada es posible realizar pruebas y simulaciones de manera más rápida, fiable y económica.

En esta situación, la expansión de ciertas disciplinas como el Big Data junto con el aprendizaje automático (*machine learning*, ML) y la ciencia de datos han hecho posible llevar a cabo grandes avances en un corto periodo de tiempo. Estas disciplinas permiten generar, a partir de un conjunto de datos (posiblemente grande), un modelo de la realidad de una forma sencilla, rápida y precisa.

Hasta ahora se ha utilizado esta técnica para la modelización de atractores extraños o para modelar el flujo que rodea un cilindro. En este trabajo vamos un paso más allá, aplicando esta técnica a sistemas complejos y reales, como puede ser la trayectoria de una aeronave. A continuación, se detallan algunos beneficios que se obtienen de este planteamiento.

Gracias a la aplicación de técnicas de ciencia de datos es posible obtener los modelos de ecuaciones dinámicas del vuelo de una aeronave para así llevar a cabo simulaciones del comportamiento de la misma, reduciendo las inversiones de tiempo y dinero necesarias para la construcción de un simulador de vuelo, realizar predicciones, etc. Esto permite a

los fabricantes basarse en los datos de vuelos anteriores para obtener el modelo dinámico correspondiente y realizar predicciones ante situaciones nuevas no vistas con anterioridad, para comprobar si la aeronave actuaría de la forma deseada.

También sería posible utilizar esta técnica para eliminar costes económicos y temporales en los túneles de viento a la hora de determinar los coeficientes aerodinámicos, junto a otros parámetros físicos, que describan la aeronave. De esta forma se realizarían algunos vuelos con la aeronave, se captarían datos de las variables relevantes, se obtendrían las ecuaciones asociadas a esos vuelos mediante la técnica propuesta y se calcularían esos parámetros físicos, que se encuentran implícitos en los coeficientes de las ecuaciones.

Asimismo, es posible obtener modelos más simples que capten la esencia de todo el comportamiento del sistema a partir de los datos recogidos de manera que los sistemas informáticos y de control sean capaces de procesar ecuaciones aún más sencillas pero con toda la información necesaria.

Por todo lo expuesto, resulta muy interesante estudiar las técnicas de ciencia de datos, como es la regresión dispersa (*sparse* en inglés) en el campo de la ingeniería aeroespacial [37]. Esta técnica se basa en que, a la hora de hacer la regresión, la mayoría de términos de una serie de candidatos son anulados. En el contexto de la predicción de ecuaciones, esto es que la mayoría de coeficientes de una librería de funciones amplia serán anulados. Esta definición se adapta perfectamente al problema de modelar el comportamiento de una aeronave a partir de datos de sus trayectorias, donde solo unos pocos términos de la ecuación no se anulan entre un conjunto grande de posibles candidatos. Asimismo, cabe destacar que utilizar esta técnica en este campo resulta totalmente novedoso, siendo una línea de investigación en la que se ha realizado poca profundización.

### 1.1.1. Big Data y ciencia de datos

La ciencia de datos es un campo que, a primera vista, puede parecer reciente, sin embargo, este área se ha desarrollado durante años.

Esta disciplina se puede definir como aquella que se encarga de analizar, procesar y almacenar conjuntos complejos de información, más conocidos como Big Data, ya bien sea por su tamaño o por su dificultad a la hora de interpretarlos. Sobre todo entra en escena cuando las técnicas de procesamiento habituales no surten efecto y, por tanto, es

necesario buscar nuevas alternativas [4, capítulo 3].

La ciencia de datos combina matemáticas, estadística, informática y la experiencia para llevar a cabo algunas de estas tareas: predicciones más precisas, optimización en el proceso de aprendizaje máquina y mejora en la toma de decisiones, entre otros.

Debido a lo expuesto anteriormente, es necesario realizar un análisis con las técnicas de ciencia de datos, ya que, para obtener buenos resultados del modelo a implementar, es preciso alimentarlo con datos cuya relevancia y estructura sea la adecuada. Para ello, se debe llevar a cabo una metodología concreta, en [4, figura 3.6] se analizan los distintos pasos.

En primer lugar se debe definir y entender correctamente la problemática en cuestión e identificar los conjuntos de datos necesarios. A continuación, es necesario filtrar, transformar y validar los datos con el fin de que no haya campos corruptos, registros faltantes o redundantes. Por último, es necesario llevar a cabo la etapa de análisis de dichos datos procesados (tanto matemática y estadísticamente como visualmente), con el objetivo de identificar los patrones internos presentes en ellos.

El Big Data proporciona diversas ventajas, algunas de ellas son [43]:

- Explicar el comportamiento actual y futuro de sistemas físicos gracias al conocimiento de datos adquiridos en el pasado.
- El Big Data basa su infraestructura en herramientas tecnológicas para llevar a cabo el ciclo de procesar, almacenar y gestionar la información. Gracias a ello es posible acceder de manera rápida y eficiente a un lugar centralizado en el que se encuentra toda la información permitiendo facilitar la colaboración entre diversos expertos.
- El Big Data también se encuentra disponible en tiempo real, por lo que es posible hacer uso de esta disciplina para adaptarse a las situaciones de manera inmediata.

Hoy en día dado el gran volumen de vuelos existente y la presencia generalizada de sistemas de captura de datos, somos capaces de disponer de una cantidad ingente de datos sobre trayectorias de aeronaves. La combinación del uso de técnicas de ciencia de datos junto con las técnicas propuestas en este proyecto, puede hacer posible la utilización de estas trayectorias para la creación de un modelo dinámico que nos permita definir el comportamiento de dichas aeronaves previo al comienzo del vuelo.

## 1.2. Objetivos

En la tabla 1.1 se proponen los siguientes objetivos generales, junto con sus objetivos específicos asociados.

Objetivos generales	Objetivos específicos
<b>OG1:</b> explorar la técnica de regresión dispersa para la estimación de las ecuaciones de sistemas dinámicos	<b>OE1.1:</b> generar trayectorias de atractores extraños y estimar sus ecuaciones mediante técnicas de regresión dispersa
<b>OG2:</b> obtención de las ecuaciones de la mecánica de vuelo de aeronaves para datos de trayectorias sintéticas utilizando regresión dispersa	<b>OE2.1:</b> generar trayectorias sintéticas de aeronaves vía software eligiendo determinados parámetros físicos de la aeronave y parámetros que definen el movimiento de la misma
	<b>OE2.2:</b> preparar los datos generados en el punto anterior con el fin de que la técnica de regresión dispersa resulte efectiva
	<b>OE2.3:</b> utilizar las técnicas propuestas para recuperar el modelo dinámico que ha generado esos datos, analizar el modelo obtenido y simular trayectorias ante condiciones iniciales distintas
<b>OG3:</b> analizar los diferentes parámetros y/u opciones disponibles para mejorar los resultados obtenidos aplicando esta técnica	<b>OE3.1:</b> mejorar los resultados obtenidos modificando determinados parámetros de entrada del algoritmo y obtener las ecuaciones para distintas librerías de funciones

Tabla 1.1: Objetivos generales y específicos del proyecto.

## 1.3. Planificación temporal

La planificación temporal realizada en este proyecto se detalla en la figura 1.2. En el eje horizontal se representan las fechas en las cuales se ha desarrollado y en el eje vertical se representan las tareas llevadas a cabo. Asimismo se ilustra el nivel de esfuerzo desempeñado en cada tarea: bajo (menos de 0.5h/día de media, en color naranja), medio (3h/día de media, en color gris) y alto (6h/día de media, en color amarillo). Aproximadamente, se han invertido unas 800 horas en la realización de este TFG. Esto es debido, por un lado, a haber comenzado pronto tanto a realizarlo y, por otro lado, a la alta dedicación en la adecuación de las ecuaciones de una aeronave para conseguir buenos resultados aplicando la técnica de regresión dispersa.

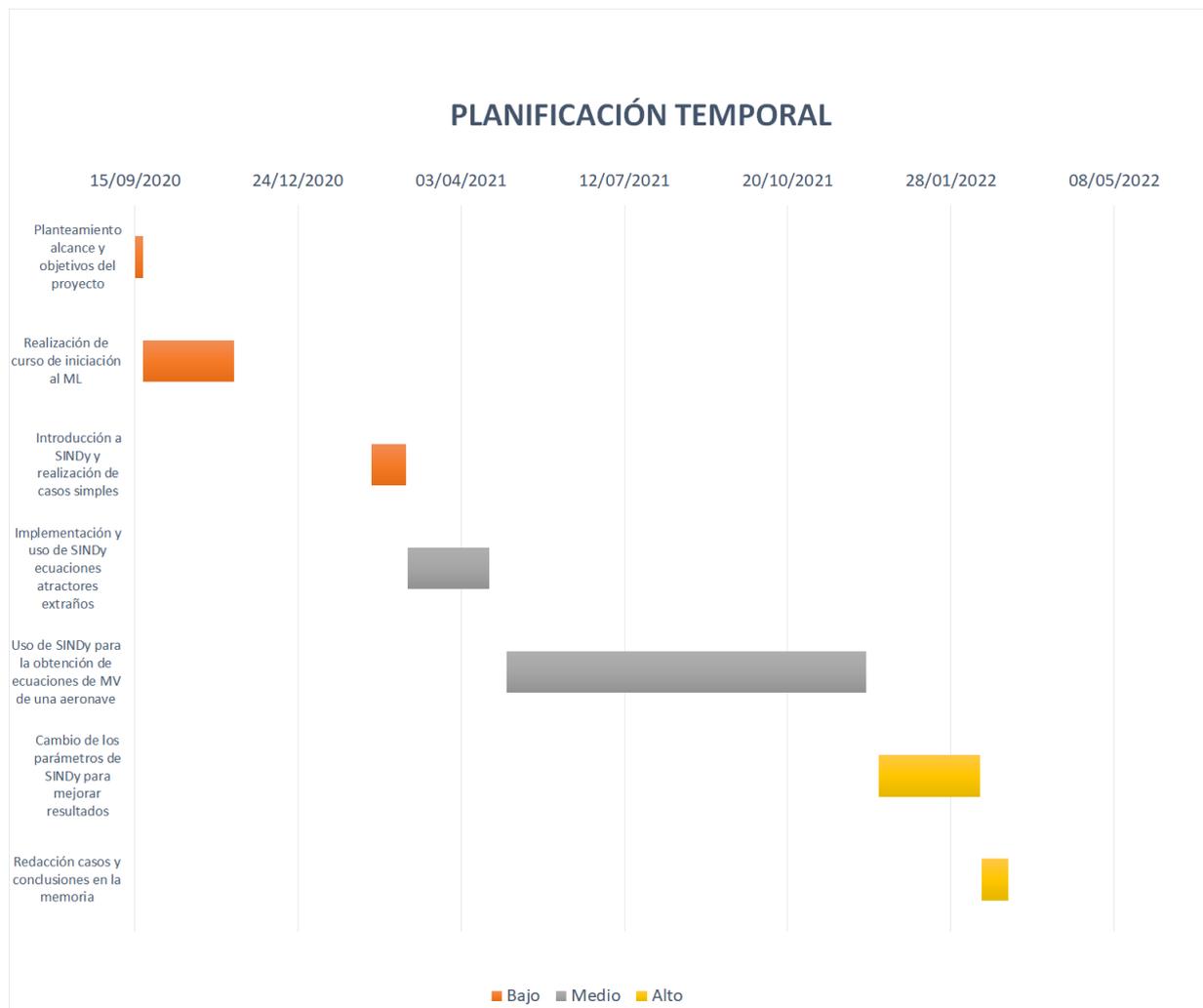


Figura 1.2: Diagrama de Gantt con la planificación temporal del proyecto.

## 1.4. Estructura de la memoria

La estructura llevada a cabo en el presente proyecto se detalla en la tabla 1.2.

Adicionalmente, se pueden consultar los siguientes recursos en línea asociados a este proyecto:

- Página web del proyecto: <https://meridiaz.github.io/id-dispersa-aeronaves/>.
- Conceptos teóricos: <https://meridiaz.github.io/id-dispersa-aeronaves/teoria.html>.
- Conceptos sobre atractores extraños y recuperación de sus ecuaciones utilizando la técnica propuesta: <https://meridiaz.github.io/id-dispersa-aeronaves/atractores.html>.
- Recuperación de las ecuaciones del vuelo de una aeronave para distintos casos: <https://meridiaz.github.io/id-dispersa-aeronaves/aeronaves.html>.
- Repositorio de código: <https://github.com/meridiaz/id-dispersa-aeronaves/tree/main>.

Título	Detalle
<b>Introducción,</b> capítulo 1	En este primer capítulo se ha realizado una contextualización del problema abordado. En concreto, se realiza una introducción del mismo, se explica la motivación por la cual es importante abordar esta problemática, se hace hincapié en la importancia de generar datos de calidad, se adjunta la planificación temporal llevada a cabo y se plantean los objetivos perseguidos
<b>Marco teórico,</b> capítulo 2	En este capítulo se detallan los conceptos matemáticos necesarios para el desarrollo de este proyecto. En concreto, se analizan los sistemas dinámicos (ilustrando algunos ejemplos como los atractores) y se introducen las ecuaciones de la mecánica de vuelo necesarias para la generación de los datos de aeronaves
<b><i>Sparse Identification of Nonlinear Dynamics (SINDy)</i>,</b> capítulo 3	En este capítulo se explica la técnica de regresión dispersa usada en este proyecto y se detalla el funcionamiento del algoritmo utilizado. Asimismo, se analiza sobre los casos de atractores extraños implementados
<b>Arquitectura general,</b> capítulo 4	En este capítulo se detalla la metodología general de los seis experimentos (adjuntando pseudocódigos) y se muestra el tiempo de ejecución de cada caso
<b>Resultados,</b> capítulo 5	En este capítulo se analiza la técnica sobre trayectorias de aeronaves. Para ello, se explican las ecuaciones utilizadas del movimiento de la aeronave en cada caso, se obtienen dichas ecuaciones utilizando el algoritmo descrito, se analizan los resultados de acuerdo a las métricas definidas y se analiza el impacto de cambiar diversos parámetros en el desempeño de los modelos obtenido
<b>Conclusiones,</b> capítulo 6	Finalmente, se termina con unas conclusiones, se analiza la consecución de objetivos planteados en este capítulo, sección 1.2, y se marcan líneas de trabajo futuras

Tabla 1.2: Estructura de la memoria.



# Capítulo 2

## Marco teórico

En este capítulo se introducen los conceptos teóricos necesarios para el posterior desarrollo de este proyecto. Primeramente, se hace una revisión del concepto de sistemas dinámicos y se analizan algunos ejemplos. Por último, se introducen las ecuaciones generales de MV a utilizar en el desarrollo de los distintos experimentos.

Este capítulo pretende ahondar en la importancia de los sistemas dinámicos en la modelización de la realidad. Este campo de estudio es especialmente relevante en MV para modelar el comportamiento de una aeronave con el tiempo según su configuración de vuelo.

### 2.1. Sistemas dinámicos

De manera general un sistema consiste en un conjunto de componentes interconectados, construidos con un propósito determinado. En la naturaleza es posible encontrar sistemas estáticos, cuya salida depende únicamente del valor de la entrada, y modelos dinámicos, cuya salida depende del valor pasado y presente de la variable de entrada [28, 39].

Un sistema dinámico es la representación matemática abstracta que modela un fenómeno creado por el hombre, biológico o físico que cambia con el tiempo [38, 40]. En concreto, define los elementos que intervienen en él y las relaciones entre ellos. Estos sistemas se pueden representar mediante un conjunto de ecuaciones diferenciales ordinarias y ecuaciones en derivadas parciales (ambos en el caso de sistemas continuos) o por transformaciones

discretas, típicamente ecuaciones en diferencias [26].

La siguiente expresión ilustra, de manera simplificada, una ecuación diferencial

$$\frac{dx}{dt} = f(x(t)). \quad (2.1)$$

La variable  $x$  es una variable de estado que puede ser un vector (por ejemplo la velocidad) o un escalar (por ejemplo una variable angular). Esta variable evoluciona con el tiempo,  $t$ .  $f$  es una función que define las reglas y relaciones para el cual el sistema cambia su estado a lo largo del tiempo.

Estas ecuaciones diferenciales permiten llevar a cabo un análisis mucho más sencillo del sistema. Gracias a ello, pueden extraerse relaciones y propiedades que primeramente podrían permanecer ocultas. Asimismo, pueden representarse fácilmente mediante su función de transferencia o mediante el espacio de los estados [39].

El espacio de estados se puede definir como el conjunto de posibles estados del sistema a lo largo del tiempo, de esta forma cada punto del espacio de estados corresponde a un estado diferente del sistema. El mínimo número de variables linealmente independientes que describen completamente el estado del sistema dinámico se denominan **variables de estado**, mientras que el conjunto de todas ellas es el **vector de estado** [35] y [44, diapositiva 9]. Por ejemplo, en el caso del movimiento de un péndulo ideal, representado en la figura 2.1, el espacio de estados se corresponde con todos los posibles valores de la tupla ángulo respecto de la vertical y velocidad angular [50].

Gracias a esta representación mediante ecuaciones diferenciales es posible conocer el estado del sistema en cualquier instante dado únicamente su estado inicial en el instante inicial. Cabe destacar que a pesar de esto, el estado obtenido puede contener un error respecto de la realidad. Por muy preciso que sea un modelo no se pueden tener en cuenta todos los factores que afecten al sistema a modelar.

Las aplicaciones de estos modelos dinámicos que modelan la realidad son [38]:

- **Predicción:** el objetivo es predecir el futuro estado del sistema basándose en las observaciones presentes y pasadas.
- **Diagnóstico:** cuyo objetivo es inferir los estados pasados del sistema que hayan provocado el estado presente.

- **Explicación del sistema:** el objetivo es entender el comportamiento del sistema y proveer una teoría que lo explique. Este paso puede implicar obtener la representación matemática del sistema que lo represente.

Asimismo, se pueden utilizar diversos criterios para clasificar los sistemas dinámicos: pueden ser continuos o discretos, lineales o no lineales, autónomos o no autónomos y deterministas o estocásticos. Un sistema determinista es aquel en el que la condición inicial únicamente ocasiona un único estado del sistema. Por otro lado, un sistema estocástico incluye un componente aleatorio en el que la condición inicial provoca varios estados diferentes del sistema. Este caso debe ser modelado mediante una variable aleatoria junto a su densidad de probabilidad. En el presente proyecto todos los sistemas dinámicos usados son deterministas.

Algunos ejemplos de modelos dinámicos son los siguientes:

- **Crecimiento de una población de bacterias:** esta población aumenta de tamaño porque cada bacteria crece y se divide en otras dos bacterias. Para modelarlo denotaremos el número de bacterias en un instante temporal como  $b_t$ . Esta variable de estado debe ser discreta no negativa. Por tanto, el sistema dinámico obtenido sería [35]

$$b_{n+1} = 2b_n.$$

Por tanto, es posible conocer el número de bacterias presentes en cualquier instante de tiempo dado su número inicial,  $b_0$ . Como se puede ver, este caso se trata de un sistema dinámico discreto, por lo que el tiempo viene representado por la variable  $n$ .

- **Sistema formado por carros:** en la figura 1.1 se muestran unos carros unidos entre sí por unos muelles y amortiguadores viscosos. El sistema matemático que modela su comportamiento puede ser obtenido a partir de la segunda ley de Newton

$$F = m \frac{d^2 x(t)}{dt^2}. \quad (2.2)$$

De esta forma identificando las distintas fuerzas que aparecen en el sistema es posible determinar la posición de los carros en cualquier instante de tiempo:

- Fuerza externa  $u$
- La fuerza que ejerce el muelle  $F = kx$  donde  $k$  representa la constante elástica del muelle y  $x$  es la elongación que sufre el muelle  $x_f - x_0$ .
- La fuerza que ejerce el amortiguador viscoso es  $F = b\dot{x}$  donde  $b$  representa el coeficiente de amortiguamiento viscoso y  $\dot{x} = \frac{dx(t)}{dt}$  representa la velocidad del objeto.

Sustituyendo las expresiones anteriores en la segunda ley de Newton aplicada a cada carro es posible obtener el conjunto de ecuaciones que describen el comportamiento del sistema,

$$M_1\ddot{p} + b_1\dot{p} + k_1p = u + k_1q + b_1\dot{q}$$

$$M_2\ddot{q} + (b_1 + b_2)\dot{q} + (k_1 + k_2)q = k_1p + b_1\dot{p}.$$

- **Péndulo simple:** este es un sistema para el cual una masa puntual se encuentra suspendida de una varilla de masa despreciable.

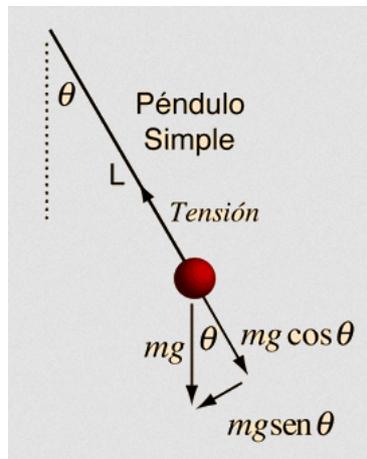


Figura 2.1: Representación del sistema del péndulo simple. Tomado de [33].

En la figura 2.1 se muestra el diagrama de fuerzas. En concreto, las fuerzas que aparecen son:

- Ángulo de giro respecto de la vertical,  $\theta$ .
- Longitud de la cuerda,  $L$ .

- Masa del objeto suspendido,  $m$ .
- Tensión de la varilla al sujetar la masa,  $T$ .
- Peso de la masa suspendida,  $mg$ .

Aplicando la segunda ley de Newton al eje tangencial, ecuación (2.2), y asumiendo ángulos pequeños,  $\sin \theta \approx \theta$ , el sistema matemático quedaría como sigue

$$\frac{d^2\theta}{dt^2} + \frac{g}{L}\theta = 0,$$

donde  $g$  representa la aceleración de la gravedad,  $\theta$  el ángulo de giro respecto de la vertical y  $R$  el radio de giro, que es igual a la longitud de la cuerda,  $L$ .

Como se puede ver se trata de una ecuación diferencial en la que es posible conocer la posición de la masa suspendida para cualquier instante dado.

### 2.1.1. Atractores

Un atractor es un conjunto de estados hacia los cuales tiende un sistema dinámico, es decir, en ciertos sistemas dinámicos, una vez eliminado el transitorio inicial, el sistema tiende a unos valores típicos o atractores [24, 46]. Otra forma de verlo es que el sistema converge a un estado del que no puede salir. El concepto de atractor está muy asociado a sistemas disipativos, que pierden energía, ya que estos también convergen a un estado final único.

Existen dos tipos de atractores, clásicos y extraños. En los **atractores clásicos** todos los estados del sistema tienden a un único punto o estado estacionario. Se caracterizan porque cuando desaparece la fuerza externa aplicada y/o existen fuerzas de fricción, las condiciones iniciales para los cuales han sido generados se mimetizan con el estado estacionario del sistema. Los subtipos que engloba son los siguientes [48]:

- **Punto fijo:** todos los estados del sistema tienden hacia un punto fijo. Ese punto será un atractor del sistema. Un ejemplo cotidiano sería una pelota de fútbol que es golpeada. Una vez que ha transcurrido el tiempo suficiente la velocidad de la pelota es nula, da igual la fuerza inicial que se aplique o la posición desde la que se parta.

- **Ciclo límite:** es una trayectoria cerrada en el espacio de estados. En este caso los estados del sistema tienden a un estado periódico. Un claro ejemplo de este tipo es el ritmo cardiaco. Por mucho que se eleve realizando actividad física, una vez que la persona descansa el ritmo cardiaco volverá a su estado normal, tanto en amplitud como en frecuencia.
- **Atractor cuasi periódico o toro:** en este caso el atractor en lugar de converger a un punto fijo o a al mismo estado, se mantiene en una superficie bidimensional de forma toroidal. En este atractor, las trayectorias se encontrarán dentro de él sin llegar a ser periódicas. Este caso es una generalización del ciclo límite a varias dimensiones [20, 41].

Por otro lado, para hablar de los atractores extraños es necesario introducir la Teoría del Caos [3, 21]. En los ejemplos anteriores una misma condición inicial produce que el sistema evolucione de la misma manera. Los sistemas caóticos son sistemas dinámicos deterministas no lineales <sup>1</sup> en los cuales una pequeña variación en las condiciones iniciales provoca que el sistema evolucione de diferente manera, aunque el cambio en las condiciones iniciales sea mínimo. Debido a ello, la Teoría del Caos también es conocida como efecto mariposa [25].

Un sistema dinámico que presente un **atractor extraño** es denominado sistema caótico y exhibirá un comportamiento no predecible. Esta definición puede parecer contradictoria con el concepto de atractor introducido en la sección anterior. En un sistema caótico que presente atractores extraños se sabe que su comportamiento general será siempre el mismo y tendrá la misma forma en todos los casos (representada por las mismas ecuaciones), pero no se puede conocer el estado exacto del sistema en un instante determinado, es impredecible [49]. A pesar de que estos sistemas son deterministas, un pequeño error de medición en las condiciones iniciales provoca resultados muy diferentes, complicando su predicción a largo plazo.

El nombre atractor extraño es acuñado porque el sistema tiende a un estado cuya forma no viene representada por una forma conocida, como un punto o un círculo, sino, que viene representado por extrañas formas.

---

<sup>1</sup>Es conveniente señalar que todo sistema caótico es no lineal, pero no al revés: un sistema no lineal no tiene por qué ser caótico y puede tener atractores clásicos

El atractor extraño más famoso es el **atractor de Lorenz**, representado en la figura 2.2 para los coeficientes  $a = 10$ ,  $b = 28$  y  $c = 8/3$ . Este atractor aparece en el sistema dinámico caótico que postuló Edward N. Lorenz en 1963 para modelar, de forma simplificada, el comportamiento de grandes masas de aire en la atmósfera. Las ecuaciones que lo modelan son las siguientes [45]

$$\begin{aligned}\dot{x} &= a(y - x) \\ \dot{y} &= x(b - z) - y \\ \dot{z} &= xy - cz.\end{aligned}\tag{2.3}$$

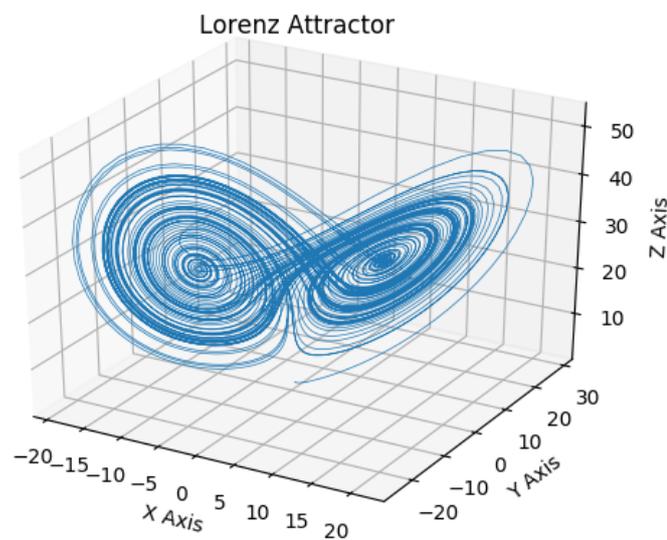


Figura 2.2: Representación tridimensional del atractor de Lorenz. Ejes en unidades adimensionales. Tomado de [29].

Debido a este comportamiento caótico, aunque dos trayectorias comiencen cercanas en el espacio, pueden acabar en desarrollándose en lóbulos distintos del atractor.

Veamos un ejemplo para poner de manifiesto esta afirmación. En la figura 2.3 se ilustra el atractor de Lorenz para dos puntos iniciales distintos  $P_1 = (0, 2, 0)$  (en color negro) y  $P_2 = (0; 2, 01; 0)$  (en color gris). Como se puede ver ambos puntos se encuentran muy cercanos en el espacio y en una parte de su camino coinciden en su coordenada  $x$ . Sin

embargo, a medida que pasa el tiempo cada trayectoria tiende a un lóbulo distinto del atractor, a pesar de haber comenzado en un punto del espacio casi idéntico.

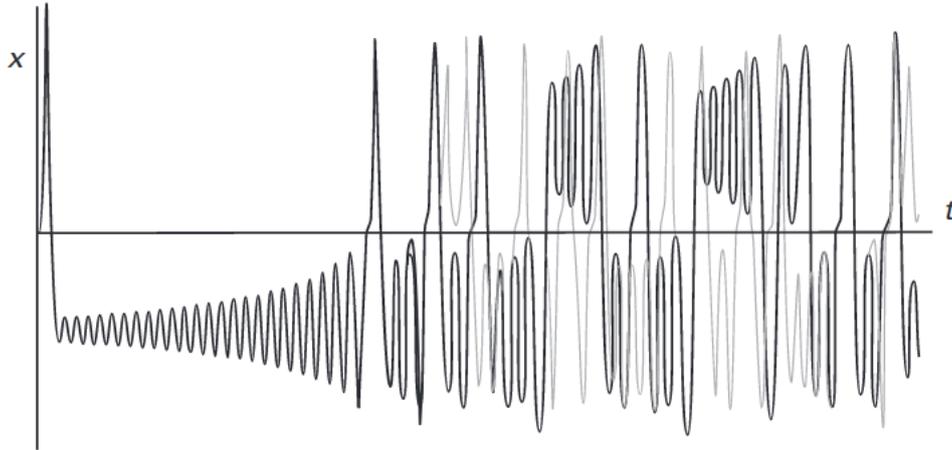


Figura 2.3: Evolución con el tiempo de la coordenada  $x$  en el atractor de Lorenz. Tomado de [3, Figura 14.2].

Asimismo existen otros atractores extraños usados en este proyecto, todos ellos representados en la figura 2.4 junto con las ecuaciones que los caracterizan:

- El **atractor de Rössler** que aparece en los sistemas de reacciones químicas oscilantes [42].
- El **atractor de Thomas** presenta una forma simétrica en las variables  $x$ ,  $y$ , y  $z$  [47].
- El **atractor del sistema caótico unificado de tres lóbulos** o en inglés *Three-Scroll Unified Chaotic System* el cual se obtiene de una particularización en los parámetros del atractor de Lü [11].

## 2.2. Ecuaciones de la mecánica de vuelo

En el capítulo 1 se ha hecho una pequeña introducción al concepto de MV, el cual abarca el estudio de submarinos, aviones, globos, helicópteros o satélites artificiales, entre otros. Sin embargo, en este proyecto únicamente se ha estudiado el comportamiento de aviones.

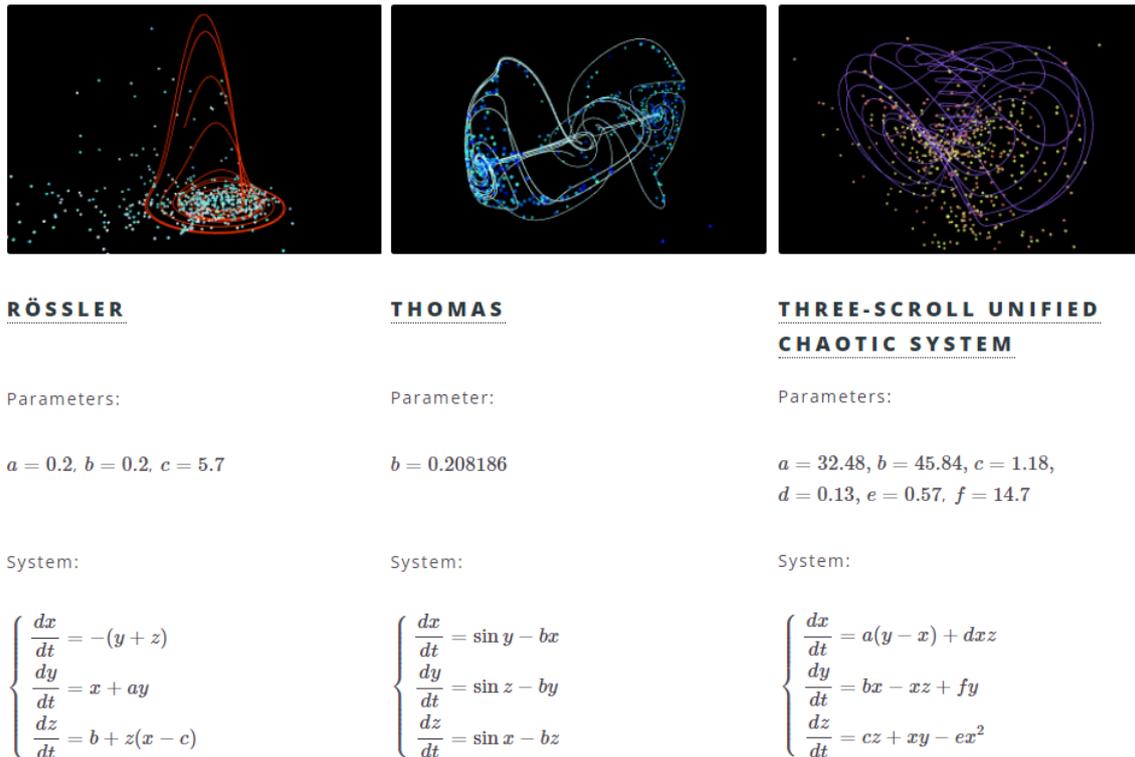


Figura 2.4: De izquierda derecha atractores de Rössler, Thomas y *Three-Scroll Unified Chaotic System*. Adaptado de [22].

La MV debe abarcar las siguientes tareas: desarrollar sistemas de control de vuelo, modelización, simulación, desarrollo de leyes de control, estimación, guiado, navegación, gestión de misión y monitorización de datos [30]. En concreto, el estudio del avión se divide en varias secciones:

- **Actuaciones:** estudia las actuaciones, cuyo propósito final consiste en establecer la configuración del avión más adecuada para llevar a cabo una misión. En concreto, sigue el centro de masas del avión a lo largo de su trayectoria.
- **Estabilidad y control:** un vez establecida la configuración de la aeronave es necesario llevar a cabo una predicción del comportamiento de esta y comprobar que determinados parámetros de la aeronave se mantienen en un rango de valores adecuado. Además, es preciso que una vez concluida una maniobra o perturbación, la aeronave pueda volver a la posición de equilibrio, esto es lo que se conoce como estabilidad.

- **Aeroelasticidad:** esta sección trata sobre las deformaciones elásticas que son importantes en cuanto a que afectan a las fuerzas que actúan sobre el avión, como las dinámicas.

En concreto en este proyecto se centra en el estudio de la aeronave como un punto, no se considera en ningún momento como un sólido rígido.

### 2.2.1. Sistema de referencia y ecuaciones del movimiento

Un sistema de referencia es un conjunto de coordenadas medidas en el espacio y tiempo necesarias para determinar la posición de un punto en el espacio [32]. Es decir, es el conjunto de convenciones que un observador necesita para medir magnitudes físicas en un sistema mecánico. Los sistemas de referencia son necesarios para entender el contexto y dimensiones sobre las que se desarrolla el movimiento de un cuerpo [31].

En la MV existen diversos sistemas de referencia: sistema de referencia inercial, sistema de referencia geocéntrico giratorio, sistema de ejes tierra, sistema de ejes de horizonte local, sistema de ejes cuerpo y sistema de ejes viento. Excepto los dos últimos, todos ellos son sistemas de referencia inerciales. Esto es que no es necesario considerar fuerzas ficticias ni la aceleración de Coriolis en el lado derecho de la ecuación.

Los sistemas ejes cuerpo y viento son sistemas no inerciales, es decir, su posición no es fija respecto un punto y no se puede utilizar la expresión (2.2) directamente.

En concreto, el sistema utilizado en este proyecto será el sistema de **ejes viento** denotado como  $F_w(O_w, x_w, y_w, z_w)$ , donde  $O_w$  es el centro u origen del sistema (cualquier punto del avión dentro del plano de simetría) y  $x_w, y_w$  y  $z_w$  son los tres ejes perpendiculares entre sí que forman un triedro a derechas. El eje  $x_w$  se orienta según la dirección y sentido de la velocidad aerodinámica,  $z_w$  se orienta en el plano de simetría hacia abajo, perpendicular a  $x_w$ . Por último,  $y_w$  se dirige formando un triedro a derechas con los otros dos ejes [5, capítulo 1].

Se han seleccionado el sistema de ejes viento porque junto con el sistema ejes cuerpo es el más utilizado y permite proyectar fácilmente los vectores de las fuerzas aerodinámicas.

En la figura 2.5 se detalla la relación entre el sistema horizonte local (cuyo origen se encuentra en el plano de simetría del avión, con el eje  $z_h$  dirigido al centro de la tierra,

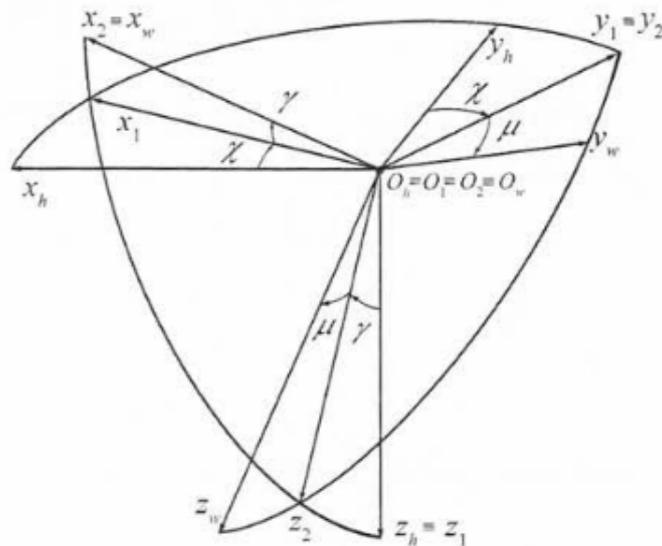


Figura 2.5: Relación entre el sistema horizonte local y viento. En el texto principal se detalla el significado de cada ángulo. Tomado de [5, figura 1.6].

el eje  $x_h$  dirigido hacia el Norte y el eje  $y_h$  formando un triedro a derechas) y viento.  $\chi$  representa el ángulo de guiñada del avión,  $\gamma$  representa el ángulo de asiento de la velocidad y  $\mu$  representa el ángulo de balance de la velocidad.

Para el desarrollo de la ecuación general del vuelo de una aeronave se debe considerar el teorema de cantidad de movimiento. Debido a la naturaleza de los casos desarrollados no es necesario utilizar el teorema del momento cinético para obtener las ecuaciones que modelan las rotaciones del avión. Asimismo, se particulariza para el caso en el que no hay presencia de viento y se aplica para las fuerzas aerodinámicas (sustentación,  $L$  y resistencia aerodinámicas,  $D$ ), no aerodinámicas (empuje,  $T$  y peso,  $mg$ ) y proyectando la velocidad angular en el sistema ejes viento se obtiene el sistema

$$\begin{aligned}
 T - D - mg \sin \gamma - m\dot{V} &= 0 \\
 mg \cos \gamma \sin \mu + mV(\dot{\gamma} \sin \mu - \dot{\chi} \cos \gamma \cos \mu) &= 0 \\
 -L + mg \cos \gamma \cos \mu + mV(\dot{\gamma} \cos \mu + \dot{\chi} \cos \gamma \sin \mu) &= 0,
 \end{aligned} \tag{2.4}$$

donde no se ha tenido en cuenta la fuerza lateral  $Q$  y se ha considerado que el empuje se encuentra orientado en el sentido que indica el eje  $x_w$ . Teniendo en cuenta que en este

proyecto los movimientos simulados son simétricos, se pueden eliminar de la ecuación anterior los ángulos  $\chi$  y  $\mu$ . La ecuación simplificada o **relación dinámica de fuerzas** es la siguiente

$$\begin{aligned} T - D - mg \sin \gamma - m\dot{V} &= 0 \\ 0 &= 0 \\ -L + mg \cos \gamma + mV\dot{\gamma} &= 0. \end{aligned} \tag{2.5}$$

Es conveniente remarcar, dado que estamos tratando de modelar sistemas dinámicos (i.e. sistemas que evolucionan con el tiempo) y como veremos el algoritmo a utilizar se aplica a este tipo de sistemas, las ecuaciones anteriores contienen derivadas temporales de algunas de las variables (velocidad, ángulos, etc.). Las derivadas a priori no son nulas, por lo que las variables evolucionarán con el tiempo. Esta evolución es la que queremos capturar en las ecuaciones a predecir con el algoritmo. Esto contrasta con sistemas estáticos donde sus variables no evolucionan en el tiempo.

En la figura 2.6 se ilustra el diagrama de fuerzas representado por el sistema (2.5).

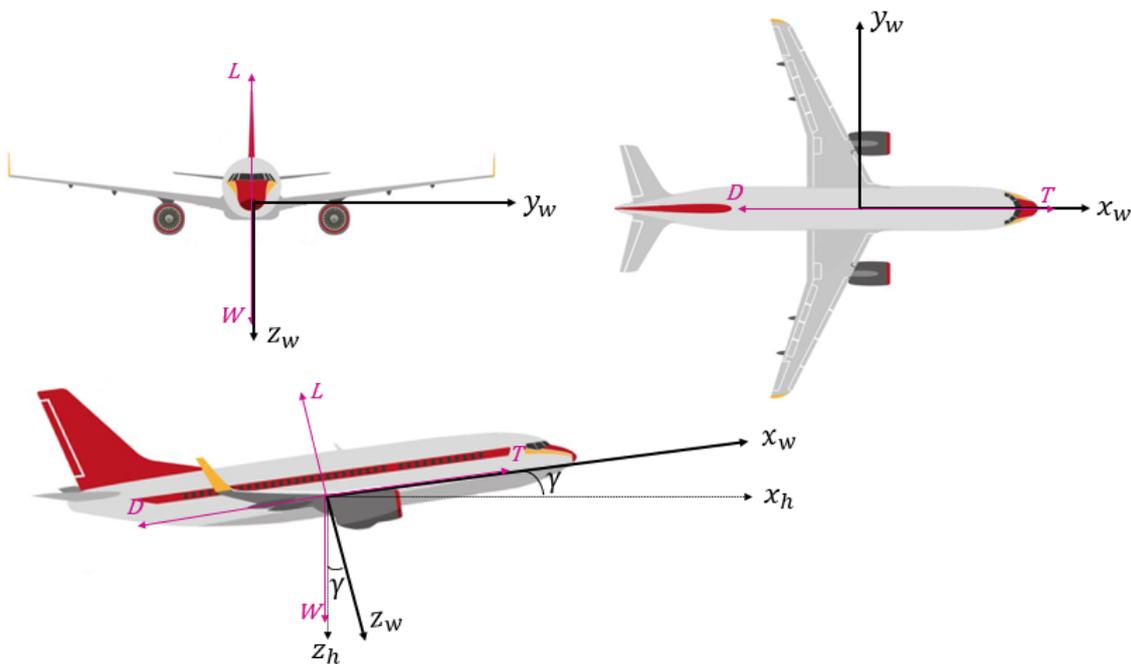


Figura 2.6: Representación de las tres vistas de la aeronave junto a los ejes viento asociados y las fuerzas consideradas que actúan en la ecuación (2.5). Elaboración propia.

Por último, pueden obtenerse **las relaciones cinemáticas** en ejes tierra

$$\begin{aligned} \dot{x}_e &= V \cos \gamma \\ \dot{y}_e &= 0 \\ \dot{z}_e &= -V \sin \gamma. \end{aligned} \tag{2.6}$$

La última ecuación puede ser reescrita en función de la altitud a la que se encuentra la aeronave teniendo en cuenta que  $\dot{h} = -\dot{z}_e$ .



# Capítulo 3

## Sparse Identification of Nonlinear Dynamics (SINDy)

En este capítulo se presentarán y se explicarán los conceptos teóricos asociados a la técnica de regresión dispersa utilizada en este proyecto, *Sparse Identification of Nonlinear Dynamics* (SINDy), cuya implementación se ha realizado en el lenguaje de programación Python a través de la librería PySINDy [18]. Por último, se explicarán los resultados obtenidos sobre las ecuaciones de atractores extraños, analizados en la subsección 2.1.1.

SINDy [7] es una técnica desarrollada principalmente en la Universidad de Washington, Seattle. El objetivo es obtener las ecuaciones que modelan la dinámica de un sistema a partir de mediciones de sus variables de interés, para sistemas de ecuaciones de la forma representada en la ecuación (2.1). Esta técnica aplica a diversas áreas como la ingeniería, finanzas, ecología o epidemiología. La necesidad de desarrollar esta técnica surge, por un lado de la gran cantidad de datos y técnicas para procesarlos de los que se dispone hoy en día, subsección 1.1.1, y por otro lado de la escasez de soluciones existentes para obtener la dinámica de un sistema a partir de estos datos medidos.

Trabajos anteriores sobre el descubrimiento de las ecuaciones de sistemas dinámicos no lineales a partir de datos se basan en técnicas como la regresión simbólica [6, 16], la modelización sin ecuaciones [10], la inferencia automatizada [17], las redes neuronales [13] o los algoritmos autorregresivos no lineales [2].

Los sistemas dinámicos no lineales son mucho más complejos que un sistema lineal, tanto que pueden llegar a parecer impredecibles, incluso caóticos comparado con los siste-

mas dinámicos lineales, haciéndolos muy difíciles de resolver. Esto principalmente se debe a que no se puede resolver un sistema de ecuaciones diferenciales no lineales utilizando métodos analíticos [14], es necesario usar métodos numéricos. Por estas razones los sistemas dinámicos no lineales se suelen aproximar como un conjunto de ecuaciones lineales en un determinado rango de valores. En ese sentido SINDy proporciona una excelente solución a este problema, ya que es capaz de resolver el problema utilizando como entrada unas matrices de datos obtenidas a partir de funciones no lineales, pero resolviéndolo mediante un problema lineal de regresión dispersa. Así, es capaz de modelar el comportamiento a priori impredecible de un sistema dinámico no lineal utilizando un enfoque basado en datos.

El algoritmo que implementa la técnica de SINDy recibe dos entradas. La primera es una matriz con mediciones de las variables de estado  $\mathbf{X} \in \mathbb{R}^{m \times n}$  donde  $m$  es el número de instantes temporales considerados y  $n$  el número de variables de estado, i.e. la dimensión del vector de estado  $\mathbf{x} \in \mathbb{R}^n$ . La segunda entrada es un conjunto de las  $p$  funciones candidatas básicas sobre las que realizar la regresión. Por último, SINDy también recibe como entrada las derivadas del vector de estado para todos los instantes,  $\dot{\mathbf{X}} \in \mathbb{R}^{m \times n}$ , las cuales son calculadas por la librería PySINDy en caso de que el usuario no las proporcione. A su salida, SINDy obtiene el mínimo número de funciones de entre las candidatas necesarias para describir las trayectorias que recibe a su entrada, junto con los coeficientes y variables de estado asociadas a cada función.

Más formalmente, las entradas de SINDy se organizan en dos matrices. La primera de ellas es la matriz de trayectorias  $\mathbf{X}$

$$\mathbf{X} = \begin{bmatrix} x^T(t_1) \\ x^T(t_2) \\ \vdots \\ x^T(t_m) \end{bmatrix} = \begin{bmatrix} x_1(t_1) & x_2(t_1) & \dots & x_n(t_1) \\ x_1(t_2) & x_2(t_2) & \dots & x_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_m) & x_2(t_m) & \dots & x_n(t_m) \end{bmatrix}. \quad (3.1)$$

Por otro lado, SINDy también recibe una matriz de funciones no lineales candidatas  $\Theta(\mathbf{X}) \in \mathbb{R}^{m \times p}$ , construida a partir de la función  $\Theta : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \times p}$ , que aproximan los datos de entrada y donde  $p$  es el número de funciones posibles a considerar. Por ejemplo, la siguiente  $\Theta(\mathbf{X})$  contiene términos constantes y polinómicos de hasta grado 3 (cada uno

de ellos perteneciente a una librería distinta)

$$\Theta(\mathbf{X}) = \begin{bmatrix} | & | & | & | \\ 1 & \mathbf{X} & \mathbf{X}^{P_2} & \mathbf{X}^{P_3} \\ | & | & | & | \end{bmatrix}, \quad (3.2)$$

donde  $\mathbf{X}^{P_2} \in \mathbb{R}^{m \times p_2}$  denota la matriz

$$\mathbf{X}^{P_2} = \begin{bmatrix} x_1^2(t_1) & x_1(t_1)x_2(t_1) & \dots & x_2^2(t_1) & \dots & x_n^2(t_1) \\ x_1^2(t_2) & x_1(t_2)x_2(t_2) & \dots & x_2^2(t_2) & \dots & x_n^2(t_2) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_1^2(t_m) & x_1(t_m)x_2(t_m) & \dots & x_2^2(t_m) & \dots & x_n^2(t_m) \end{bmatrix}.$$

Como se puede ver  $\Theta(\mathbf{X})$  contiene en cada columna el valor de cada una de las funciones candidatas,  $\mathbf{f}$ , para todos los instantes temporales, y en cada fila todas las combinaciones posibles de funciones candidatas entre todas las variables de estado para un determinado instante temporal.  $L$  es el número de librerías de funciones a considerar (términos constantes y polinómicas de grado 1, 2 y 3 en el ejemplo anterior, por lo que  $L = 4$  en este caso) y  $p = \sum_{i=0}^L p_i$  representa el número total de funciones a considerar. Existe un gran cantidad de soluciones posibles, por lo que SINDy utiliza regresión dispersa para determinar los coeficientes de la matriz  $\Xi \in \mathbb{R}^{p \times n}$  que activan aquellas funciones de la matriz  $\Theta(\mathbf{X})$ . Se define

$$\begin{aligned} \Xi &= [\xi_1 \quad \xi_2 \quad \dots \quad \xi_n] \\ \dot{\mathbf{X}} &= \Theta(\mathbf{X})\Xi, \end{aligned} \quad (3.3)$$

donde  $\xi_k \in \mathbb{R}^p$  son los coeficientes asociados a la variable de estado  $k$ .

SINDy utiliza **regresión dispersa**, la cual consiste en asumir que solo existen unos pocos términos relevantes por los cuales aproximar la ecuación. Un ejemplo de este tipo de algoritmo sería el representado en la siguiente ecuación

$$\min_{\Xi} \|\dot{\mathbf{X}} - \Theta(\mathbf{X})\Xi\|_F^2 \text{ sujeto a } \|\Xi\|_0 \leq \epsilon, \quad (3.4)$$

donde  $\|\cdot\|_F$  representa la norma de Frobenius,  $\|\Xi\|_0$  es la norma 0 de la matriz  $\Xi$  y denota la suma de valores distintos de cero. Por otro lado,  $\epsilon$  representa un escalar que

limita el número de valores distintos de cero de la matriz  $\Xi$ . Las funciones de la matriz  $\Theta(\mathbf{X}) \in \mathbb{R}^{m \times p}$  asociadas a la variable de estado  $k$  son activadas por los coeficientes del vector  $\xi_k \in \mathbb{R}^p$ . La principal suposición que se realiza es que  $m \gg p$ . De manera que la derivada de cada elemento se denota por  $\dot{\mathbf{X}} \approx \Theta(\mathbf{X})\Xi$  [27].

Por tanto, disperso consiste en que el vector  $\xi_k$  contendrá en su mayoría términos nulos. Más adelante, a la hora de explicar los optimizadores que puede utilizar PySINDy, se mostrarán ejemplos de algoritmos de regresión dispersa.

En la figura 3.1 se muestra de forma esquemática el funcionamiento de SINDy, el cual comprende los pasos II-IV. En primer lugar, es necesario obtener los datos con los que alimentar al algoritmo, es decir, se debe obtener la matriz  $\mathbf{X}$  y  $\dot{\mathbf{X}}$ . En segundo lugar, se utilizan estas dos matrices para crear la librería de funciones,  $\Theta$ , y se crea la matriz  $\Theta(\mathbf{X})$  a partir de los datos que utilizará SINDy para obtener el sistema de ecuaciones que modela estos datos. En tercer lugar, SINDy estima los coeficientes de la matriz  $\Xi$  que activan los términos de la matriz  $\Theta(\mathbf{X})$ . Por último, conocidos los coeficientes es posible obtener el sistema de ecuaciones predicho.

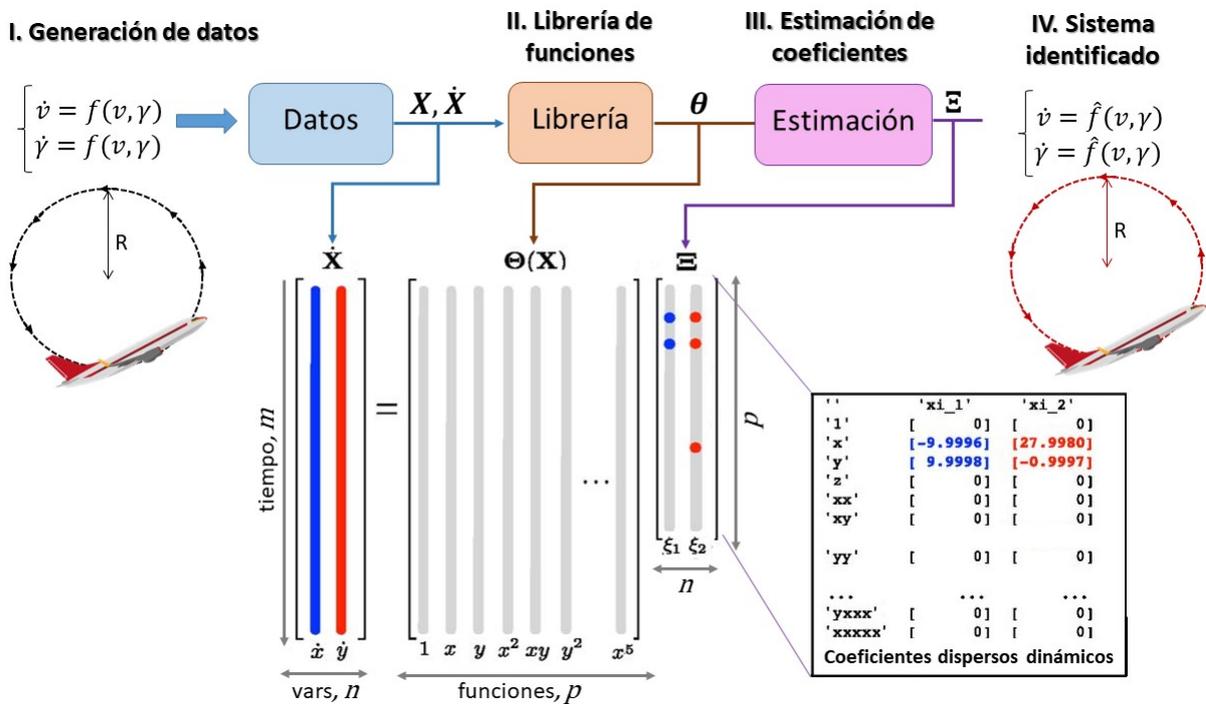


Figura 3.1: Diagrama del funcionamiento de SINDy. Adaptado de [7, figura 1]

La implementación del algoritmo SINDy en la librería PySINDy permite configurar

diversas entradas a su algoritmo. Las más relevantes son:

- **Matriz de datos,  $\mathbf{X}$** : tal y como se indica anteriormente debe contener en sus filas los distintos instantes temporales y en sus columnas las variables de estado.
- **Matriz de derivadas,  $\dot{\mathbf{X}}$**  (opcional): similar al caso anterior pero ahora debe contener el valor de las derivadas del vector de estado en cada instante temporal.
- **Paso de tiempo** (opcional, por defecto 1): PySINDy permite indicar el lapso de tiempo que transcurre entre las muestras del vector de tiempos  $\mathbf{t}$ .
- **Librería de funciones candidatas**: PySINDy debe recibir las funciones candidatas para las cuales aproximar  $\dot{\mathbf{X}}$ . Existen diversas librerías ya creadas como la polinómica o de Fourier (incluye términos trigonométricos).
- **Optimizador** (opcional, por defecto *Sequentially thresholded least squares algorithm*, STLSQ): PySINDy permite indicar qué optimizador usará en el proceso de regresión dispersa para la obtención de los pesos o coeficientes. Algunos de ellos son:
  - **STLSQ**: este optimizador minimiza, de forma iterativa, la siguiente función objetivo

$$\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \alpha \|\mathbf{w}\|_2^2.$$

En este caso al término de regresión dispersa lineal se le añade un factor de regularización  $\alpha \|\mathbf{w}\|_2^2$  con el fin de conseguir que el valor de los pesos sea pequeño [7] y [36, optimizers package].

El funcionamiento general de este optimizador consiste en a cada iteración se calculan los valores de  $\mathbf{w}$  y se da un valor de 0 a aquellos coeficientes inferiores a un umbral (por defecto 0.1), con la intención de conseguir una solución dispersa. Un ejemplo de implementación de este algoritmo en Matlab (sin el regularizador de norma 2) se puede ver en el código 3.1.

- **Regresión regularizada relajada dispersa** (*Sparse relaxed regularized regression*, SR3): este optimizador trata de minimizar la función objetivo

$$\min_{\mathbf{u}, \mathbf{w}} 0,5 \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda R(\mathbf{u}) + \frac{0,5}{\nu} \|\mathbf{w} - \mathbf{u}\|_2^2.$$

Esta propuesta es más robusta frente a errores y falsos positivos [8, 19]. En este caso el término de regularización se encuentra relajado. Por último, el término  $\frac{0.5}{\nu} \|\mathbf{w} - \mathbf{u}\|_2^2$  es utilizado para conseguir que los valores de  $\mathbf{u}$  y  $\mathbf{w}$  se parezcan lo máximo posible.

- **SR3 restringido** (*Sparse relaxed regularized regression with linear equality constraints*): este optimizador es similar al anterior, pero permite al usuario indicar una serie de restricciones lineales del tipo menor o igual a aplicar sobre los pesos [8]. La restricción es de la forma

$$\mathbf{C}\mathbf{w} \leq \mathbf{d},$$

donde  $\mathbf{C}$  representa la matriz con los coeficientes asociados a las restricciones sobre los pesos  $\mathbf{w}$ , mientras que  $\mathbf{d}$  es un vector que contiene el lado derecho de las restricciones.

En este proyecto se ha utilizado el optimizador SR3 para todos los casos, excepto en aquellos en los que se introducen restricciones sobre un coeficiente determinado, en estos casos se utilizará el optimizador SR3 restringido.

- **Diferenciador** (opcional, por defecto diferencia de derivadas finitas o *finite difference derivatives*): SINDy permite, como ya hemos indicado, recibir como argumento opcional la matriz de derivadas. Si no se le pasa dicha matriz, el algoritmo necesita calcularla, y utilizará para ello el diferenciador que le indiquemos en este argumento. Existen diferentes diferenciadores según la naturaleza de los datos (ruidosos o que presentan un paso de tiempo no uniforme). En el caso del método por defecto calcula las derivadas usando la aproximación de Taylor de primer orden. Otros métodos son *spline* [1] y *savitzky golay* [15].
- **Suposición inicial o *initial guess*** (opcional): matriz  $\in \mathbb{R}^{n \times p}$  que indica la suposición inicial a partir de la cual el optimizador debe empezar a buscar el valor de los pesos o matriz  $\Xi^T$ .
- **Múltiples trayectorias** (opcional, por defecto falso): indica si existen varias trayectorias para los datos presentes en la matriz  $\mathbf{X}$ . Estas trayectorias pueden variar sus condiciones iniciales o contener datos para distintos instantes temporales.

Es importante indicar que los propios creadores de SINDy indican que el sistema de referencia utilizado para la representación de los datos con los que alimentar al algoritmo debe de ser “afortunada”. Esto quiere decir que este sistema de referencia debe proveer un modelo dinámico disperso sin contener demasiados términos en la ecuación que modela el sistema. De lo contrario, es posible que el algoritmo no obtenga el resultado deseado. Asimismo, es necesario seleccionar una librería de funciones adecuada que modele correctamente la dinámica subyacente en los datos.

```

1 %% compute Sparse regression: sequential least squares
2 Xi = Theta\dXdT; % initial guess: Least-squares
3 % lambda is our sparsification knob.
4 for k=1:10
5     smallinds = (abs(Xi)<lambda); % find small coefficients
6     Xi(smallinds)=0; % and threshold
7     for ind = 1:n % n is state dimension
8         biginds = ~smallinds(:,ind);
9         % Regress dynamics onto remaining terms to find sparse Xi
10        Xi(biginds,ind) = Theta(:,biginds)\dXdT(:,ind);
11    end
12 end

```

Código 3.1: Ejemplo de implementación del algoritmo STLSQ en el lenguaje Matlab. Tomado de [7, code 1]

### 3.1. Aplicaciones sobre atractores

En esta sección se aplica la técnica de regresión dispersa propuesta por SINDy a los atractores extraños de Lorenz, Rössler, Thomas y *Three-Scroll Unified Chaotic System*. Sus ecuaciones y características han sido detallados en la subsección 2.1.1. Estos casos simples han sido desarrollados para la familiarización y comprensión del algoritmo.

En este caso se ha elegido como métrica el error cuadrático medio en los coeficientes de sus ecuaciones para la evaluación del algoritmo. Esta decisión ha sido tomada debido a que pequeños errores en los coeficientes de las ecuaciones pueden provocar errores grandes en las trayectorias (Teoría del Caos), por lo que medir el error en las trayectorias no es la

opción más adecuada.

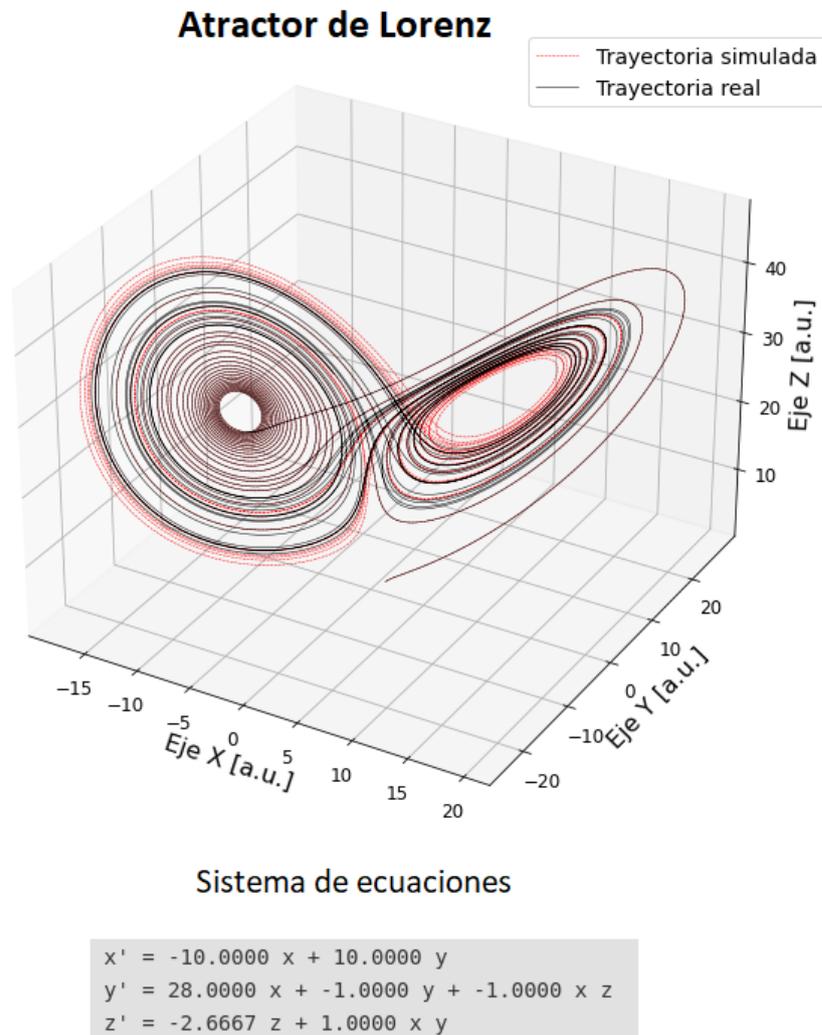


Figura 3.2: Representación del atractor de Lorenz. Ejes en unidades adimensionales [a.u.].

En términos generales, se observa que para todos los atractores, excepto el de Lorenz, ha sido necesario utilizar varias trayectorias para la correcta recuperación de las ecuaciones. Estas trayectorias han sido generadas para un número de muestras aleatorias entre 500 y 1000. Asimismo, los coeficientes utilizados para generar las trayectorias sintéticas de cada atractor son los indicados en la figura 2.4. Por último, la librería de funciones utilizada en todos los casos ha sido una polinómica de grado 2 que incluye términos constantes, excepto para el atractor de Thomas en el que ha sido necesario incluir también una librería trigonométrica de Fourier de 10 frecuencias. Para todos los casos se representa la trayectoria sintética utilizada para alimentar al algoritmo, la trayectoria simulada en

coordenadas cartesianas:  $x$ ,  $y$  y  $z$  y las ecuaciones obtenidas.

La trayectoria real sintética y la trayectoria simulada por el algoritmo para el atractor de Lorenz, junto con las ecuaciones obtenidas por el algoritmo de SINDy, puede observarse en la figura 3.2. Como se puede ver, el algoritmo es capaz de recuperar perfectamente las ecuaciones. Sin embargo, en determinadas partes de la trayectoria difieren la sintética y la simulada, posiblemente debido a la precisión interna del equipo, que ocasiona grandes diferencias entre las trayectorias.

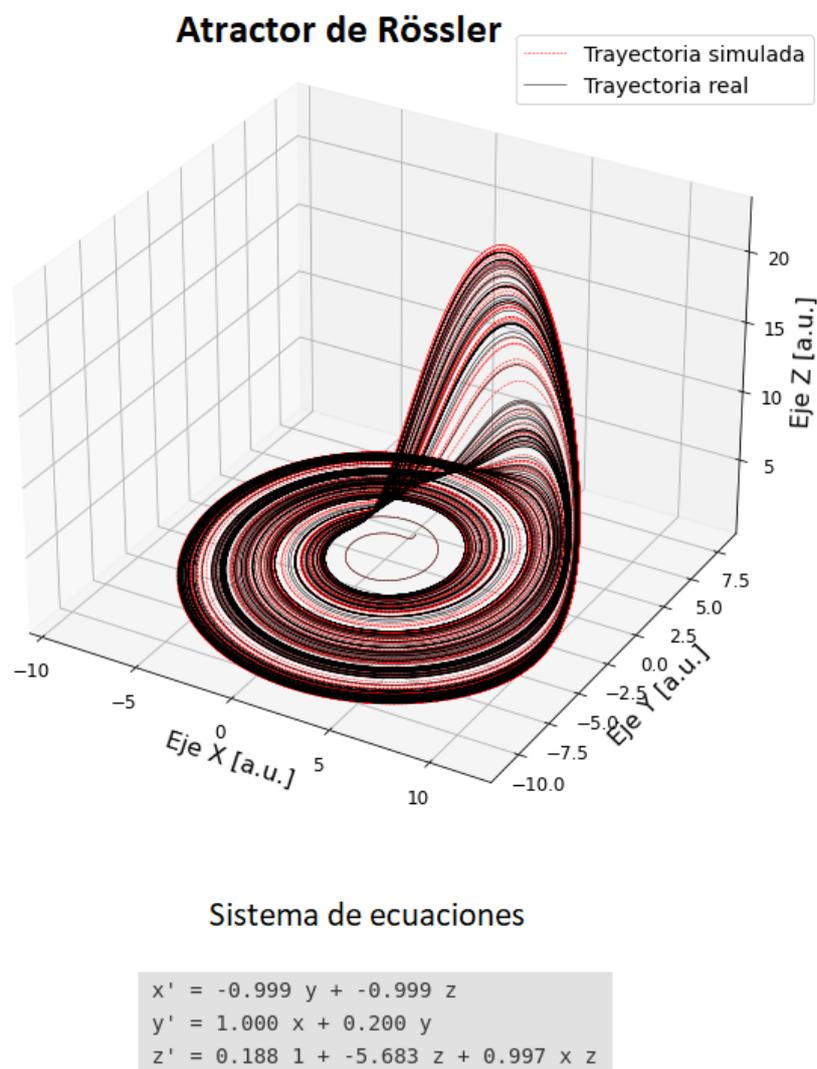


Figura 3.3: Representación del atractor de Rössler. Ejes en unidades adimensionales [a.u.].

En este caso, a diferencia del resto de atractores, no se han generado varias trayectorias, esto es debido a que los coeficientes de la ecuación recuperada para varias trayectorias

eran considerablemente diferentes a los utilizados para generar la trayectoria sintética. Asimismo, el paso de tiempo elegido para generar los datos con los que alimentar al algoritmo ha sido de 0.001.

En segundo lugar, para la obtención del atractor de Rössler ha sido necesario generar muestras con un paso de tiempo de 0.01 y, como se ha indicado anteriormente, incluir los datos de múltiples trayectorias. La representación de la trayectoria sintética (real), la simulada por el algoritmo y las ecuaciones obtenidas por el mismo se pueden observar en la figura 3.3.

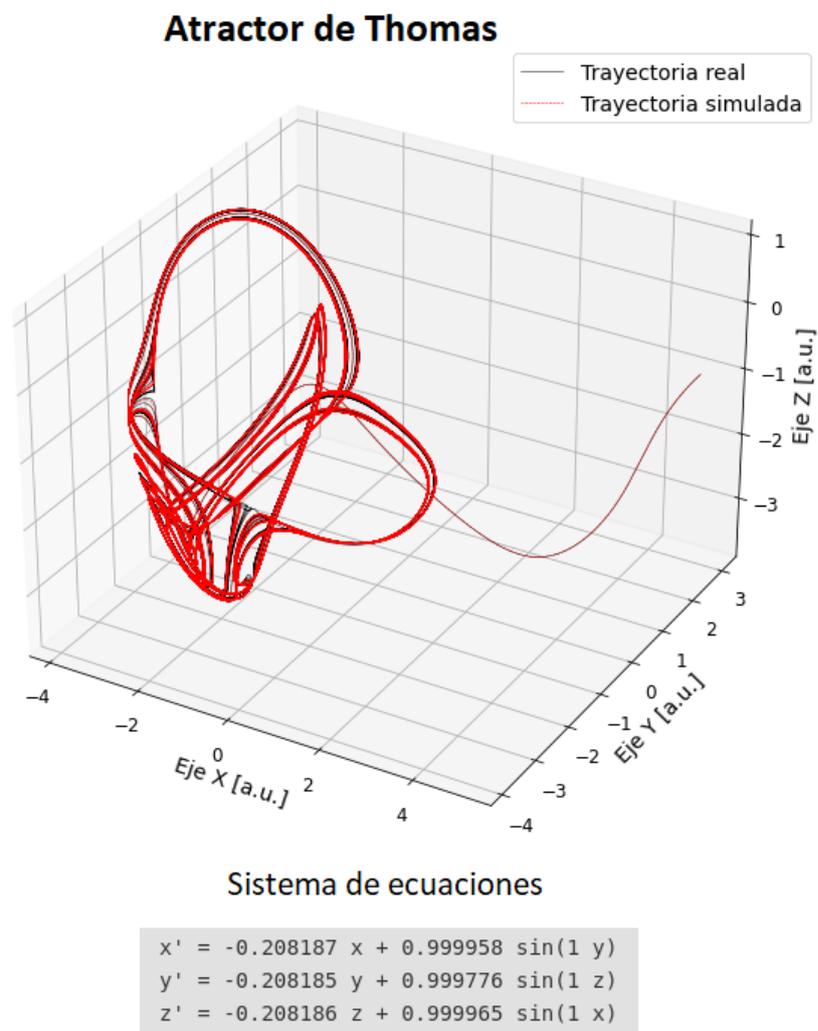


Figura 3.4: Representación del atractor de Thomas. Ejes en unidades adimensionales [a.u.].

En tercer lugar, para la obtención del atractor de Thomas se ha procedido de manera

similar que para el atractor anterior, excepto que la librería utilizada también incluye términos trigonométricos. De nuevo, ha sido necesario generar muestras con un paso de tiempo de 0.01 y, como se ha indicado anteriormente, incluir los datos de múltiples trayectorias. La representación de la trayectoria sintética (real), la simulada por el algoritmo y las ecuaciones obtenidas por el mismo se pueden observar en la figura 3.4.

A diferencia del resto de atractores, en el de Thomas se puede observar que la trayectoria sintética y simulada no se desplazan hacia el otro lóbulo del atractor incluso para instantes temporales elevados. Si se alimenta con condiciones iniciales distintas, la trayectoria puede desarrollarse en un lóbulo distinto, pero de nuevo no se desplaza.

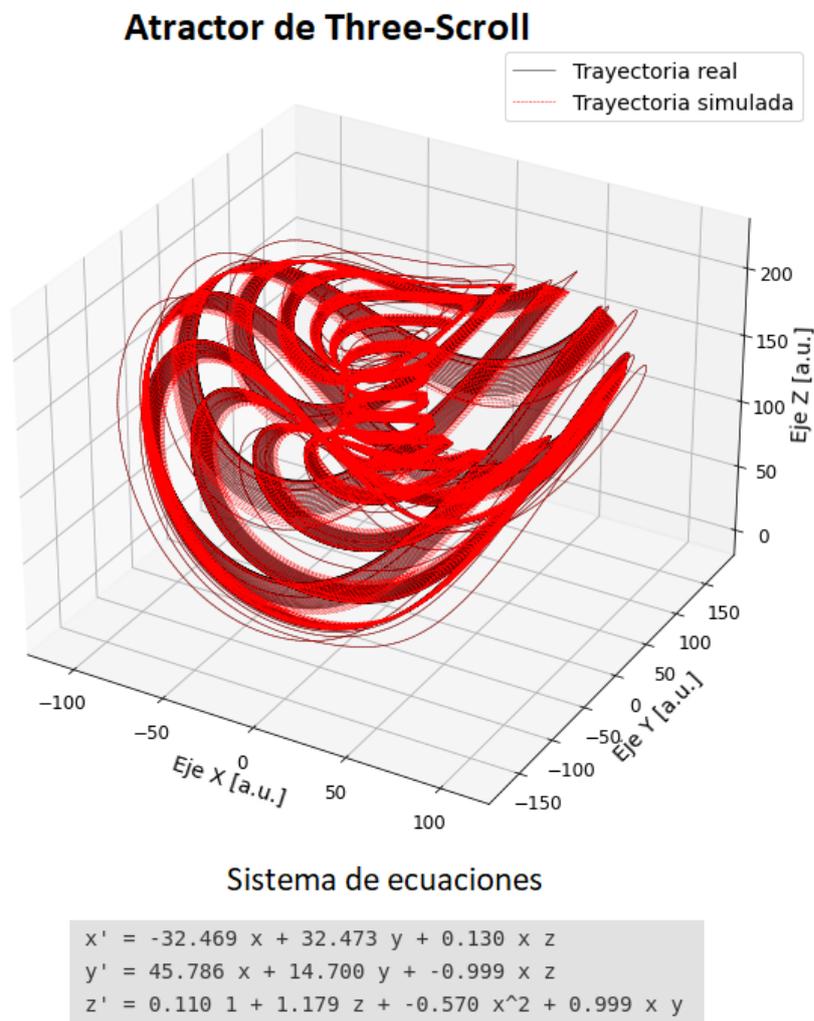


Figura 3.5: Representación del atractor de *Three-Scroll Unified Chaotic System*. Ejes en unidades adimensionales [a.u.].

Por último, para la obtención del atractor de *Three-Scroll Unified Chaotic System* ha sido necesario generar muestras con un paso de tiempo de 0.001 y, como se ha indicado anteriormente, incluir los datos de múltiples trayectorias. Se ha elegido este lapso de tiempo debido a la presencia del coeficiente  $d$  en la primera ecuación, el cual si se alimenta con un paso de tiempo superior el algoritmo no detecta este término y muestra errores muy altos en el coeficiente  $a$  de la primera ecuación. La representación de la trayectoria sintética (real), la simulada por el algoritmo y las ecuaciones obtenidas por el mismo se pueden observar en la figura 3.5.

Para concluir, en la figura 3.6 puede observarse el error cuadrático medio en los coeficientes de la ecuación para cada atractor. Se aprecia que el atractor *Three-Scroll Unified Chaotic System* es el que más error obtiene seguido del atractor de Rössler. En ambos casos es debido a que algunos de los coeficientes obtenidos por el algoritmo, figuras 3.5 y 3.3, difieren de su valor real, figura 2.4. A continuación, el atractor de Thomas es el que mayor error obtiene, puede ser debido a que, tal y como se puede apreciar en la figura 3.4, los coeficientes obtenidos en la ecuación varían en algunos de sus decimales con el valor real de los mismos. Por último, el que mejor resultado obtiene es el atractor de Lorenz, con una predicción prácticamente perfecta de sus coeficientes cuyo error puede ser debido a la precisión decimal de la máquina, figura 3.2.

En todos los casos, vemos que el error no supera el valor de  $10^{-4}$  que, teniendo en cuenta que estamos ante valores de  $p$  (número de funciones a considerar en la matriz  $\Xi \in \mathbb{R}^{p \times n}$ ) de 35, 10, 70 y 10 para los atractores de Lorenz, Rössler, Thomas y *Three-Scroll Unified Chaotic System*, respectivamente, se trata de un error muy bajo y el algoritmo SINDy es capaz de obtener las ecuaciones de forma fiable. Con ello concluimos que utilizar dicho algoritmo en trayectorias caóticas puede ser una línea de investigación muy prometedora.

En la tabla 3.1 se muestra de manera comparativa las ecuaciones obtenidas por el algoritmo y las ecuaciones teóricas asociadas, junto a los errores de sus coeficientes representados de manera gráfica en la figura 3.6.

Atractor	Ecuación a obtener	Ecuación obtenida	Error
<b>Lorenz</b>	$\frac{dx}{dt} = 10,0000(-x + y)$ $\frac{dy}{dt} = -xz + 28,0000x - y$ $\frac{dz}{dt} = xy - \frac{8}{3}z$	$\frac{dx}{dt} = 10,0000(-x + y)$ $\frac{dy}{dt} = -xz + 28,0000x - y$ $\frac{dz}{dt} = xy - 2,6667z$	$6,0855 * 10^{-31}$
<b>Rössler</b>	$\frac{dx}{dt} = -y - z$ $\frac{dy}{dt} = x + 0,2000y$ $\frac{dz}{dt} = 0,2000 + zx - 5,7000z$	$\frac{dx}{dt} = -0,9996y - 0,9992z$ $\frac{dy}{dt} = 0,9999x + 0,2000y$ $\frac{dz}{dt} = 0,1887 + 0,9970xz - 5,6831z$	$1,3532 * 10^{-3}$
<b>Thomas</b>	$\frac{dx}{dt} = \sin y - 0,208186x$ $\frac{dy}{dt} = \sin z - 0,208186y$ $\frac{dz}{dt} = \sin x - 0,208186z$	$\frac{dx}{dt} = 0,999956 \sin y - 0,208186x$ $\frac{dy}{dt} = 0,999822 \sin z - 0,208185y$ $\frac{dz}{dt} = 0,999964 \sin x - 0,208186z$	$1,1621 * 10^{-10}$
<b>Three- scroll Unified Chaotic System</b>	$\frac{dx}{dt} = 32,4800(y - x) + 0,1300xz$ $\frac{dy}{dt} = 45,8400x + 14,7000y - xz$ $\frac{dz}{dt} = 1,1800z + xy - 0,57x^2$	$\frac{dx}{dt} = -32,4701x + 32,4733y + 0,1299xz$ $\frac{dy}{dt} = 45,7977x + 14,6993y - 0,9994xz$ $\frac{dz}{dt} = 1,1799z + 0,9994xy - 0,5696x^2$	$4,3563 * 10^{-4}$

Tabla 3.1: Tabla comparativa entre las ecuaciones obtenidas por SINDy y las teóricas para el caso de atractores extraños.

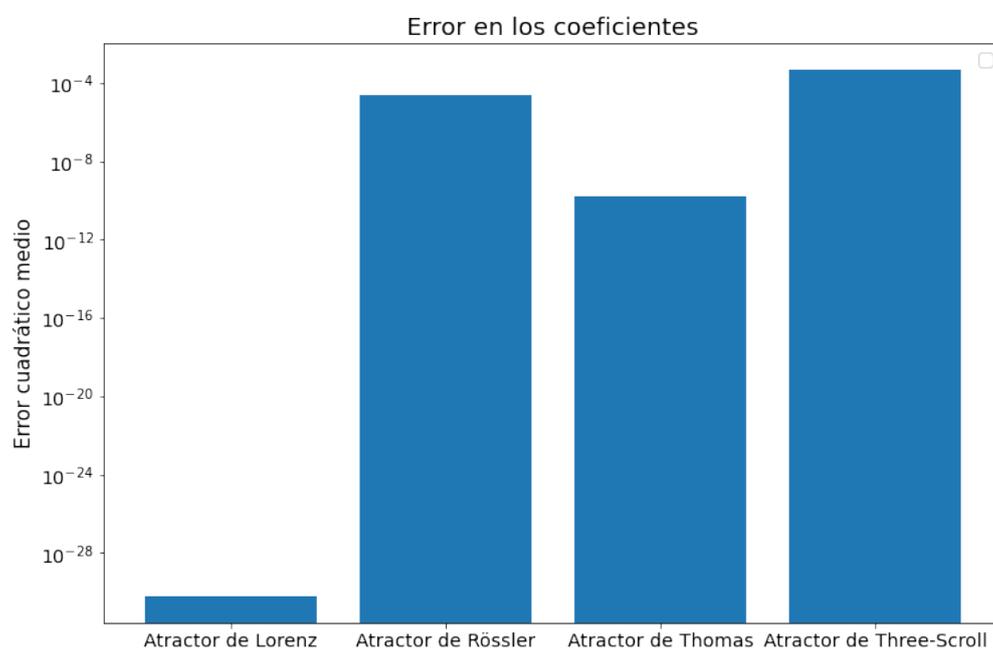


Figura 3.6: Error cuadrático medio en los coeficientes de la ecuación para los atractores de Lorenz, Rössler, Thomas y *Three-Scroll Unified Chaotic System*.

# Capítulo 4

## Arquitectura general

En este capítulo se detalla la estructura seguida para la implementación de los casos del próximo capítulo. En concreto, se detallan los pasos a seguir (junto con su implementación en pseudocódigo) y se adjunta una gráfica con el tiempo de ejecución de cada caso.

En el desarrollo de cada uno de los seis experimentos planteados se ha seguido una estructura similar:

1. **Obtención de las ecuaciones de la mecánica de vuelo y adimensionalización/normalización de las mismas** en ejes viento, ver subsección 2.2.1. Tal y como se detallará más adelante, eliminar las dimensiones o normalizar los valores entre 0 y 1 permite que el algoritmo calcule de manera más precisa los términos que intervienen en la ecuación.
2. **Generación de las trayectorias sintéticas:** en este paso se generan mediante software trayectorias sintéticas a partir de datos físicos de la aeronave o valores ya conocidos, como es la superficie alar o la densidad del aire. Para ello, se implementan las ecuaciones de la aeronave para unas condiciones iniciales aleatorias, con densidad de probabilidad uniforme en un rango que se especificará en cada caso. A continuación, se normalizan/adimensionalizan, según el caso, vía software (dividiendo entre la velocidad máxima) y se añade ruido. También se obtiene una trayectoria de validación que será utilizada para medir el error obtenido.
3. **Creación y entrenamiento del modelo de SINDy:** una vez generadas las trayectorias, se elige la librería de funciones a utilizar, el optimizador y, si es necesario,

las restricciones y suposición inicial. Por último, se entrena el modelo de SINDy para obtener las ecuaciones que mejor se adaptan a los datos proporcionados.

4. **Evaluación de resultados:** una vez obtenido el modelo que SINDy considera más adecuado se evalúan cualitativamente las ecuaciones obtenidas verificando que tienen los términos adecuados, se prueban las trayectorias obtenidas por dicho modelo obteniendo el error cuadrático medio entre la trayectoria proporcionada por el algoritmo y la sintética y, en algunos casos, se calcula el error cuadrático medio entre los coeficientes del modelo que obtiene SINDy y los valores esperados.
5. **Variación de las entradas y parámetros que el algoritmo permite configurar,** detalladas en el capítulo 3. El objetivo es comprobar cómo afecta la variación de estas entradas en el desempeño del modelo calculado.

```

1 t = generar_vector_tiempos(n_puntos , paso_tiempo)
2 coefs = coeficientes_caso()
3 # condiciones iniciales para las trayectorias de train aleatorias
4 cond_inic = condiciones_iniciales_aleatorias(n_var , n_trayec , rangos)
5 # condiciones iniciales para la trayectoria de validacion no aleatorias
6 cond_inic_val = [cond1 , cond2 , ...]
7 # generar los datos segun el caso
8 data_train , data_dot_train = generar_trayectoria(t , coefs , cond_inic ,
           n_trayec)
9 data_val , data_dot_val = generar_trayectoria(t , coefs , cond_inic_val , 1)
10 # normalizar los datos si es necesario
11 if normalizar:
12     data_train , data_dot_train = normalizar(data_train , data_dot_train)
13     data_val , _ = normalizar(data_val , data_dot_val)
14 # anadir ruido AWGN
15 if ruido:
16     data_train , data_dot_train = sumar_ruido(data_train , data_dot_train , pn
           , pn_dot)

```

Código 4.1: Pseudocódigo para la generación de las trayectorias.

En el código 4.1 se muestra el pseudocódigo que implementa el paso número 2 de la estructura anterior, en el cual se obtienen los datos con los que alimentar al algoritmo. En

primer lugar, se genera el vector de tiempos y coeficientes para el desarrollo del caso. En segundo lugar, se generan condiciones iniciales aleatorias con densidad de probabilidad uniforme, según el número de incógnitas y trayectorias elegidas, para los datos de entrenamiento, y no aleatorias (deterministas) para los datos de validación, utilizados para medir el error en la trayectoria. En tercer lugar y utilizando estas condiciones iniciales, los coeficientes de la ecuación, el vector de tiempos generado y especificando el número de trayectorias, se genera la matriz de datos,  $\mathbf{X}$  así como la de derivadas,  $\dot{\mathbf{X}}$ . Los siguientes pasos consisten en normalizar la trayectoria (dividiendo entre la velocidad máxima de la misma) y sumarle ruido, según el caso.

Para la generación del ruido se ha considerado ruido aditivo, blanco y gaussiano (AWGN en inglés) que es utilizado para contaminar la matriz  $\mathbf{X}$  y  $\dot{\mathbf{X}}$ . Para generarlo se ha normalizado dicho ruido con respecto a la potencia de la señal, de manera que la potencia del ruido representa  $pn$  veces la potencia de la señal original.

```

1 lib = crear_libreria_funciones()
2 matriz_real_coefs = crear_mat_coefs(data_train, coefs, lib)
3 # generar initial guess y restricciones en caso necesario
4 initial_guess, restric_lhs, restric_rhs = generar_prereq(data_train, lib,
5     n_restris, rhs)
6 # crear el modelo de SINDy
7 model_sindy = ps.SINDy(
8     optimizer=ps.sindy_opt(initial_guess=initial_guess,
9     lhs=restric_lhs, rhs=restric_rhs),
10    feature_library=lib,
11    feature_names=['var1', 'var2'...])
12 # entrenar el modelo de SINDy
13 model_sindy.fit(data_train, t=paso_tiempo,
14    multiple_trajectories=True, x_dot=data_dot_train)
15 # obtener una trayectoria para las ecuaciones calculadas
16 x_sim = model_sindy.simulate(cond_inic_val, t)
17 # calcular error en trayectoria y coeficientes
18 error_trayec = mean((x_sim-data_val)**2)
19 error_coefs = mean((model_sindy.coefficients()-matriz_real_coefs)**2)

```

Código 4.2: Pseudocódigo para el entrenamiento del modelo y validación de las ecuaciones obtenidas por SINDy.

En el código 4.2 se implementa en pseudocódigo los pasos 3 y 4 de la estructura anterior, los cuales se encargan de crear el modelo, entrenarlo y evaluar los resultados obtenidos. El primer lugar, se crea la librería de funciones a utilizar por el algoritmo para la obtención de las ecuaciones. Conocidas las funciones que componen la librería es posible obtener la matriz de coeficientes  $\Xi$  que debería obtener SINDy. En segundo lugar, se obtienen las restricciones y suposición inicial, si se requiere. En tercer lugar, se crea el modelo de SINDy mediante la sentencia `ps.SINDy`, donde `ps` es un alias para la librería PySINDy, especificando el optimizador, suposición inicial, restricciones, librería de funciones y nombre de las variables. En cuarto lugar, se alimenta al modelo con los datos generados en el código 4.1. A continuación, se calcula la trayectoria que simula el algoritmo para las ecuaciones que ha devuelto dada la condición inicial de la trayectoria de validación. Por último, se calcula el error cuadrático medio de los coeficientes de la ecuación y el error entre las trayectorias de validación real y simulada.

Con el fin de obtener resultados robustos que no dependan de parámetros internos aleatorios o de las condiciones iniciales generadas, en todos los experimentos del presente trabajo se han llevado a cabo diez simulaciones <sup>1</sup>, y posteriormente se obtiene la mediana entre esas diez ejecuciones.

Asimismo es conveniente indicar que toda la implementación llevada a cabo ha sido realizada en el lenguaje de programación Python, en la versión 3.8.10, y cada caso ha sido ejecutado en un servidor de 1TB de almacenamiento, 128 GB de memoria y un procesador Intel Xeon Gold 6130 a 2.10GHz con 32 núcleos. La versión utilizada de la librería PySINDy ha sido la 1.6.3.

Para concluir se adjunta la figura 4.1 en la que se indica el tiempo de ejecución de cada caso. Esta gráfica se ha generado utilizando 20 trayectorias en los datos de entrenamiento. Además, para aumentar la robustez de los resultados se ha repetido todo el proceso (creación de los datos, y del modelo, entrenamiento de este último y validación de resultados) en diez ocasiones y en la gráfica se reporta la media de estas diez ejecuciones. Tal y como se puede apreciar, el caso A (para más información sobre cada uno de los casos, consultar la tabla 5.1) es el menos tiempo tarda, pues es aquel con las ecuaciones

---

<sup>1</sup>En algunos casos concretos se ha utilizado mayor número de simulaciones debido a la variabilidad de los resultados en cada ejecución

más sencillas, mientras que los casos C-1 y C-2 son los que mayor tiempo necesitan, esto se debe a que utilizan un vector de tiempos con una resolución temporal mayor, o dicho de otra forma, un paso de tiempo entre muestras menor. Por otro lado, el caso D-1 es el siguiente que mayor tiempo necesita debido a que se trata de un caso cuyos términos de la ecuación son complejos, mientras que los casos B y D-2 obtienen un tiempo parecido, pues en todos ellos existe un número similar de términos en la ecuación.

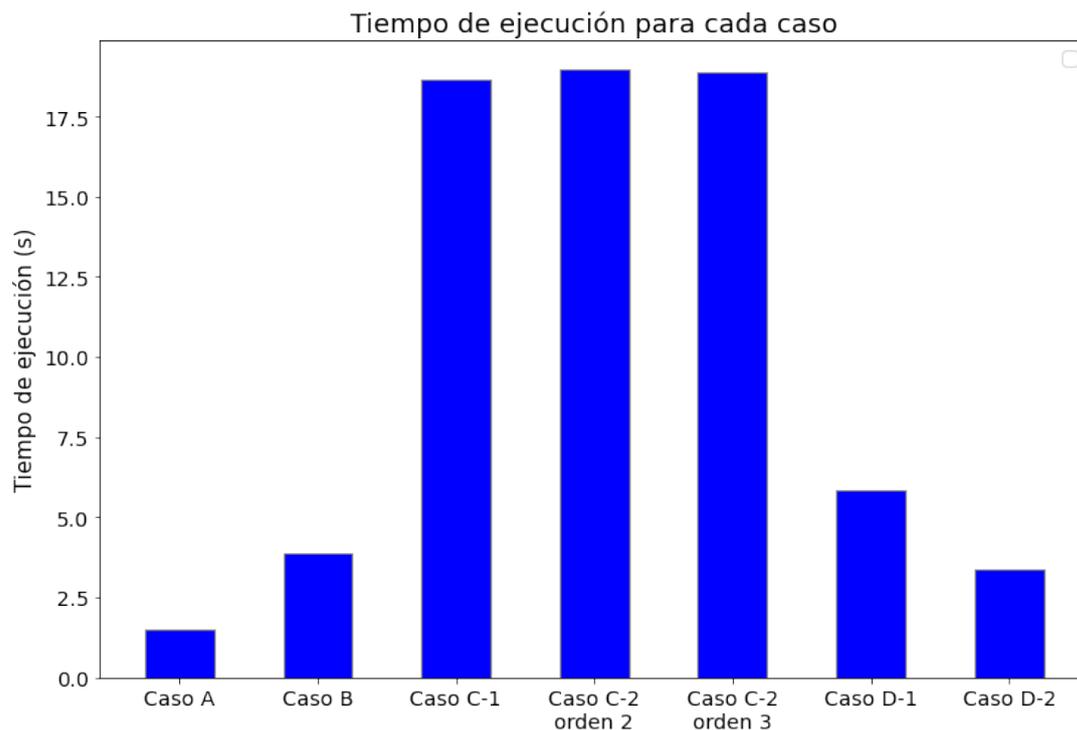


Figura 4.1: Tiempo de ejecución para cada uno de los casos desarrollados sobre aeronaves.



# Capítulo 5

## Resultados

En este capítulo se detallan los experimentos o casos implementados para comprobar la viabilidad del uso del algoritmo SINDy en aplicaciones relacionadas con la navegación de aeronaves y de manera genérica en el campo de la ingeniería aeroespacial. Para ello se sigue la estructura planteada en el capítulo 4. El índice de los casos desarrollados se encuentra en la tabla 5.1. Este capítulo concluye con un análisis entre las ecuaciones obtenidas y las teóricas y el error asociado a los coeficientes de las mismas.

Se ha planteado un conjunto de casos de complejidad incremental: en primer lugar, se parte del caso sencillo de un planeador (caso A), en el que no hay empuje, la velocidad y ángulo de asiento de la velocidad ( $\gamma$ ) son constantes y las únicas variables que cambian con el tiempo son la altura y distancia horizontal recorrida.

A continuación, con el fin de incrementar gradualmente la complejidad, en el caso B se modela un vuelo rectilíneo y simétrico de un turboreactor, el cual, a diferencia del planeador, tiene un término de empuje constante, la velocidad varía con el tiempo y se asumen valores grandes (con el fin de despreciar el término de resistencia aerodinámica inducida de la ecuación) y al ser un vuelo rectilíneo el ángulo de asiento de la velocidad es nulo, por lo que la aeronave no desciende como sí hacía en el caso anterior.

En el caso C la velocidad ya no se considera como valores grandes, y por tanto se mantiene el término de resistencia aerodinámica inducida (caso C-1). En el siguiente paso, caso C-2, se aproxima el término de  $1/v^2$  de estas ecuaciones por una aproximación en serie de Taylor.

Para concluir y con el objetivo de modelar otro movimiento que no sea rectilíneo, se

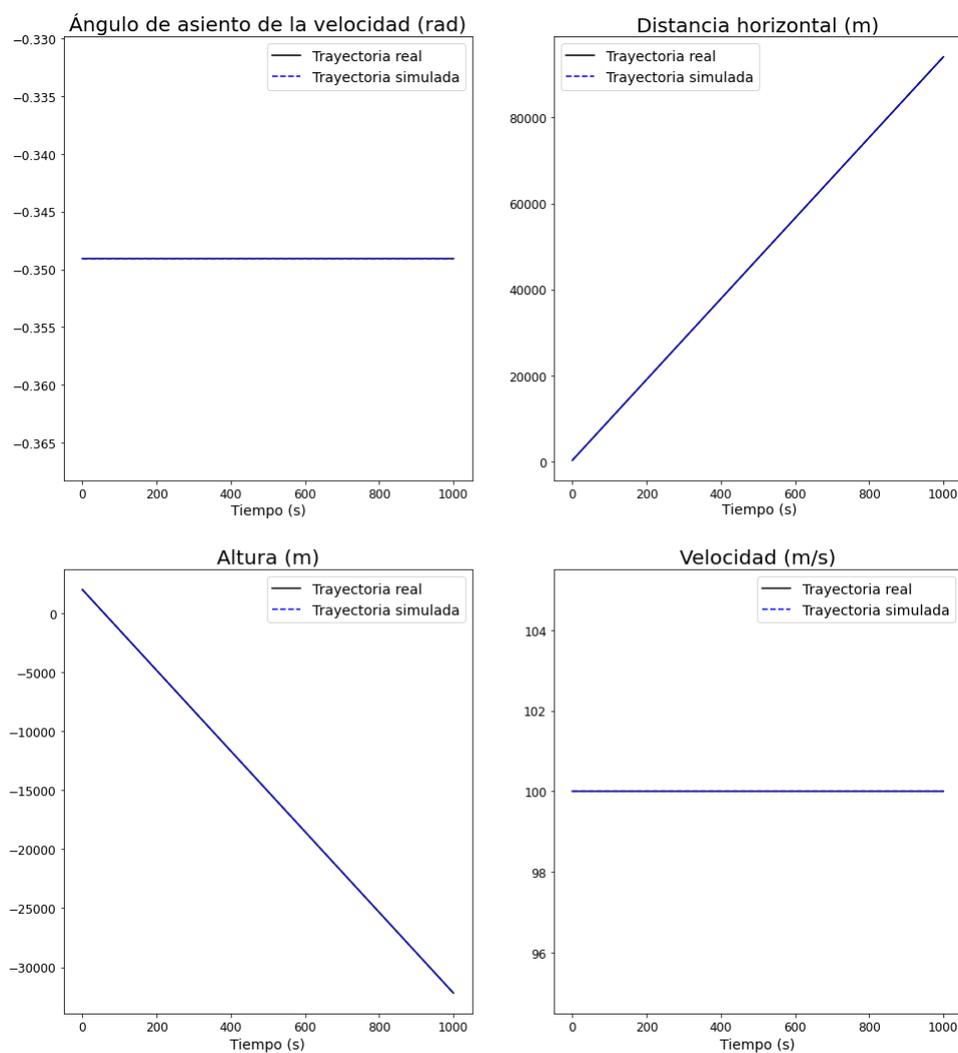
utiliza la misma aeronave (turborreactor) pero se le añade complejidad a su trayectoria, modelando un lazo en el plano vertical. Existen dos posibles configuraciones de vuelo para llevarlo a cabo: la primera consiste en asumir una velocidad y  $\gamma$  variables con el tiempo a empuje constante (caso D-1), mientras que la segunda se lleva a cabo asumiendo un empuje y ángulo de asiento de la velocidad variables con el tiempo a velocidad constante (caso D-2).

Caso general	Caso particular/Detalle	Apartado
<b>Planeador</b>	Sin empuje con ángulo de asiento de la velocidad constante	Caso A, sección 5.1
	Turborreactor en vuelo rectilíneo uniforme anulando el término de resistencia aerodinámica inducida por tratarse de velocidades grandes	Caso B, sección 5.2
<b>Turborreactor en vuelo rectilíneo</b>	Turborreactor en vuelo rectilíneo uniforme incluyendo dicho término de resistencia inducida por tratarse de velocidades pequeñas	Caso C-1, subsección 5.3.1
	Turborreactor en vuelo rectilíneo uniforme aproximando uno de los términos de la ecuación por una serie de Taylor	Caso C-2, subsección 5.3.2
<b>Lazo ideal en el plano vertical</b>	Lazo ideal en el plano vertical para velocidad y ángulo de asiento de la velocidad variables y empuje constante	Caso D-1, subsección 5.4.1
	Lazo ideal en el plano vertical para empuje y ángulo de asiento de la velocidad variables y velocidad constante	Caso D-2, subsección 5.4.2

Tabla 5.1: Índice de los casos sobre aeronaves desarrollados en el proyecto.

## 5.1. Caso A: Planeador en descenso uniforme

Este primer caso es el más sencillo que se ha desarrollado y el objetivo es familiarizarse con la implementación de las ecuaciones de la mecánica de vuelo vía software y entender el funcionamiento del algoritmo. En este caso no se adimensionalizan o normalizan las ecuaciones debido a su simplicidad.



Sistema de ecuaciones

```
gamma'= 0.000
x'= 1.000 v*cos(gamma)
h'= 1.000 v*sen(gamma)
v'= 0.000
```

Figura 5.1: Caso A. Representación de las variables que intervienen en la trayectoria de un planeador.

Se pretende implementar el vuelo de un planeador (sin empuje) con un ángulo de asiento de la velocidad  $\gamma$  y velocidad aerodinámica uniformes. Este es el único caso en el que se obtienen las ecuaciones cinemáticas representadas en la ecuación (2.6). Se ha decidido llevar a cabo este enfoque por ilustrar un ejemplo de implementación de estas ecuaciones y no de las dinámicas.

La evolución de la trayectoria con el tiempo se muestra en la figura 5.1, donde el eje  $x$  representa el tiempo y el eje  $y$  cada una de las variables que intervienen en la ecuación: ángulo de asiento de la velocidad, distancia horizontal recorrida, altura descendida y velocidad. Asimismo, se incluyen las ecuaciones obtenidas por el algoritmo alimentando al mismo con 10 trayectorias, utilizando el argumento múltiples trayectorias detallado en el capítulo 3. Tal y como se puede apreciar, el algoritmo de SINDy es capaz de aproximar perfectamente estas trayectorias simples.

En la figura 5.2 se muestra la evolución del error cuadrático medio en la trayectoria según el número de trayectorias utilizadas para alimentar al algoritmo. Como se puede ver utilizar el parámetro opcional que proporciona al modelo los valores de los que partir (suposición inicial) resulta beneficioso para un número de trayectorias bajo, ya que si no se obtiene la solución trivial de todo ceros. Sin embargo, utilizar este parámetro puede no implicar una gran mejoría en los resultados, tal y como se muestra en los próximos casos.

Análisis sobre el número de trayectorias

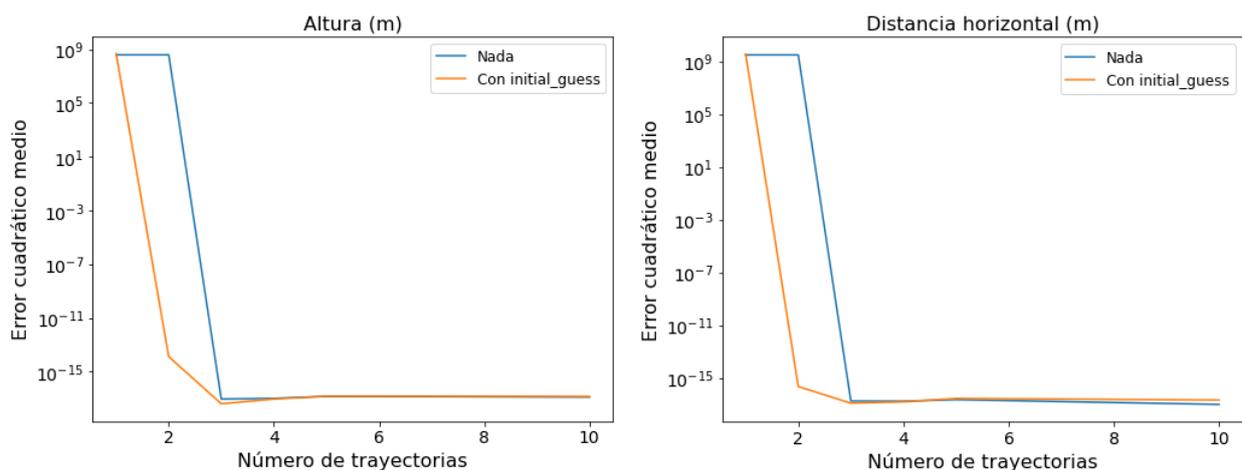


Figura 5.2: Caso A. Representación del error cuadrático medio de la trayectoria frente al número de trayectorias.

## 5.2. Caso B: Vuelo rectilíneo simétrico para velocidades grandes

En este caso se modela un vuelo con empuje orientado según el sentido que indica el eje  $x_w$  y describiendo un vuelo rectilíneo, uniforme y simétrico, por lo que el ángulo de asiento de la velocidad,  $\gamma$ , es nulo. Por tanto, partiendo de las ecuaciones dinámicas (2.5) se obtienen

$$\text{Eje } z_w : L = W \quad (5.1)$$

$$\text{Eje } x_w : -D + T = m\dot{V} \quad (5.2)$$

$$L = \frac{1}{2}\rho_{aire}V^2Sc_L \quad (5.3)$$

$$D = \frac{1}{2}\rho_{aire}V^2S(c_{D0} + kc_L^2). \quad (5.4)$$

Sustituyendo  $L$  por su expresión (ver ecuación (5.3)) se despeja de la ecuación del eje  $z_w$  el coeficiente de sustentación,  $c_L$ . A continuación, se incluye la expresión de  $D$  (ver ecuación (5.4)) en la ecuación del eje  $x_w$  y se sustituye el  $c_L$  en el término de resistencia aerodinámica inducida.

$$D = \frac{1}{2}\rho_{aire}V^2Sc_{D0} + k\frac{W^2}{\frac{1}{2}\rho_{aire}V^2S} \quad (5.5)$$

$$\dot{V} = \frac{T}{m} - \frac{1}{2}\rho_{aire}V^2\frac{S}{m}c_{D0} - k\frac{g^2}{\frac{1}{2}\rho_{aire}V^2\frac{S}{m}}, \quad (5.6)$$

donde  $c_{D0}$  representa el coeficiente de resistencia aerodinámica parásita,  $S$  la superficie alar,  $k$  término de proporcionalidad de la polar,  $\rho_{aire}$  es la densidad atmosférica,  $g$  la aceleración de la gravedad y  $V$  la velocidad aerodinámica de la aeronave. De esta forma, se ha pasado de dos incógnitas,  $c_L$  y  $V$ , y dos ecuaciones dinámicas, a una ecuación, concretamente (5.6), y una incógnita,  $V$ .

Sustituyendo por constantes que representen los términos invariantes con el tiempo y asumiendo velocidades aerodinámicas grandes, se desprecia el término de resistencia aerodinámica inducida de la ecuación (5.6),

$$A = \frac{1}{2}\rho_{aire}\frac{S}{m}c_{D0} \quad (5.7)$$

$$C = \frac{T}{m} \quad (5.8)$$

$$\dot{V} = -AV^2 + C. \quad (5.9)$$

Adimensionalizando la velocidad por la siguiente relación  $\hat{V} = V/U_c$  y sustituyendo en la ecuación anterior, se obtiene

$$\hat{V} = \frac{dV}{dt} \frac{1}{U_c} = \frac{\dot{V}}{U_c} \quad (5.10)$$

$$U_c \hat{V} = -AU_c^2 \hat{V}^2 + C \quad (5.11)$$

$$\hat{V} = -AU_c \hat{V}^2 + \frac{C}{U_c}. \quad (5.12)$$

Por tanto, si se alimenta al algoritmo con una librería de funciones que incluya términos constantes y cuadráticos, los coeficientes a buscar serían

$$a = AU_c = \frac{1}{2} \rho_{aire} \frac{S}{m} c_{D0} U_c \quad (5.13)$$

$$c = \frac{C}{U_c} = \frac{T}{mU_c} \quad (5.14)$$

$$\hat{V} = -a\hat{V}^2 + c. \quad (5.15)$$

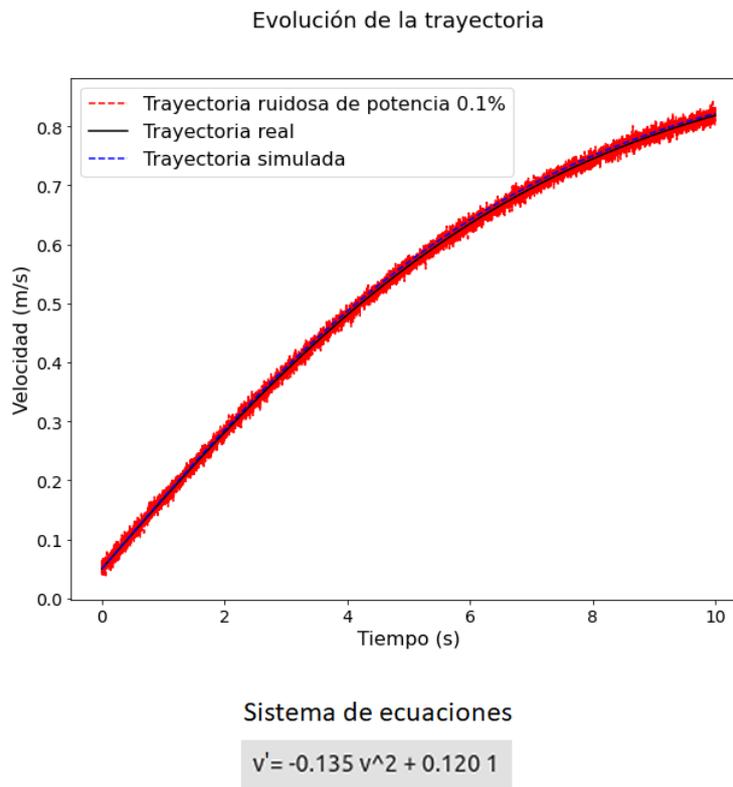


Figura 5.3: Caso B. Representación de la velocidad de la trayectoria frente al tiempo.

En este caso se ha elegido como valor de  $U_c$  el valor máximo de la velocidad. Por tanto, se alimenta al algoritmo con trayectorias de velocidad cuyos valores se encuentran

## 5.2. CASO B: VUELO RECTILÍNEO SIMÉTRICO PARA VELOCIDADES GRANDES<sup>51</sup>

entre 0 y 1, tal y como se puede ver en la figura 5.3. Asimismo, se incluye la ecuación obtenida por el algoritmo alimentando al mismo con 10 trayectorias. Además, en esta figura se representan tres trayectorias: la real generada sintéticamente, la simulada por el algoritmo, y una trayectoria ruidosa sintética con la que se alimenta al algoritmo en algunos casos que se analizarán a continuación cuya potencia de ruido es un 0.1% la potencia de la señal sin contaminar. Tal y como se puede ver el algoritmo es capaz de recuperar totalmente la trayectoria incluso con ruido en ella.

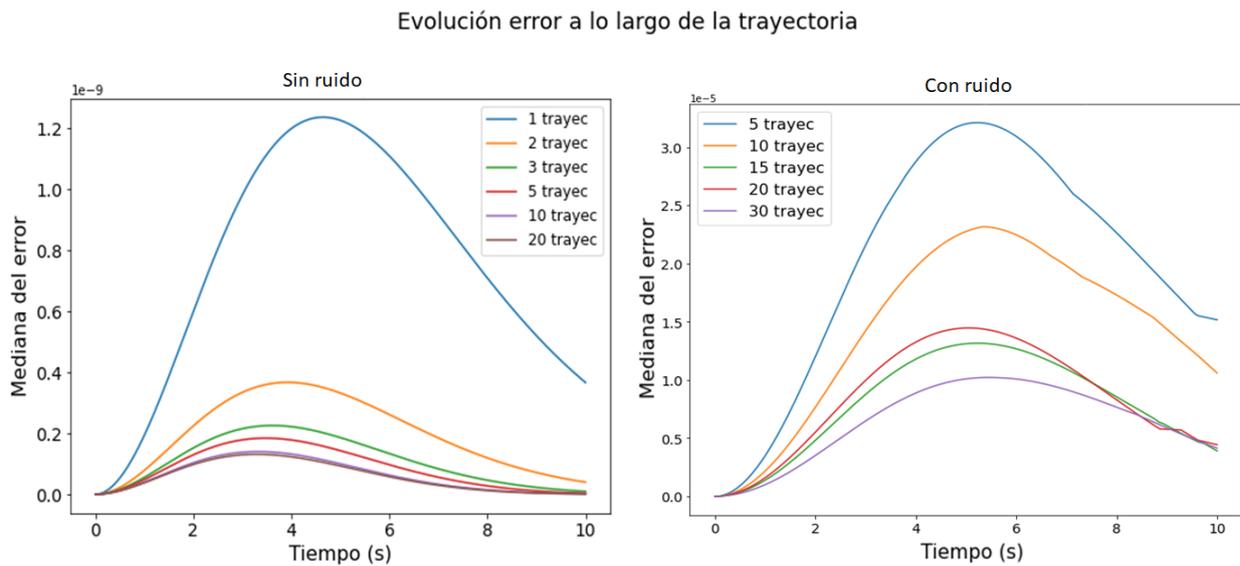


Figura 5.4: Caso B. Representación del error cuadrático de la trayectoria frente al tiempo sin ruido (izquierda) y con ruido (derecha).

En la figura 5.4 se ilustra el error cuadrático frente al tiempo. En el caso de la gráfica de la derecha se contamina la trayectoria real con la que alimentará al algoritmo con ruido que supone un 0,1% de potencia de la señal sintética. Como se puede ver el error para ambos casos y todas las trayectorias alcanza un valor máximo entorno a 4 segundos, esto puede ser debido a que inicialmente el algoritmo aproxima la trayectoria por una función lineal, pero a medida que avanza el tiempo observa se trata de una parábola y, por tanto, reduce el error para los instantes de tiempo finales. Asimismo, es conveniente destacar el hecho de que incrementar el número de trayectorias mejora considerablemente el error en ambos casos, sin ruido se reduce de  $1,2 \times 10^{-9}$  (para 1 trayectoria) a  $10^{-10}$  (para 10 trayectorias) utilizando tan solo 9 trayectorias más (un orden de magnitud inferior).

Los resultados son análogos a aquellos en los que se contamina la trayectoria con ruido, excepto que en eje  $y$  la escala es de  $10^{-5}$ .

Por último, se representa el error tanto en la trayectoria como en los coeficientes de la ecuación (5.15) en la figura 5.5 variando el número de trayectorias utilizando una librería polinómica de grado 5, con la intención de mostrar que, aunque SINDy reciba gran cantidad de funciones que no son relevantes, es capaz de obtener únicamente aquellas que describen los datos recibidos. Como se puede observar, el error en ambas métricas sigue la misma tendencia, y presenta valores muy pequeños por debajo de  $10^{-9}$ . Esto nos indica que el algoritmo es capaz de obtener de manera precisa una ecuación muy similar a la teórica, (5.15), lo que ocasiona que el error en la trayectoria sea muy pequeño, tal y como comentábamos anteriormente al ver la figura 5.3.

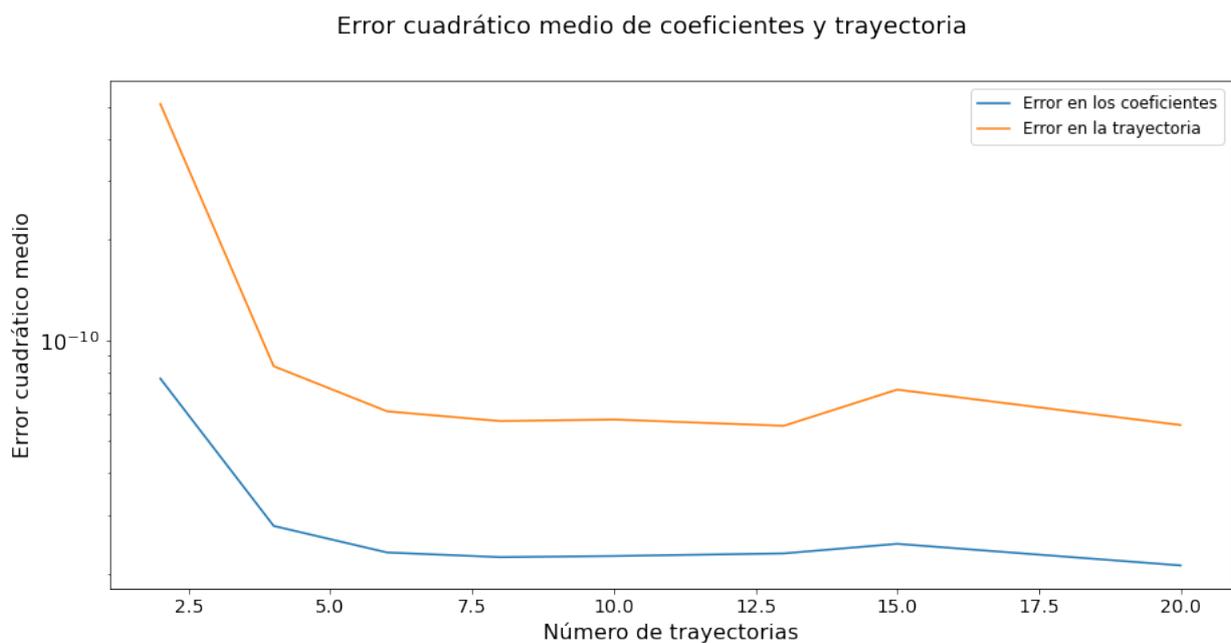


Figura 5.5: Caso B. Representación del error cuadrático medio de la trayectoria y de los coeficientes de la ecuación (5.15).

### 5.3. Caso C: Vuelo rectilíneo simétrico para velocidades pequeñas

En este caso, a diferencia del anterior, no se desprecia el término de la resistencia aerodinámica inducida debido a que se consideran velocidades pequeñas, para ello las trayectorias han sido generadas para valores de velocidad de 5-5.5 m/s en el caso C-1 y 5-15 m/s en el caso C-2. De esta manera las ecuaciones se derivan de (5.6) incluyendo el término de resistencia inducida, y siguiendo un procedimiento similar al caso anterior se obtiene

$$U_c \hat{V} = -AU_c^2 \hat{V}^2 + C - \frac{B}{\hat{V}^2 U_c^2} \quad (5.16)$$

$$\hat{V} = -AU_c \hat{V}^2 + \frac{C}{U_c} - \frac{B}{\hat{V}^2 U_c^3} \quad (5.17)$$

$$B = k \frac{g^2}{\frac{1}{2} \rho_{aire} \frac{S}{m}} \quad (5.18)$$

En este caso los valores de  $A$  y  $C$  siguen las expresiones anteriores (5.7) y (5.8), respectivamente, y el valor de  $B$  es el indicado en la ecuación (5.18). De la expresión anterior se simplifica a

$$b = \frac{B}{U_c^3} \quad (5.19)$$

$$\hat{V} = -a \hat{V}^2 + c - \frac{b}{\hat{V}^2}, \quad (5.20)$$

donde los coeficientes  $a$  y  $c$  siguen las ecuaciones (5.13) y (5.14), respectivamente.

De nuevo se ha elegido como valor de  $U_c$  el valor máximo de la velocidad. Por tanto, se alimenta al algoritmo con trayectorias de velocidad cuyos valores se encuentran entre 0 y 1.

#### 5.3.1. Caso C-1: Vuelo rectilíneo simétrico para velocidades pequeñas

En este caso se evaluará la precisión del algoritmo ante trayectorias sintéticas que siguen la ecuación (5.20). En la figura 5.6 se representa la evolución de la velocidad con el tiempo. Tal y como se puede ver a pesar de introducir un ruido que representa el 10%

la potencia de la señal, el algoritmo es bastante preciso y consigue recuperar la ecuación buscada. Esto parece indicar que SINDy es robusto ante trayectorias ruidosas, que es lo que nos encontraríamos en un caso real. En dicha figura también se incluye la ecuación que obtiene el algoritmo alimentando al mismo con 25 trayectorias.

Es conveniente destacar que este caso es especialmente complejo debido a la presencia de los términos  $\hat{V}^2$  y  $1/\hat{V}^2$  en la misma ecuación, concretamente en (5.20). Inicialmente, se trató de utilizar dos grupos de trayectorias, uno con valores pequeños que activasen el término de  $1/\hat{V}^2$  y el otro con valores grandes que activasen el término de  $\hat{V}^2$ . Sin embargo, el algoritmo aproximaba esta ecuación mediante una hipérbola. A continuación, se intentó utilizar una restricción para que el término de  $1/\hat{V}^2$  apareciese en los resultados, pero esto ocasionaba que los coeficientes obtenidos en el resto de términos de la ecuación fuesen muy distintos del valor real de los mismos.

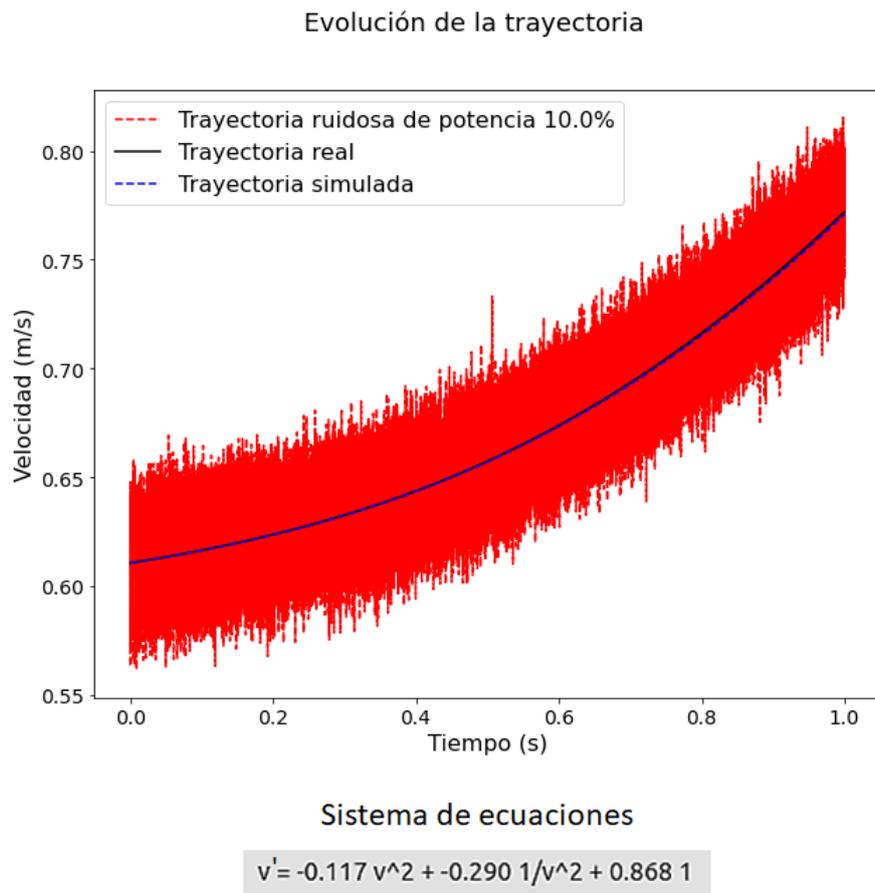


Figura 5.6: Caso C-1. Representación de la velocidad de la trayectoria frente al tiempo.

Finalmente, la solución aportada es la siguiente, el valor del coeficiente  $B$  es un orden

de magnitud de  $10^5$  veces superior que el de  $A$  debido a los parámetros elegidos, tanto físicos como la superficie alar o atmosféricos como la densidad del aire. Esto ocasiona que el algoritmo no sea capaz de percibir el término de  $\hat{V}^2$ , por lo que la solución aportada es multiplicar el coeficiente  $A$  por un valor de 10. Gracias a esta decisión y al hecho de normalizar la ecuación, se ha conseguido que ambos términos apareciesen en la solución que el algoritmo proporciona.

Tal y como se puede observar en la figura 5.7 incrementar el número de trayectorias resulta beneficioso para la disminución del error. Sin embargo, a partir de 5 trayectorias la mejora en el error es menor. Asimismo, es conveniente destacar que tan solo utilizar 4 trayectorias más (comparando las líneas de 1 y 5 trayectorias) el error disminuye considerablemente. Por último, se puede observar como para 5, 15 y 25 trayectorias no se observa una mejora significativa al incluir nuevas trayectorias.

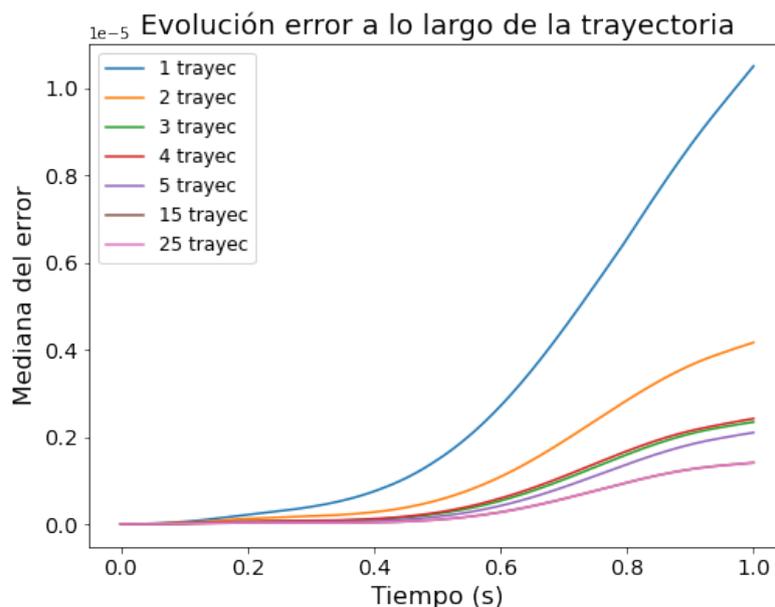


Figura 5.7: Caso C-1. Representación del error cuadrático de la trayectoria frente al tiempo.

Por otro lado, en la figura 5.8 se muestra la evolución del error de la trayectoria frente a la potencia de ruido con la que se contaminan las trayectorias utilizadas para alimentar el algoritmo. De nuevo, se puede observar claramente como incrementar el número de trayectorias resulta beneficioso y, tal como cabría esperar, a medida que crece la potencia de ruido el error es mayor. Tal y como se puede ver en la gráfica, para valores altos de

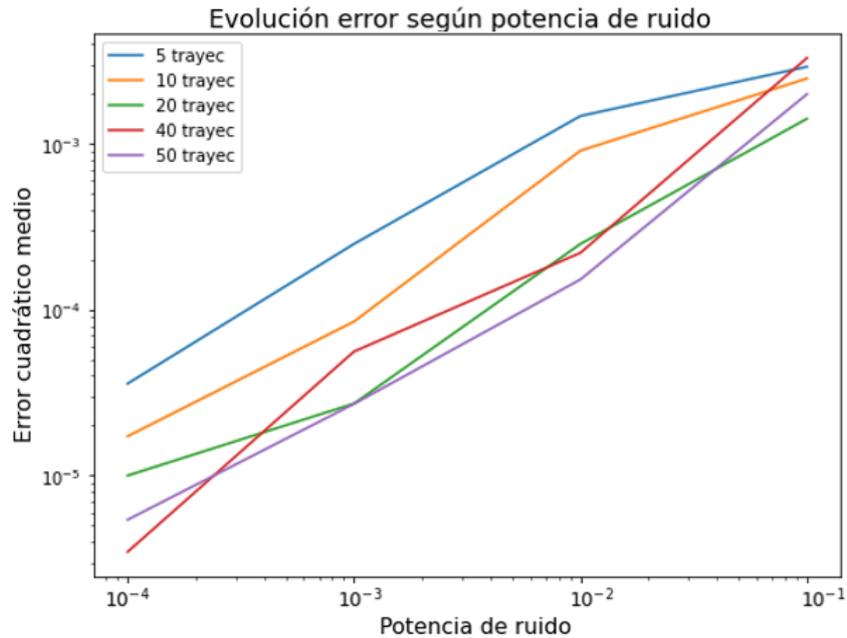


Figura 5.8: Caso C-1. Representación del error cuadrático medio de la trayectoria frente a la potencia de ruido.

potencia de ruido tener un mayor número de trayectorias puede llegar a resultar perjudicial (alimentar el algoritmo con datos que describen 40 y 50 trayectorias funcionan peor que el modelo que recibe 20 para una potencia de ruido de  $10^{-1}$ , por ejemplo). Esto puede ser debido a que mayor número de trayectorias implican mayor número de muestras ruidosas y por tanto, el algoritmo puede equivocarse con mayor facilidad. A pesar de ello, los resultados obtenidos mantienen un error por debajo de  $10^{-2}$  incluso para un ruido que representa un 10 % de la potencia de la señal.

En la figura 5.9 se representa el error cuadrático medio de las trayectorias y coeficientes de la ecuación (5.20) frente al número de trayectorias para una librería polinómica de grado 5 y un término  $1/V^2$ . Sin embargo, esta aproximación resulta bastante apropiada, pues el error en la trayectoria es pequeño, y disminuye aún más conforme se incrementa el número de trayectorias, a pesar de que los coeficientes de la ecuación obtenida no sean los adecuados. Por último, es conveniente destacar que incrementar el número de trayectorias no mejora el error de los coeficientes de la ecuación, esto puede ser debido a que un mayor número de trayectorias implica un mayor número de muestras a analizar y por tanto que sea más complejo obtener un resultado preciso que se adapte a todas las trayectorias

### 5.3. CASO C: VUELO RECTILÍNEO SIMÉTRICO PARA VELOCIDADES PEQUEÑAS<sup>57</sup>

utilizadas.

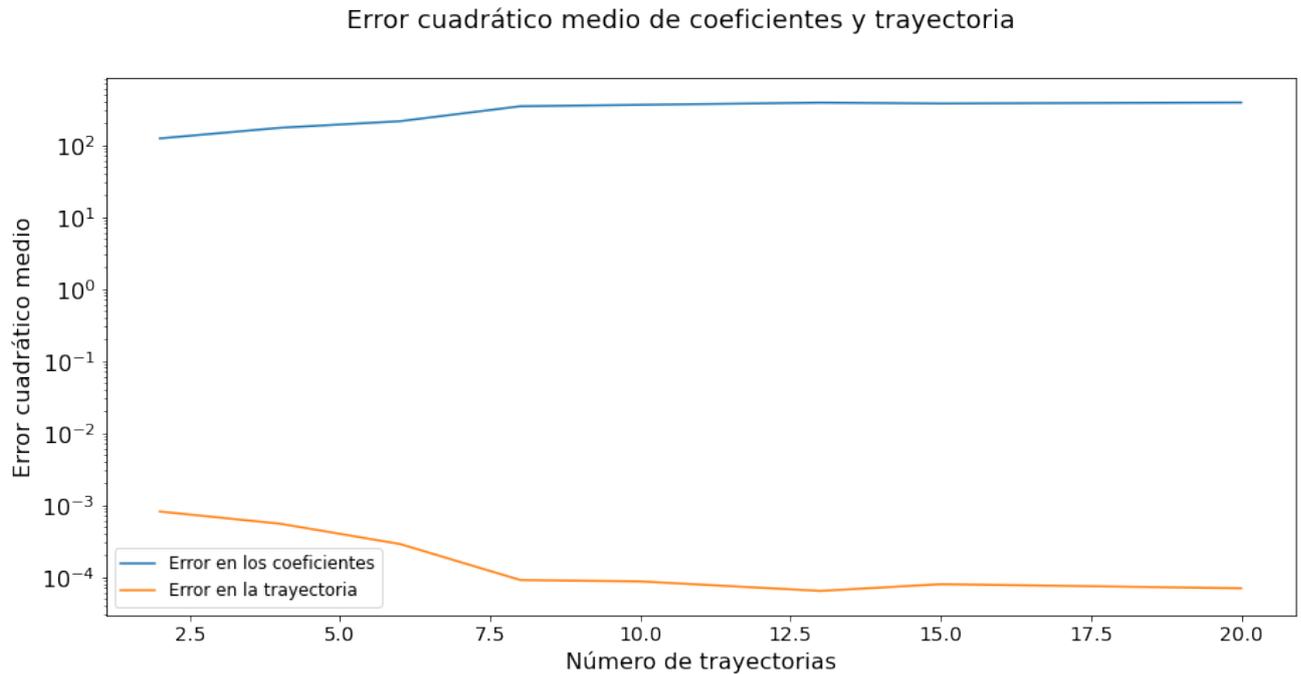


Figura 5.9: Caso C-1. Representación del error cuadrático medio de la trayectoria y de los coeficientes de la ecuación (5.20).

Por último, en la figura 5.10 se representa el error cuadrático medio de los coeficientes de la ecuación (5.20) frente a la potencia de ruido utilizada para contaminar la matriz de datos  $\mathbf{X}$ . Como se puede observar y de manera similar a la figura 5.8 incrementar la potencia de ruido aumenta el error. Sin embargo, podemos observar que el error se mantiene dentro de unos valores razonables, ya que a pesar de introducir una potencia de ruido que representa el 10% de la potencia de la señal original, el error se mantiene entorno a 1 para 25 trayectorias, es decir, los coeficientes varían, en media, una unidad respecto a su valor real.

#### 5.3.2. Caso C-2: Aproximación en serie de Taylor de un vuelo rectilíneo simétrico para velocidades pequeñas

En este caso se pretende que el algoritmo obtenga la ecuación (5.20) aproximando el término  $1/\hat{V}^2$  por un desarrollo en serie de Taylor de orden 2 y 3, es decir, este caso evalúa

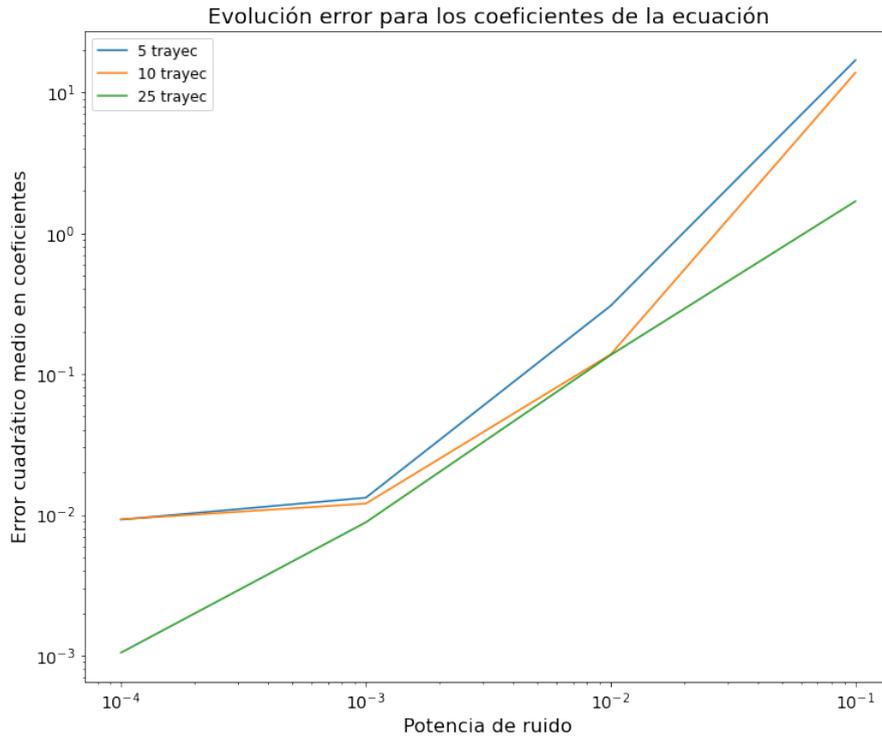


Figura 5.10: Caso C-1. Representación del error cuadrático medio de los coeficientes de la ecuación (5.20) frente a la potencia de ruido.

la capacidad del algoritmo de obtener ecuaciones dinámicas simplificadas que modelen el comportamiento de la aeronave. Para ello, se alimentará dicho algoritmo con trayectorias que siguen la ecuación (5.20).

El desarrollo en serie de Taylor aproxima funciones mediante polinomios de distinto grado, términos factoriales y derivadas de la función a aproximar. Este desarrollo se realiza en torno a un punto fijado,  $x_0$ , matemáticamente [34]

$$f(x) = \sum_{n=0}^{\infty} \frac{1}{n!} \left. \frac{d^n f(x)}{dx^n} \right|_{x=x_0} (x - x_0)^n \quad (5.21)$$

$$f(x) = f(x_0) + \left. \frac{df(x)}{dx} \right|_{x=x_0} (x - x_0) + \left. \frac{d^2 f(x)}{2! dx^2} \right|_{x=x_0} (x - x_0)^2 + \dots,$$

donde si limitamos el número de términos del sumatorio a un valor fijo  $N$ , obtendríamos una versión simplificada del desarrollo. Este valor se conoce como el orden de la serie. En este caso se ha elegido como centro del desarrollo  $x_0 = 1$ , de manera que la aproximación en serie de Taylor para  $N = 2$  sería

$$\frac{1}{v^2} = 1 + \frac{-2}{1!}(v - 1) + \frac{6}{2!}(v - 1)^2 = 3v^2 - 8v + 6.$$

Retomando la ecuación (5.20) el desarrollo de orden 2 obtenido sería

$$\begin{aligned}\hat{V} &= -a\hat{V}^2 + c - 3b\hat{V}^2 + 8b\hat{V} - 6b \\ \hat{V} &= -(a + 3b)\hat{V}^2 + 8b\hat{V} + (-6b + c).\end{aligned}\tag{5.22}$$

Para  $N = 3$  sería

$$\frac{1}{v^2} = 1 + \frac{-2}{1!}(v - 1) + \frac{6}{2!}(v - 1)^2 + \frac{-24}{3!}(v - 1)^3 = -4v^3 + 15v^2 - 20v + 10.$$

Retomando la ecuación (5.20) se obtiene

$$\begin{aligned}\hat{V} &= -a\hat{V}^2 + c - b(-4\hat{V}^3 + 15\hat{V}^2 - 20\hat{V} + 10) \\ \hat{V} &= 4b\hat{V}^3 - (a + 15b)\hat{V}^2 + 20b\hat{V} + (-10b + c).\end{aligned}\tag{5.23}$$

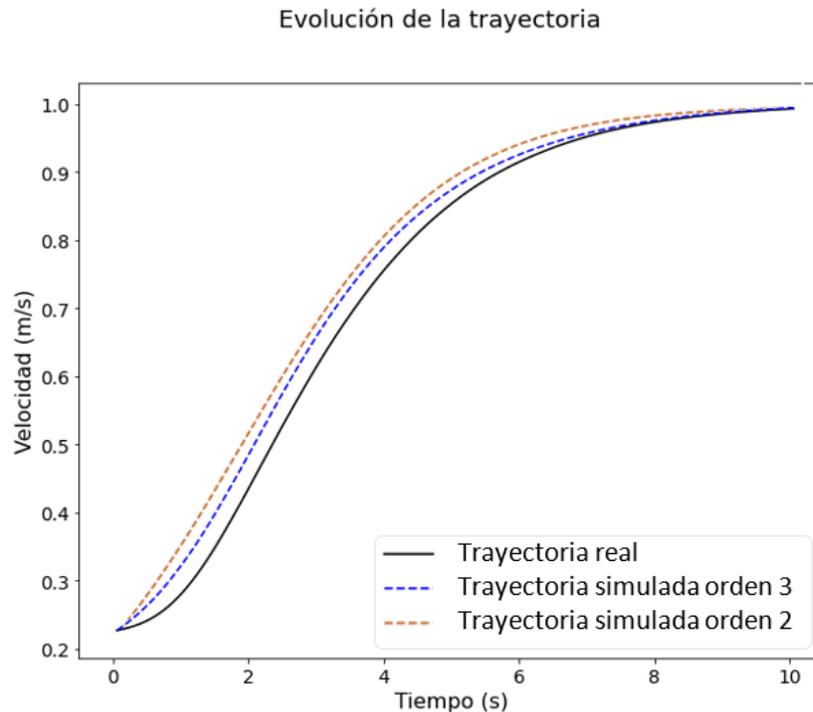
Para ambas aproximaciones los coeficientes  $a$ ,  $b$  y  $c$  siguen las ecuaciones (5.13), (5.19) y (5.14), respectivamente.

Utilizando una librería polinómica de grado 2 y 3, y alimentando al algoritmo con trayectorias que se rigen por la ecuación (5.20) se obtienen las trayectorias representadas en la figura 5.11. Como se puede observar, la aproximación realizada por el algoritmo es bastante precisa, de forma general simula correctamente la trayectoria, en especial en la fase de ascenso (aproximadamente lineal) y en la fase estacionaria de la misma. Como resulta razonable, la aproximación de grado 3 resulta más precisa que la de grado 2. Asimismo, se incluyen las ecuaciones obtenidas por el algoritmo según la aproximación considerada alimentando al mismo con 20 trayectorias.

Por otro lado, en la figura 5.12 se muestra el error cuadrático medio para ambas aproximaciones frente al número de trayectorias utilizadas. Tal y como se puede observar, al incrementar el número de trayectorias mejora el error para una aproximación de orden 3, mientras que orden 2 no mejora a partir de 7 trayectorias. Esto puede ser debido a que una aproximación de tercer orden contiene más términos sobre los que cambiar los coeficientes asociados y así ajustar mejor la trayectoria.

Otro aspecto a analizar es utilizar tiempos de validación mayores que los usados para entrenar/alimentar el algoritmo. Se evalúa si el algoritmo ofrece un desempeño igual de bueno ante tiempos que no ha visto con anterioridad.

En la figura 5.13 se analiza este aspecto y, tal y como se puede observar, la aproximación de orden 3 ofrece un error menor para el mismo número de trayectorias que de orden



#### Sistema de ecuaciones

Orden 2  $v' = 0.674 v + -0.677 v^2$

Orden 3  $v' = -0.248 1 + 1.971 v + -2.713 v^2 + 0.993 v^3$

Figura 5.11: Caso C-2. Representación de la velocidad de la trayectoria frente al tiempo para una aproximación en serie de Taylor de orden 2 y 3.

2. De forma general se observa que a medida que crece el tiempo de validación el error disminuye, esto puede ser debido a que la trayectoria alcanza un estado estacionario, tal y como se puede apreciar en la figura 5.11, el cual es más sencillo de modelar. Como vemos, el algoritmo obtiene una simplificación de la ecuación (5.20) bastante precisa tanto para un estado transitorio como para un estado estacionario, con errores inferiores a  $10^{-3}$ .

Por último, en la figura 5.14 se pretende analizar la precisión del algoritmo, para una aproximación en serie de Taylor de tercer orden, ante trayectorias con distinto número de muestras y para diferentes instantes temporales. El resultado obtenido es que incrementar el número de muestras utilizadas resulta beneficioso y reduce el error, pero debe complementarse con utilizar trayectorias lo suficientemente largas temporalmente. Incluso utilizando vectores temporales pequeños y pocas muestras el error se mantiene en un valor de  $10^{-3}$ .

### 5.3. CASO C: VUELO RECTILÍNEO SIMÉTRICO PARA VELOCIDADES PEQUEÑAS61

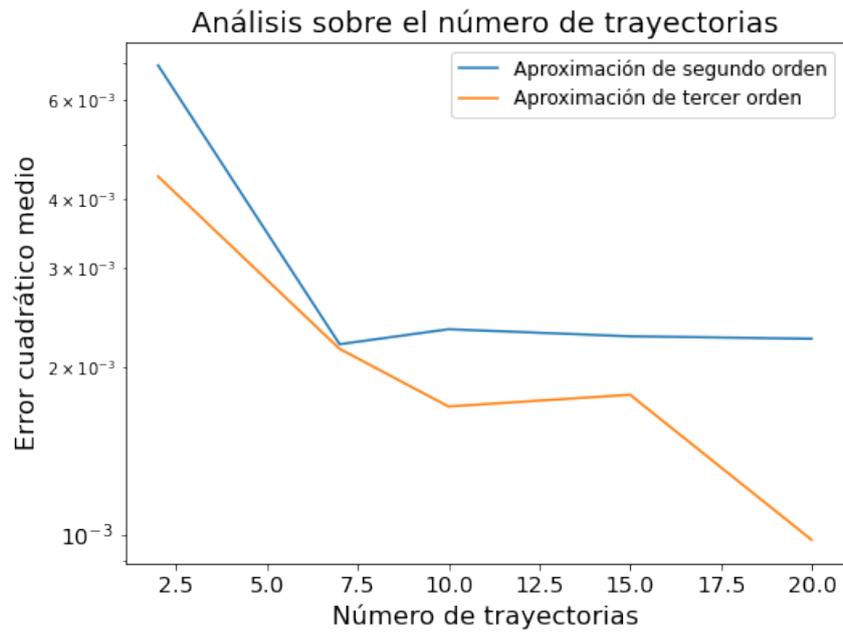


Figura 5.12: Caso C-2. Representación del error cuadrático medio de la trayectoria frente al número de trayectorias.

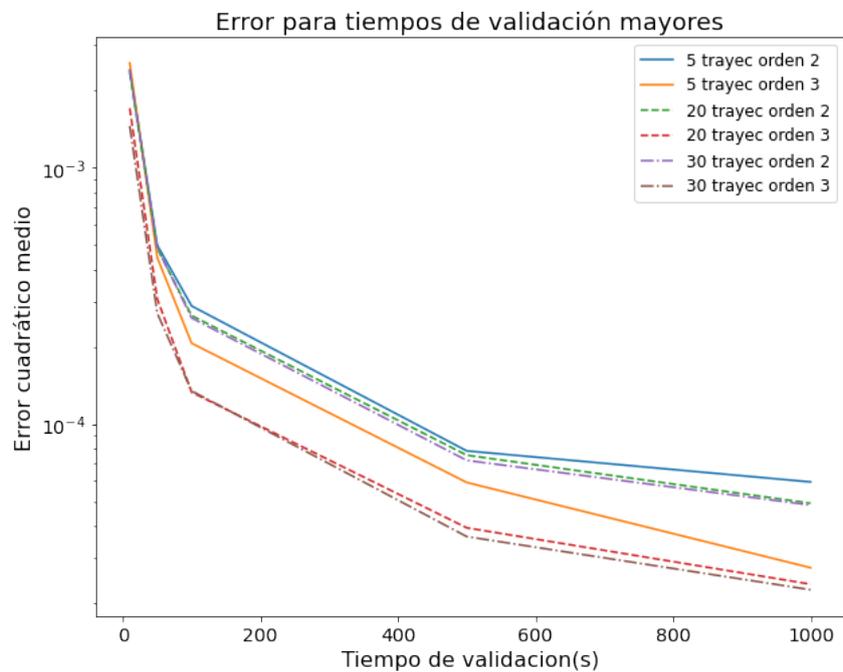


Figura 5.13: Caso C-2. Representación del error cuadrático medio de la trayectoria frente al la duración temporal de la trayectoria de validación.

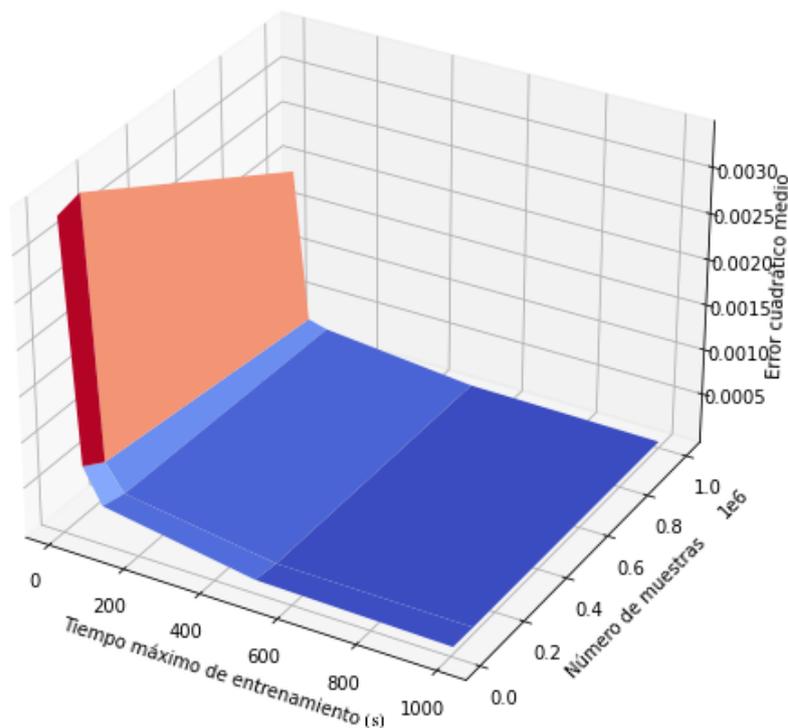


Figura 5.14: Caso C-2. Representación del error cuadrático medio de la trayectoria variando parámetros del vector de tiempo.

## 5.4. Caso D: Lazo ideal en el plano vertical

En este caso se pretende que el algoritmo calcule las ecuaciones asociadas a un lazo ideal en el plano vertical simétrico. El procedimiento general es el explicado a continuación, aunque en cada caso se detallarán los pasos exactos a seguir. Se parte de las ecuaciones (2.5) y se derivan dos casos, por un lado se mantiene el empuje constante, caso D-1, y por otro se mantiene constante la velocidad, caso D-2. En ambos, el ángulo de asiento de la velocidad,  $\gamma$  varía según se desarrolla la trayectoria, cuya variación se describe según la ecuación diferencial

$$\dot{\gamma} = \frac{V}{R}, \quad (5.24)$$

donde  $R$  representa el radio del lazo y es un valor dado del problema. Este movimiento queda representado por tres incógnitas ( $c_L$ ,  $V$  o  $T$  y  $\gamma$ ) y dos ecuaciones dinámicas (2.5), (se desarrollarán en cada caso) y una angular (5.24). Además, se considera que el cambio en la densidad del aire es despreciable y se fija el valor de este parámetro.

Debido a la complejidad de este movimiento se sustituye la ecuación (5.3) en la tercera ecuación dinámica (2.5) y se despeja el  $c_L$ . A continuación en la primera ecuación dinámica se sustituye la resistencia aerodinámica por la expresión (5.4) y en el valor de la resistencia aerodinámica inducida se sustituye por la expresión anteriormente despejada del  $c_L$ . Las ecuaciones se reducen a dos (una dinámica y una angular) y las incógnitas dependientes del tiempo serían  $V$  o  $T$  y  $\gamma$ .

### 5.4.1. Caso D-1: Lazo ideal en el plano vertical para velocidad y ángulo de asiento de la velocidad variables

Primeramente, se ha tratado de alimentar al algoritmo con trayectorias normalizadas. Sin embargo, los resultados obtenidos no han sido satisfactorios y es necesario realizar otras transformaciones a las ecuaciones. Para ello, se adimensionaliza la ecuación de la velocidad por un tiempo característico  $t_c$  y una velocidad característica  $U_c$ .

Se mantiene la definición  $\hat{V} = V/U_c$  y se incorporan las siguientes

$$\hat{V} = \frac{\dot{V}t_c}{U_c} \quad (5.25)$$

$$\hat{\gamma} = \dot{\gamma}t_c. \quad (5.26)$$

Nótese que  $\hat{V}$  carece de unidades, mientras que la ecuación angular que define  $\hat{\gamma}$  mantiene unidades adimensionales de radianes.

Sustituyendo las expresiones anteriores en las ecuaciones (2.5) y (5.24) se obtienen

$$\frac{\hat{V}}{t_c} = \frac{T}{mU_c} - \frac{1}{2}\rho_{aire}\frac{S}{m}(c_{D0} + kc_L^2)U_c\hat{V}^2 - \frac{g \sin \gamma}{U_c} \quad (5.27)$$

$$\hat{V}\hat{\gamma} = \frac{1}{2}\rho_{aire}\frac{S}{m}c_L U_c\hat{V}^2 - \frac{g \cos \gamma}{U_c} \quad (5.28)$$

$$\frac{\hat{\gamma}}{t_c} = \hat{V}\frac{U_c}{R}. \quad (5.29)$$

Tal y como hemos hecho en el caso anterior, definiremos una serie de coeficientes cons-

tantes para simplificar la representación de las ecuaciones

$$A_1 = \frac{1}{2}\rho_{aire}\frac{S}{m}U_c t_c \quad (5.30)$$

$$A_2 = t_c \frac{g}{U_c} \quad (5.31)$$

$$A_3 = \frac{T}{mU_c} t_c \quad (5.32)$$

$$A_4 = t_c \frac{U_c}{R}. \quad (5.33)$$

Con el fin de simplificar términos y disminuir la complejidad de las ecuaciones se eligen como tiempo y velocidad característicos

$$t_c = \frac{U_c}{g} \quad (5.34)$$

$$U_c = \sqrt{\frac{2gm}{S\rho_{aire}}}. \quad (5.35)$$

De esta forma los coeficientes anteriores se simplifican

$$A_1 = A_2 = 1 \quad (5.36)$$

$$A_3 = \frac{T}{mg} \quad (5.37)$$

$$A_4 = \frac{2m}{\rho_{aire}SR}. \quad (5.38)$$

Sustituyendo estos coeficientes en las ecuaciones (5.27), (5.28) y (5.29)

$$\hat{V} = A_3 - (c_{D0} + kc_L^2)\hat{V}^2 - \sin \gamma \quad (5.39)$$

$$\hat{V}\hat{\gamma} = \frac{c_L\hat{V}^2}{t_c} - \frac{\cos \gamma}{t_c} \quad (5.40)$$

$$\hat{\gamma} = A_4\hat{V}, \quad (5.41)$$

despejando el  $c_L$  de la segunda ecuación y sustituyendo en la primera se obtiene

$$c_L = \frac{\hat{V}\hat{\gamma}t_c}{\hat{V}^2} + \frac{\cos \gamma}{\hat{V}^2} = A_4 + \frac{\cos \gamma}{\hat{V}^2} \quad (5.42)$$

$$\hat{V} = A_3 - (c_{D0} + kA_4^2)\hat{V}^2 - 2kA_4 \cos \gamma - \sin \gamma - \frac{k(\cos \gamma)^2}{\hat{V}^2}. \quad (5.43)$$

Agrupando términos, las ecuaciones a obtener por el algoritmo son

$$\hat{V} = A_3 - A_5\hat{V}^2 - A_6 \cos \gamma - A_7 \sin \gamma - \frac{A_8(\cos \gamma)^2}{\hat{V}^2} \quad (5.44)$$

$$\hat{\gamma} = A_4\hat{V},$$

donde

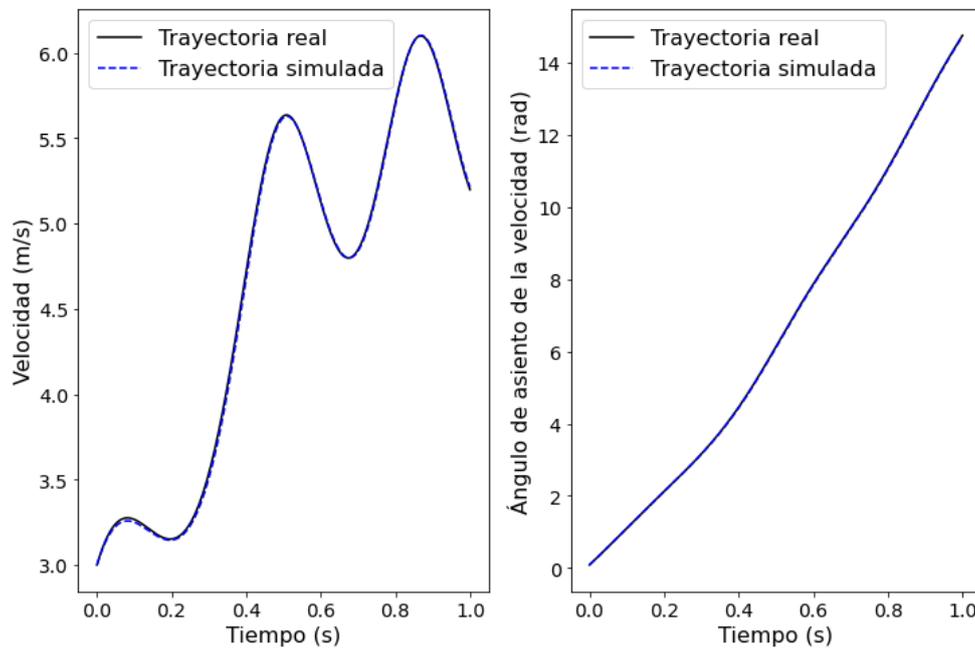
$$A_5 = c_{D0} + kA_4^2 \quad (5.45)$$

$$A_6 = 2kA_4 \quad (5.46)$$

$$A_7 = 1 \quad (5.47)$$

$$A_8 = k. \quad (5.48)$$

Evolución de la trayectoria



Sistema de ecuaciones

$$\begin{aligned} v' &= -0.374 v^2 + -10.005 \sin(\gamma) + -0.448 \cos(\gamma) + -4.865 \cos(\gamma)^2/v^2 + 6.176 \cdot 1 + 6.176 \cdot 1 \\ \gamma &= 3.175 v \end{aligned}$$

Figura 5.15: Caso D-1. Representación de la velocidad y ángulo de asiento de la velocidad ( $\gamma$ ) de la trayectoria frente al tiempo.

Inicialmente se ha tratado de alimentar al algoritmo con datos de trayectorias que se rigen por las ecuaciones (5.44). Sin embargo, algunos de los términos con coeficientes más pequeños no eran detectados por SINDy, por esa razón se ha indicado en el parámetro de entrada que indica al algoritmo el paso de tiempo a utilizar en el entrenamiento un valor de 0.0001s en lugar de 0.001s (siendo este último el utilizado para generar los datos), con lo que se consigue que el algoritmo detecte todos los términos de la ecuación. Sin embargo,

los coeficientes obtenidos por el algoritmo de SINDy tendrán un valor 10 veces superior a su valor real en las ecuaciones (5.44) y (5.41), por lo que necesitamos reajustarlos a la hora de generar las trayectorias utilizando el modelo obtenido. En la figura 5.15 se representa la evolución de la velocidad y ángulo de asiento de la velocidad ( $\gamma$ ) en función del tiempo junto a la trayectoria obtenida por el algoritmo. Asimismo se incluyen las ecuaciones obtenidas alimentando al mismo con 10 trayectorias.

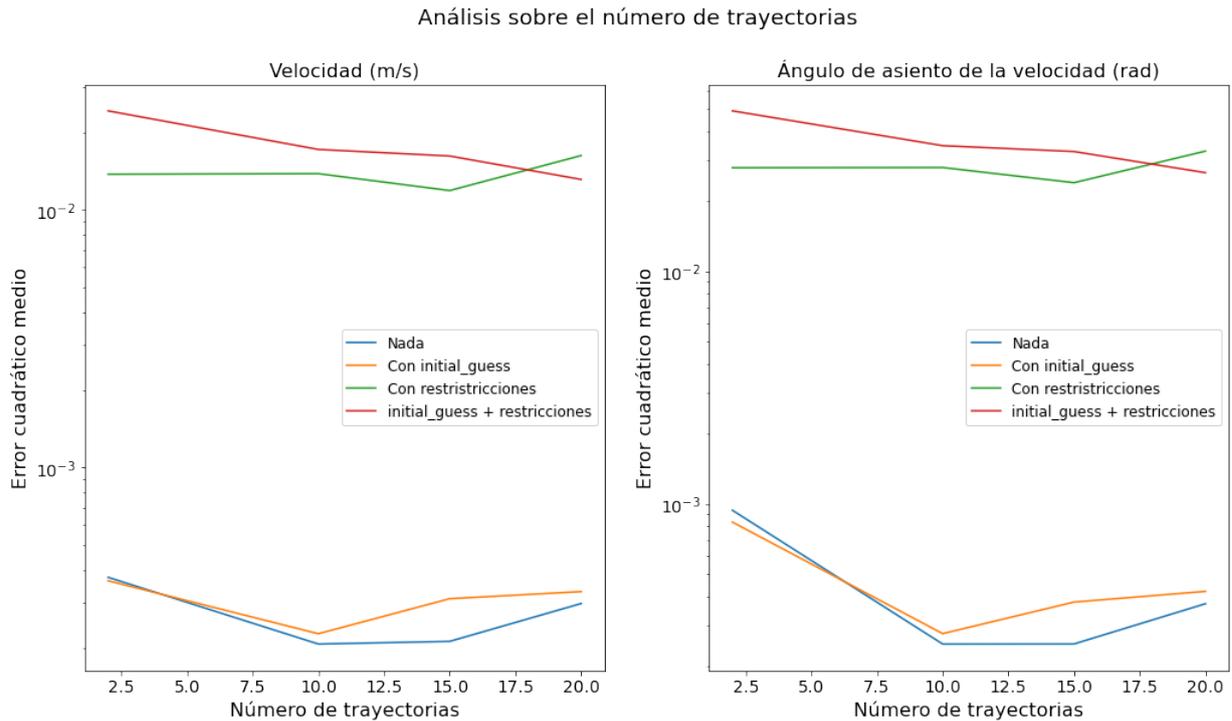


Figura 5.16: Caso D-1. Representación del error cuadrático medio de la trayectoria frente al número de trayectorias.

Por otro lado, en la figura 5.16 se representa el error cuadrático medio frente al número de trayectorias configurando algunas de las entradas opcionales, como la suposición inicial o la utilización de restricciones para ajustar el coeficiente  $A_7$  (ya que en la ecuación devuelta por el algoritmo se desviaba de su valor original). Lo que se ha buscado con este experimento es evaluar el desempeño del algoritmo al incluir información a priori de la ecuación, como puede ser el valor inicial de los coeficientes o ajustar sus valores mediante restricciones. Inicialmente, se puede pensar que utilizar restricciones mejorará el error en la trayectoria, pero no sucede así. Esto puede ser debido a que introducir una restricción en el problema de optimización puede cambiar el punto óptimo que se está

buscando. Lo que se ha observado es que introducir esta restricción modifica el valor del coeficiente  $A_7$  que el algoritmo obtiene, provocando que la trayectoria simulada y la real difieran. Asimismo, también se puede observar que incrementar el número de trayectorias no mejora el error, esto puede ser debido a que al tratarse de errores pequeños, la mejora que se puede obtener es mínima.

En la citada figura, se puede observar que los errores son inferiores a  $10^{-1}$ , por lo que el algoritmo recupera correctamente la trayectoria.

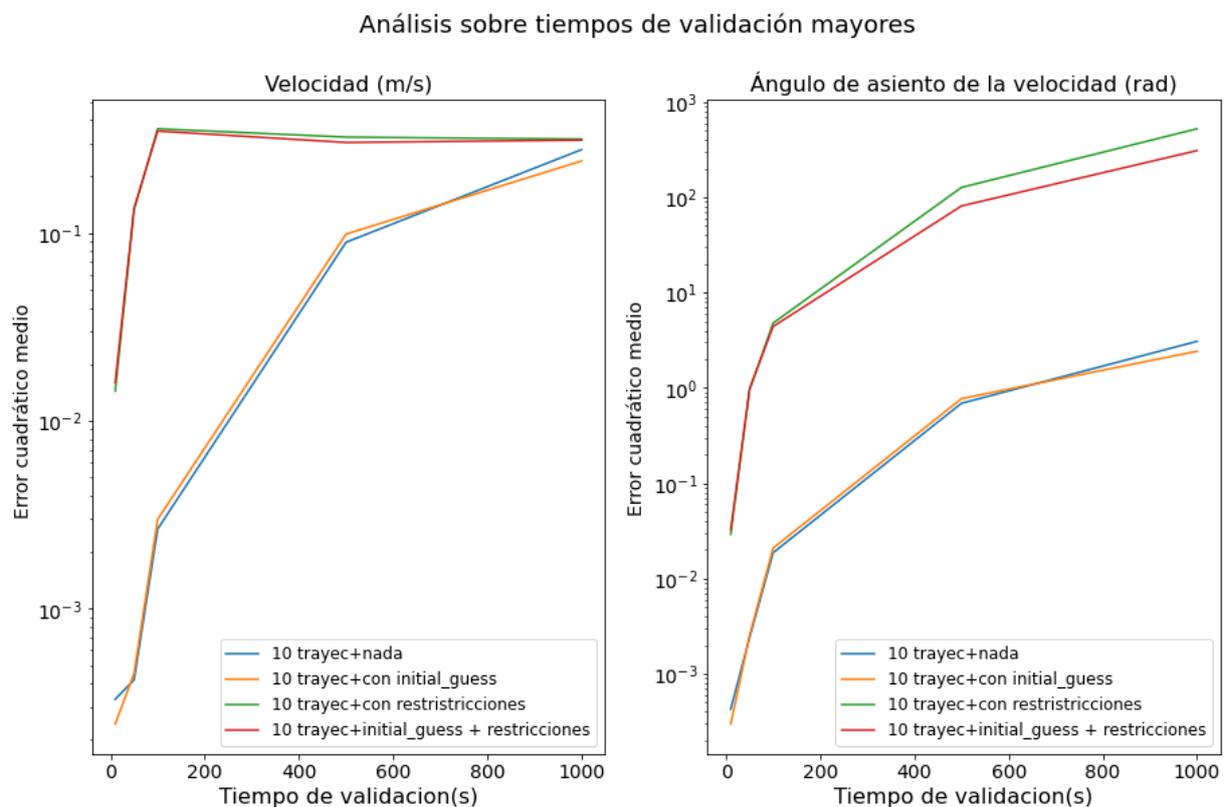


Figura 5.17: Caso D-1. Representación del error cuadrático medio de la trayectoria al variar la duración temporal de la trayectoria de validación.

En la figura 5.17 se muestra el error cuadrático medio para distintos valores máximos de tiempo usados para validar la solución aportada por el algoritmo. Tal y como se puede observar, para tiempos de validación pequeños el algoritmo presenta errores bajos, esto puede ser debido a que estos tiempos menores se asemejan al tiempo utilizado para generar los datos con los que se alimenta al algoritmo. Por otro lado, se observa que para tiempos mayores parece que se alcanza un error de saturación, esto puede ser debido a

que se alcanza la fase estacionaria de la trayectoria. Por último, de nuevo observamos que utilizar restricciones no disminuye el error y utilizar una suposición inicial no ayuda a mejorar el resultado.

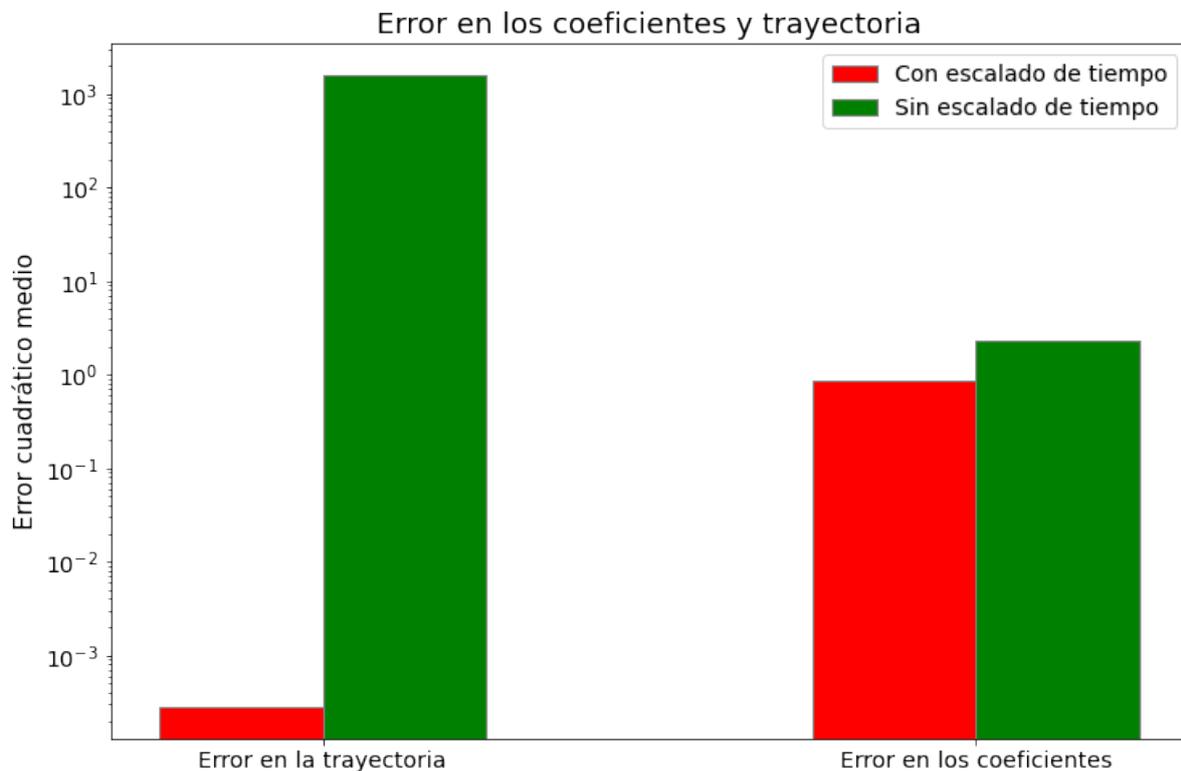


Figura 5.18: Caso D-1. Representación del error cuadrático medio de la trayectoria y de los coeficientes del sistema de ecuaciones (5.44).

En la figura 5.18 se muestra una comparativa entre alimentar el algoritmo con un paso de tiempo correcto (sin escalado de tiempo) o uno más pequeño (con escalado de tiempo) midiendo el error cuadrático medio en la trayectoria y coeficientes del sistema de ecuaciones (5.44). Tal y como se puede apreciar, la diferencia entre el error en los coeficientes es menor que en el error en la trayectoria. Aún así, y considerando que el eje  $y$  de la figura presenta escala logarítmica, la diferencia entre las dos alternativas en el caso del error en las trayectorias es grande, siendo esta diferencia de más de 6 órdenes de magnitud. Es conveniente destacar que el error en los coeficientes sin escalado de tiempo es mucho menor que el error en la trayectoria para el mismo caso, ya que cabría esperar que un error en los coeficientes no demasiado alto ocasionase un error bajo en la trayectoria. Esto puede ser debido a que se calcula el error cuadrático medio de toda la la matriz de

coeficientes  $\Xi$ , por lo que si algún coeficiente no se encuentra presente en los coeficientes obtenidos por el algoritmo, no incrementará mucho este error. Sin embargo, en el caso del error en la trayectoria se alimenta con una trayectoria de validación que el algoritmo no ha visto con anterioridad, por lo que la ecuación obtenida por el algoritmo puede que no se adapte a esta trayectoria y el error se incremente considerablemente.

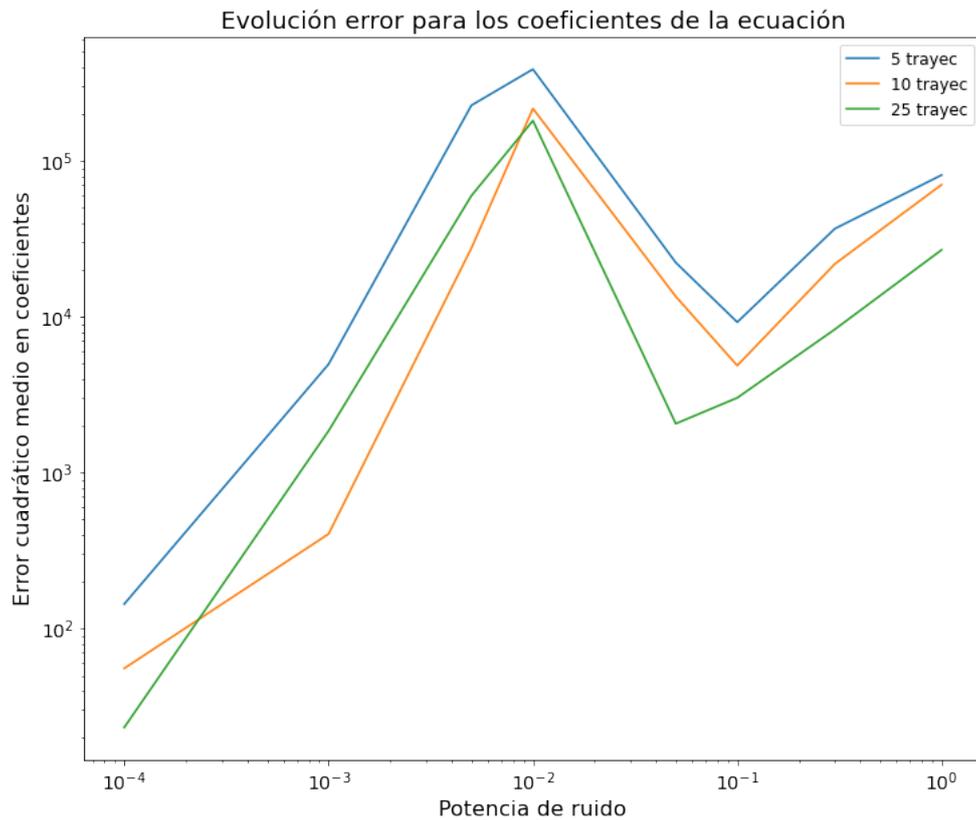


Figura 5.19: Caso D-1. Representación del error cuadrático medio los coeficientes del sistema de ecuaciones (5.44) frente a la potencia de ruido.

Por último, en la figura 5.19 se representa el error cuadrático medio de los coeficientes del sistema de ecuaciones (5.44). A diferencia de los casos anteriores se puede apreciar como en este caso el error no se mantiene bajo, sino que incluso ante potencias pequeñas del orden de  $10^{-4}$  se alcanzan errores de  $10^2$ . Este comportamiento puede ser explicado debido a que se trata de un sistema de ecuaciones complejo con combinaciones de funciones trigonométricas y polinómicas. Como vemos, ante ecuaciones con mayor número de términos y más complejas, SINDy no presenta un comportamiento robusto frente al ruido. Además, tal y como podemos observar a partir de una potencia de ruido de  $10^{-2}$

se alcanza un pico, por lo que se puede intuir que esta es una potencia de saturación a partir de la cual el comportamiento de SINDy es aleatorio.

### 5.4.2. Caso D-2: Lazo ideal en el plano vertical para empuje y ángulo de asiento de la velocidad variables

En este caso las incógnitas dependientes del tiempo son  $T$ ,  $c_L$  y  $\gamma$  que quedan descritas por 3 ecuaciones, dos ecuaciones dinámicas representadas por (2.5) y una ecuación angular representada en (5.24). Concretamente, en este caso obtenemos las ecuaciones

$$T - D - W \sin \gamma = 0 \quad (5.49)$$

$$L - W \cos \gamma = mV_0\dot{\gamma} \quad (5.50)$$

$$\dot{\gamma} = \frac{V_0}{R},$$

donde  $V_0$  denota la velocidad aerodinámica y su valor se fija a 10m/s.

Se procede despejando el  $c_L$  de la segunda ecuación y se sustituye en el término de resistencia aerodinámica inducida de la primera ecuación, obteniéndose

$$c_L = \frac{2}{\rho_{aire}SV_0^2} \left( W \cos \gamma + m \frac{V_0^2}{R} \right) \quad (5.51)$$

$$T = \frac{1}{2}S\rho_{aire}V_0^2c_{D0} + \frac{2k}{S\rho_{aire}V_0^2} \left( W^2(\cos \gamma)^2 + m^2 \frac{V_0^4}{R^2} + \frac{2WmV_0^2}{R} \cos \gamma \right) + W \sin \gamma \quad (5.52)$$

$$\dot{\gamma} = \frac{V_0}{R}. \quad (5.53)$$

Agrupando los términos constantes

$$A_1 = \frac{1}{2}S\rho_{aire}V_0^2 \quad (5.54)$$

$$A_2 = A_1c_{D0} \quad (5.55)$$

$$A_3 = \frac{k}{A_1} \frac{m^2V_0^4}{R^2} = \frac{2km^2V_0^2}{\rho_{aire}SR^2} \quad (5.56)$$

$$A_4 = \frac{k}{A_1} W^2 = \frac{2kW^2}{\rho_{aire}SV_0^2} \quad (5.57)$$

$$A_5 = \frac{k}{A_1} \frac{2WmV_0^2}{R} = \frac{4kWm}{\rho_{aire}SR} \quad (5.58)$$

$$A_6 = \frac{V_0}{R} \quad (5.59)$$

$$A_7 = A_2 + A_3 \quad (5.60)$$

y sustituyendo en el sistema anterior

$$T = A_7 + A_4(\cos \gamma)^2 + A_5 \cos \gamma + W \sin \gamma \quad (5.61)$$

$$\dot{\gamma} = A_6. \quad (5.62)$$

Sin embargo, el algoritmo obtiene las ecuaciones en derivadas, por tanto si derivamos la ecuación (5.61), sustituimos por el valor de  $\dot{\gamma}$  de la ecuación (5.53) se obtiene

$$\dot{T} = -2A_4 \frac{V_0}{R} \cos \gamma \sin \gamma - A_5 \frac{V_0}{R} \sin \gamma + W \frac{V_0}{R} \cos \gamma.$$

Para simplificar, podemos aplicar la relación trigonométrica del ángulo doble, obteniendo

$$\dot{T} = -A_4 \frac{V_0}{R} \sin 2\gamma - A_5 \frac{V_0}{R} \sin \gamma + W \frac{V_0}{R} \cos \gamma. \quad (5.63)$$

Definiendo los siguientes coeficientes

$$A_8 = A_4 \frac{V_0}{R} \quad (5.64)$$

$$A_9 = A_5 \frac{V_0}{R} \quad (5.65)$$

$$A_{10} = W \frac{V_0}{R}, \quad (5.66)$$

se sustituyen en las ecuaciones anteriores, obteniéndose

$$\dot{T} = -A_8 \sin 2\gamma - A_9 \sin \gamma + A_{10} \cos \gamma \quad (5.67)$$

$$\dot{\gamma} = A_6.$$

Inicialmente, se ha alimentado al algoritmo con trayectorias normalizadas. Sin embargo, en el sistema de ecuaciones obtenido por el algoritmo SINDy aparecían términos que no se encontraban en las ecuaciones. Para intentar solucionar este problema, se han probado dos casos. El primero consiste en alimentar con trayectorias sintéticas sin normalizar que siguen el sistema de ecuaciones integrado (5.67). La razón de no normalizar en este caso concreto es debido a que todos los coeficientes de la ecuación (5.67) tenían un valor parecido, por lo que normalizar no era tan necesario como en otros casos (casos B y C) donde la diferencia entre las magnitudes de los coeficientes de los términos hacía que SINDy no obtuviese aquellos términos más pequeños. El segundo caso consiste en

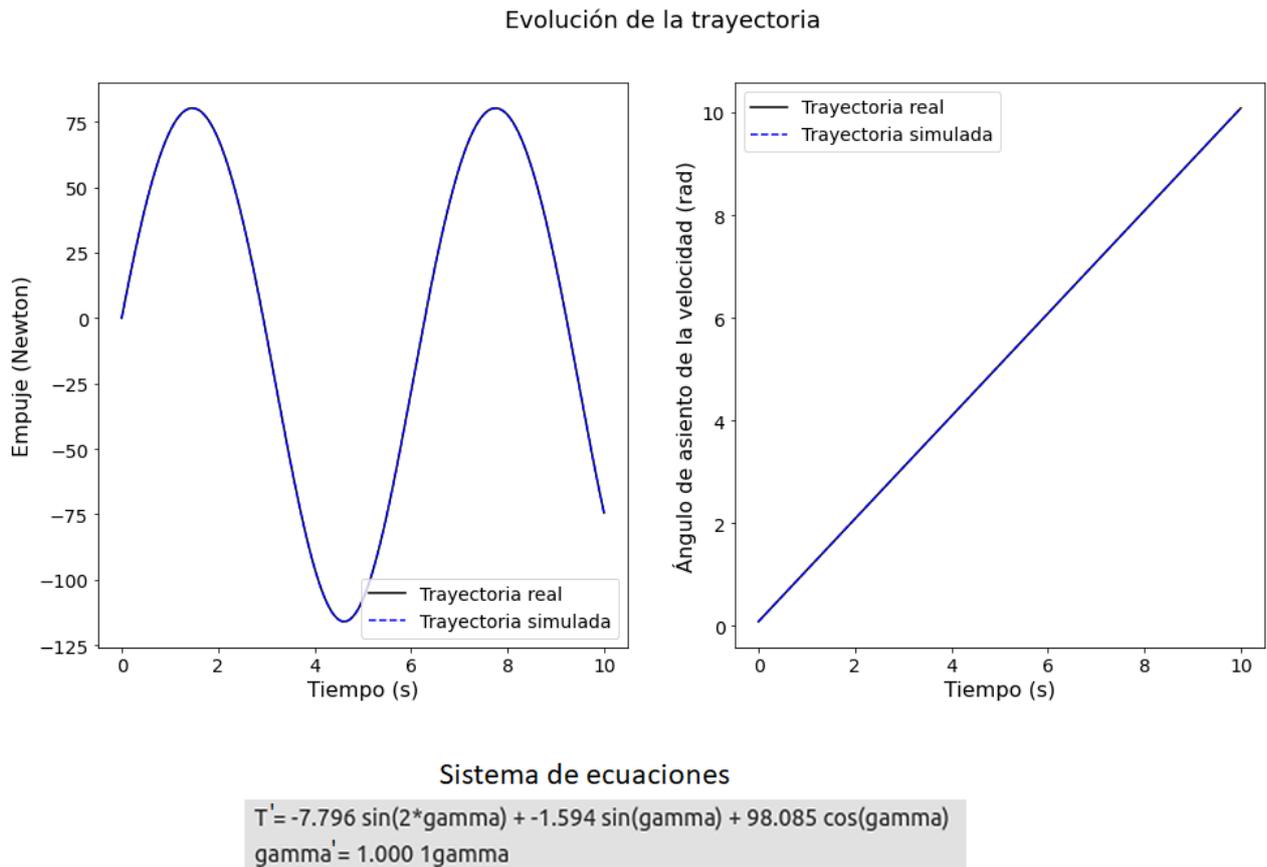


Figura 5.20: Caso D-2. Representación del empuje y ángulo de asiento de la velocidad ( $\gamma$ ) de la trayectoria frente al tiempo.

además alimentar con las derivadas de las ecuaciones indicadas, y así evitar que la librería PySINDy las calcule. En ambos casos se obtienen errores muy pequeños, por lo que en los resultados que se muestran a continuación se ha alimentado con las derivadas.

En la figura 5.20 se representa la evolución del empuje y de  $\gamma$  con el tiempo indicando si se trata la trayectoria real sintética y la trayectoria simulada por el algoritmo. Asimismo se incluyen las ecuaciones obtenidas por el algoritmo alimentando al mismo con 15 trayectorias. Tal y como se puede observar, el algoritmo de SINDy recupera las trayectorias satisfactoriamente.

En la figura 5.21 se analiza el error frente al número de trayectorias para una librería polinómica de grado 2 y una librería trigonométrica. Tal y como se puede observar el resultado es bastante prometedor, el algoritmo es capaz de recuperar la trayectoria con errores del orden de  $10^{-3}$  incluso para una librería con un número grande de funciones que

no se encuentran presentes en las ecuaciones desarrolladas. De nuevo, se puede observar que aumentar el número de trayectorias resulta beneficioso.

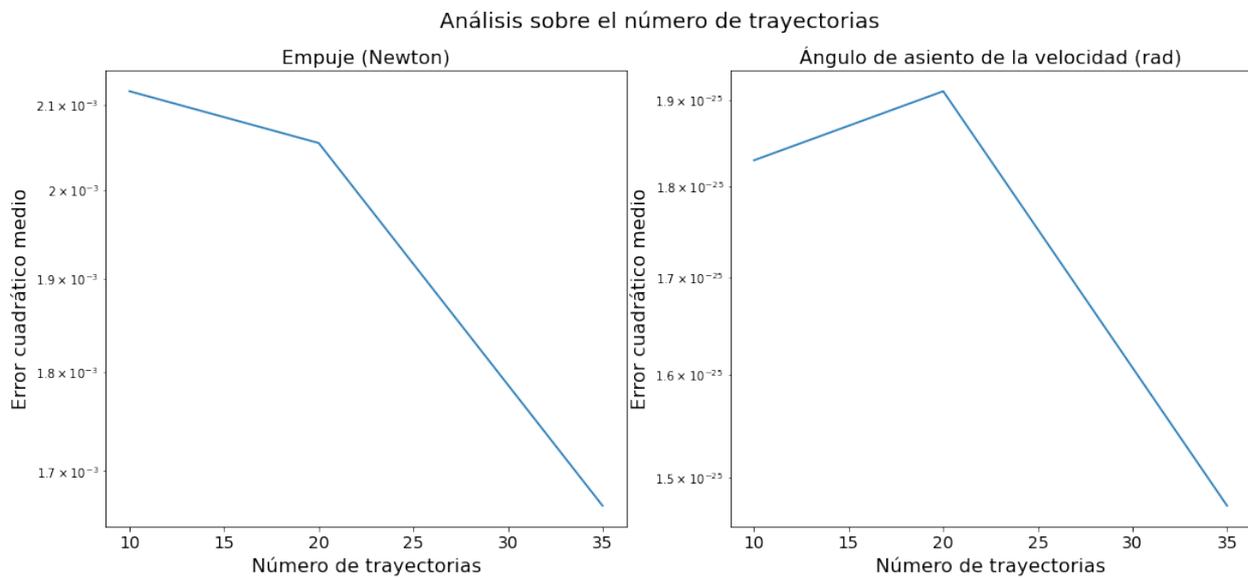


Figura 5.21: Caso D-2. Representación del error cuadrático medio de la trayectoria frente al número de trayectorias.

Por último, se ha realizado de nuevo el experimento de añadir ruido a las trayectorias alimentadas al algoritmo, con la intención de probar la robustez del mismo en un caso más complejo, y en la figura 5.22 se representa el error cuadrático medio en la trayectoria según la potencia de ruido. Esta potencia de ruido se incluye en las muestras y en las derivadas con las que se alimenta el algoritmo. No se ha incluido la representación del error para el ángulo de asiento de la velocidad por tratarse de errores muy pequeños del orden de  $10^{-25}$  muy probablemente debidos a la precisión de la máquina y aproximables a 0.

Tal y como se puede observar, a medida que crece la potencia de ruido los errores son cada vez mayores. A diferencia de casos anteriores, se puede ver como el caso del empuje frente al ruido es menos robusto que otras variables en casos anteriores, esto puede ser debido a la complejidad de las ecuaciones. Incrementar el número de trayectorias resulta ligeramente beneficioso pero no provee una mejora considerable en el error.

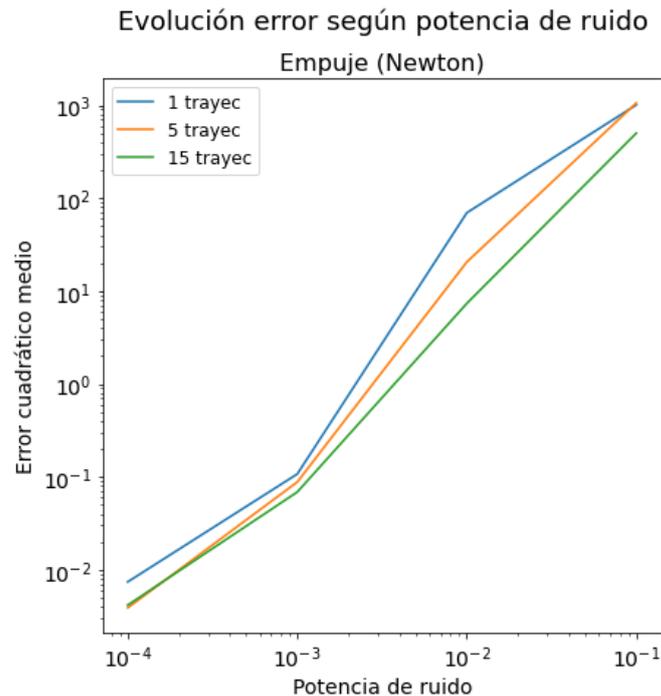


Figura 5.22: Caso D-2. Representación del error cuadrático medio de la trayectoria frente a la potencia de ruido.

## 5.5. Análisis general

Para concluir este capítulo se representa en la tabla 5.2 una comparativa entre las ecuaciones teóricas y las ecuaciones obtenidas por el algoritmo y el error cuadrático medio en los coeficientes. Esta tabla, al igual que la figura 5.23 se han generado utilizando 20 trayectorias para los datos de entrenamiento. De nuevo, para aumentar la robustez de los resultados se ha repetido todo el proceso de generación de datos, creación del modelo, entrenamiento del mismo y validación de resultados en diez ocasiones y se muestra la mediana del error de estas diez ejecuciones.

En la tabla 5.2 se puede observar que de manera general el algoritmo es capaz de recuperar correctamente las ecuaciones, mostrando un error en los coeficientes de las mismas bajo. Sin embargo, en el caso C-2, tanto para la aproximación de orden 2 como de orden 3, se puede ver que el error asociado es mayor. Esto puede ser debido a que el algoritmo es alimentado con trayectorias que siguen la ecuación (5.20) (véase la ecuación del caso C-1 en la tabla) pero se utiliza una librería polinómica de grado 2 y 3 en la que no aparece el término  $1/v^2$ . El algoritmo, en lugar de obtener los coeficientes asociados al

desarrollo en serie de Taylor entorno al punto 1, trata de aproximar la trayectoria en todos los puntos, la cual no ha sido generada utilizando esta aproximación. Esto se confirma si vemos el error asociado a los coeficientes en la tabla y el error asociado a la trayectoria, subsección 5.3.2, estos difieren considerablemente.

Por otro lado, vemos que para el caso D-1 el error asociado a los coeficientes es mayor que el caso homólogo D-2, esto es debido a que el coeficiente asociado al término  $(\cos \gamma)^2/v^2$  no es detectado correctamente por el algoritmo.

En la figura 5.23 se representa el error cuadrático medio de los coeficientes de las ecuaciones teóricas y predichas representadas en la tabla 5.2. No se ha incluido el caso A debido a que la estimación de los coeficientes era perfecta, y el error que mostraba era debido a la resolución del ordenador, obstaculizando la correcta visualización de los órdenes de magnitud del resto de casos.

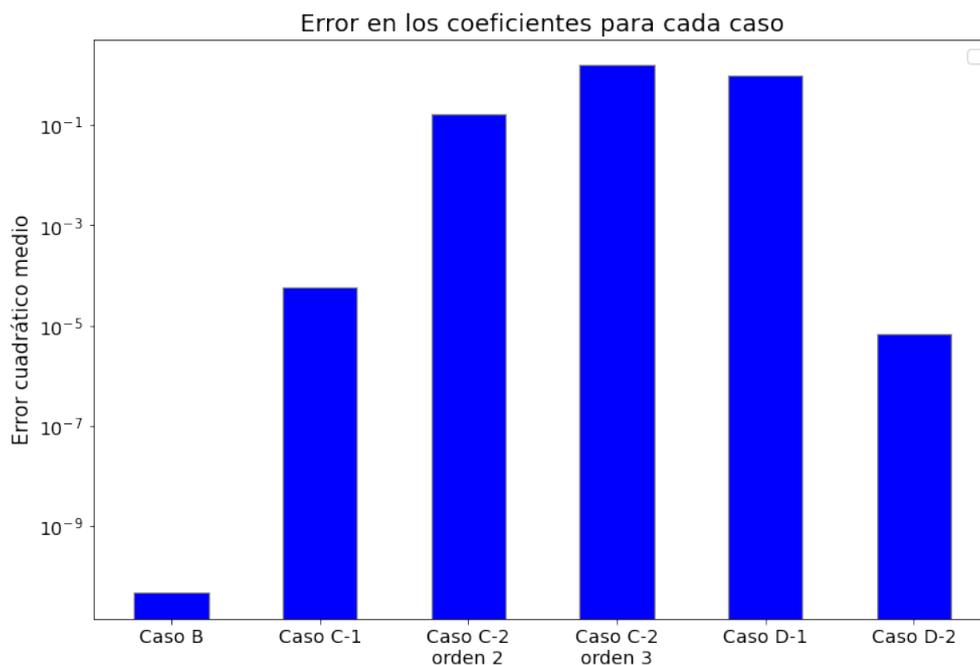


Figura 5.23: Error cuadrático medio en los coeficientes para los casos analizados.

En este proyecto se ha aplicado la técnica propuesta en SINDy, sobre regresión dispersa, para la recuperación de las ecuaciones que modelan la dinámica de los datos con los que es alimentado. En concreto, se ha aplicado sobre los atractores extraños de Lorenz, Rössler, Thomas y *Three-Scroll Unified System* y sobre las ecuaciones que modelan el comportamiento de una aeronave en vuelo para un planeador, vuelo rectilíneo en el

plano horizontal (para velocidades grandes y pequeñas y aproximar este último por una serie de Taylor) y un lazo en el plano vertical. Una vez obtenidas las ecuaciones que el algoritmo considera más adecuadas se han cambiado algunos parámetros de entrada para mejorar los resultados y comprobar cómo afecta la variación de esas entradas a la salida. Se ha podido observar que se recuperan correctamente las ecuaciones ante ruido AWGN de los casos más sencillos, incrementar el número de trayectorias ofrecidas al algoritmo que cambian su punto inicial resulta beneficioso aunque puede llegar un punto en el que incrementar este parámetro no suponga una gran mejoría en los resultados, adimensionalizar/normalizar las ecuaciones que generan las trayectorias resulta beneficioso y permite recuperar de manera más precisas dichas ecuaciones, y por último, utilizar trayectorias de validación temporalmente más largas que las utilizadas para entrenar al algoritmo ofrece muy buenos resultados.

Caso	Ecuación a obtener	Ecuación obtenida	Error
<b>Caso A</b>	$\dot{\gamma} = 0$	$\dot{\gamma} = 0$	$1,2996 * 10^{-25}$
	$\dot{x} = 1,000v \cos \gamma$	$\dot{x} = 1,000v \cos \gamma$	
	$\dot{h} = 1,000v \sin \gamma$	$\dot{h} = 1,000v \sin \gamma$	
	$\dot{v} = 0$	$\dot{v} = 0$	
<b>Caso B</b>	$\dot{v} = -0,1350v^2 + 0,120$	$\dot{v} = -0,1351v^2 + 0,1200$	$4,7699 * 10^{-11}$
<b>Caso C-1</b>	$\dot{x} = -0,1107v^2 - 0,2831\frac{1}{v^2}$	$\dot{x} = -0,1175v^2 - 0,2865\frac{1}{v^2}$	$5,8712 * 10^{-5}$
	$+0,8538$	$+0,8641$	
<b>Caso C-2</b>	Orden 2: $\dot{v} = -0,3427v^2 + 0,1139v$	Orden 2: $\dot{v} = -0,6845v^2 + 0,6819v$	Orden 2:
	$+0,2297$	$+0$	$0,1638$
	Orden 3: $\dot{v} = 0,0569v^3 - 0,5135v^2$	Orden 3: $\dot{v} = 0,7812v^3 - 2,2542v^2$	Orden 3:
	$+0,2847v + 0,1728$	$+1,6556v - 0,1807$	$1,4787$
<b>Caso D-1</b>	$\dot{v} = 12,2324 - 0,3183v^2$	$\dot{v} = 12,3344 - 0,3190v^2$	$0,9300$
	$-0,2316 \cos \gamma$	$-0,2279 \cos \gamma$	
	$-10,0000 \sin \gamma$	$-10,0034 \sin \gamma$	
	$\frac{0,7300(\cos \gamma)^2}{v^2}$	$\frac{4,7957(\cos \gamma)^2}{v^2}$	
	$\dot{\gamma} = 1,5866v$	$\dot{\gamma} = 1,5874v$	
<b>Caso D-2</b>	$\dot{T} = -7,8025 \sin 2\gamma$	$\dot{T} = -7,7972 \sin 2\gamma$	$6,7587 * 10^{-6}$
	$-1,5907 \sin \gamma$	$-1,5941 \sin \gamma$	
	$+98,1000 \cos \gamma$	$+98,0919 \cos \gamma$	
	$\dot{\gamma} = 1$	$\dot{\gamma} = 1$	

Tabla 5.2: Tabla comparativa entre las ecuaciones obtenidas por SINDy y las teóricas para las trayectorias de aeronaves.



# Capítulo 6

## Conclusiones

A lo largo de este proyecto se ha tratado de analizar si es posible utilizar la técnica SINDy, basada en regresión dispersa, en el campo de la ingeniería aeroespacial para la predicción de las ecuaciones que modelan el comportamiento de una aeronave en vuelo a partir de datos de sus trayectorias. Para ello, se han seleccionado diversos casos de ejemplo de las trayectorias de una aeronave, como pueden ser un planeador, un vuelo rectilíneo horizontal (para velocidades grandes y pequeñas) y para el caso de un lazo en el plano vertical. Posteriormente, se han generado trayectorias sintéticas para dichos casos y con ellas se ha alimentado al algoritmo de SINDy. Con las ecuaciones obtenidas por el mismo, se ha evaluado el error tanto en los coeficientes de las ecuaciones como en la trayectoria simulada por dichas ecuaciones, para evaluar por un lado si SINDy era capaz de obtener correctamente las ecuaciones, y por otro si las trayectorias obtenidas por las ecuaciones de SINDy se adecuaban a la trayectoria original de la aeronave. Para concluir se han variado algunos parámetros opcionales que ofrece este algoritmo con el fin de evaluar su impacto en los resultados obtenidos.

El resultado obtenido es que 1. SINDy recupera correctamente las ecuaciones con una librería de términos genérica, con errores en los coeficientes bastante pequeños y 2. las trayectorias obtenidas por las ecuaciones de SINDy son muy similares a las que debe obtener. Por lo tanto, se puede concluir que es posible utilizar SINDy en este ámbito.

Esto puede suponer diversas ventajas, por ejemplo gracias al uso de SINDy en este campo sería posible obtener modelos más precisos simplemente a partir de los datos medidos por los sensores ya instalados en las aeronaves. También nos permitiría obtener

modelos más simples que, aunque no contengan todos los elementos de la ecuación teórica, también describan de manera precisa el movimiento de la aeronave. Estos modelos podrían servir, por ejemplo, para predecir la trayectoria de una aeronave de manera precisa ejecutando ensayos de vuelo sencillos en diversas condiciones, u obtener los coeficientes de un perfil aerodinámico sin necesidad de utilizar un túnel de viento

Sin embargo, hay que tener en cuenta que puede ser necesario llevar a cabo una serie de transformaciones a los datos. Tal y como se ha visto en el capítulo anterior normalizar y, sobretudo, adimensionalizar ecuaciones complejas facilita que el algoritmo obtenga las ecuaciones correctamente, siendo totalmente necesario en alguno de los casos. Por otro lado, es necesario prestar atención a las diferencias entre los órdenes de magnitud de los distintos coeficientes de las ecuaciones. Tal y como se ha visto, una diferencia substancial en ellos puede ocasionar que el algoritmo no detecte aquellos con un valor más pequeño. Para evitar esto, es necesario normalizar/adimensionalizar la ecuación o multiplicar dichos coeficientes por un valor.

También se ha llegado a diversas conclusiones en cuanto a los parámetros opcionales que ofrece este algoritmo, concretamente: se ha visto en varios experimentos que utilizar múltiples trayectorias cuya condición inicial sea distinta resulta muy beneficioso; aplicar restricciones para forzar que uno de los términos aparezca en la salida del algoritmo puede empeorar el error al cambiar el punto óptimo; aportar las derivadas de las variables cuando se dispone de ellas, para evitar que el algoritmo tenga que calcularlas numéricamente, ayuda a mejorar el error; usar una suposición inicial, la cual puede mejorar el error si se utiliza un menor número de trayectorias, pero en ningún caso lo empeora; y cambiar el paso de tiempo permite al algoritmo detectar aquellos coeficientes cuyo orden de magnitud en la ecuación es pequeño.

Es especialmente interesante alimentar al algoritmo utilizando datos de trayectorias breves temporalmente, ya que este es capaz de obtener trayectorias mucho más largas en el tiempo con bastante precisión. Es decir, con una trayectoria de unos pocos segundos, el algoritmo de SINDy es capaz de obtener ecuaciones que modelen trayectorias de la duración que queramos, no limitada a la duración e las trayectorias utilizadas para alimentar el algoritmo.

También es necesario destacar la robustez frente al ruido que ofrece este algoritmo en

determinados tipos de ecuaciones. En la mayoría de casos el error ante ruidos elevados es pequeño, por lo que usar esta técnica en un entorno real (donde las trayectorias son ruidosas) puede ser muy interesante. Sin embargo, ante ecuaciones con mayor número de términos y matemáticamente más complejas, el error en los coeficientes de la ecuación es elevado. Para mejorar el resultado sería interesante aplicar el algoritmo SINDy para trayectorias ruidosas, propuesto en [9]. No obstante, la utilización de SINDy sigue siendo viable a la hora de estimar los coeficientes en un entorno real, pues las trayectorias con términos muy complejos no son habituales.

Por último, cabe destacar que el algoritmo SINDy es capaz de obtener las ecuaciones de atractores extraños con una elevada precisión, por lo que es posible utilizar esta técnica para recuperar modelos dinámicos a partir de trayectorias caóticas.

La estructura de lo que resta de capítulo es la siguiente, primeramente se analizará la consecución de los objetivos planteados en la sección 1.2, se explicarán los conocimientos del grado aplicados al proyecto y los conocimientos adquiridos durante la realización del mismo, y por último, se plantean unas líneas de trabajo futuras.

## 6.1. Consecución de objetivos

En la sección anterior se detallan las conclusiones alcanzadas en la realización de este proyecto. Sin embargo, en este apartado se detallará la consecución de los objetivos planteados en la sección 1.2.

En los capítulos 3 y 5 se ha analizado el uso de SINDy para la obtención de modelos dinámicos (OG1), tanto para atractores extraños, sección 3.1 (OE1.1), como para las ecuaciones dinámicas de una aeronave.

En el capítulo 5 (OG2), siguiendo la arquitectura descrita en el capítulo 4, se han implementado las ecuaciones que modelan la dinámica de una aeronave ante determinados casos (OE2.1), se han adimensionalizado/normalizado (OE2.2), se ha utilizado SINDy para recuperar las ecuaciones, siendo en algunos casos necesario aplicar otras transformaciones para mejorar el resultado, y se ha validado el modelo obtenido midiendo el error en la trayectoria real sintética y en la simulada (OE2.3).

Por último, en ese mismo capítulo se han modificado algunos parámetros que el algorit-

mo permite configurar, como es aportar las derivadas, cambiar el número de trayectorias presentes en los datos o modificar el paso de tiempo (OE3.1) con el fin de mejorar los resultados obtenidos.

## 6.2. Aplicación de lo aprendido

Algunas materias del grado que me han ayudado en la realización de este proyecto son:

1. **Mecánica de vuelo:** aplicación de las ecuaciones generales de MV a determinados casos específicos para la generación de sus datos.
2. **Sistemas de mando y control:** en esta asignatura he aprendido a modelar sistemas dinámicos mediante ecuaciones diferenciales, lo que me ha ayudado a entender la importancia de este concepto.
3. **Optimización de espacio aéreo:** gracias a esta asignatura he aprendido sobre optimización de sistemas, lo que me ha ayudado a entender más fácilmente el funcionamiento de SINDy.
4. **Ecuaciones diferenciales y cálculo numérico:** esta asignatura me ha ayudado a entender ecuaciones diferenciales y a saber cómo despejar variables de este tipo de ecuaciones.
5. **Programación:** gracias a esta asignatura he aprendido el lenguaje de programación Python y en general adquirir conocimientos sobre pensamiento computacional. Todo ello necesario para la implementación de los casos de este proyecto.

## 6.3. Lecciones aprendidas

Algunas lecciones que he aprendido durante la elaboración de este proyecto son las siguientes:

1. Mejor comprensión de las ecuaciones que modelan el comportamiento de aeronaves, y sobretodo su implementación en un lenguaje de programación para obtener las trayectorias sintéticas.
2. Aprendizaje del funcionamiento del paquete PySINDy y de la técnica subyacente, SINDy.
3. Búsqueda de documentación y artículos científicos en inglés.
4. Redacción de una memoria técnica.

## 6.4. Trabajos futuros

Algunas líneas de trabajo futuras que plantea este proyecto son:

- **Utilizar SINDy para la obtención de las ecuaciones que modelan otros vehículos aéreos**, como cohetes o drones. Como ya se ha visto, este algoritmo puede ser utilizado para obtener las ecuaciones que rigen en movimiento de una aeronave, las cuales son ecuaciones dinámicas. Por lo tanto, podríamos intentar predecir sistemas más complejos que se rijan por el mismo tipo de ecuaciones, por ejemplo, las de otros vehículos aéreos que sean más difíciles de modelar, lo cual resulta una línea de investigación interesante.

En el caso de un cohete las ecuaciones que modelan su movimiento en dos dimensiones serían

$$\begin{aligned}
 \dot{x} &= V \cos(\theta - \alpha) \\
 \dot{y} &= V \sin(\theta - \alpha) \\
 \dot{V} &= A \cos(\alpha) - BV^2 - C \sin(\theta - \alpha) \\
 \dot{\theta} &= u \\
 \dot{\alpha} &= Du - EV\alpha - F \frac{1}{V} \sin(\alpha) + G \frac{1}{V} \cos(\theta - \alpha) \\
 \ddot{\theta} &= HV^2\alpha + IVu + JV\dot{\alpha},
 \end{aligned}$$

donde  $x, y, \theta, \alpha, V$  y  $\dot{\theta}$  son las incógnitas dependientes del tiempo. En este caso es especialmente interesante alimentar al algoritmo con trayectorias que incluyan tanto

velocidades pequeñas como grandes para activar todos los términos de la ecuación. Asimismo, sería recomendable eliminar las dos primeras ecuaciones (las que modelan  $\dot{x}$  e  $\dot{y}$ ), ya que no se encuentran presentes en el resto del sistema, facilitando que SINDy recupere correctamente todos los términos.

- **Evaluar el funcionamiento de SINDy ante coeficientes variables.** En este proyecto se han utilizado coeficientes que no dependen del tiempo, como por ejemplo el  $c_{D0}$ . Sin embargo, utilizar coeficientes variables con el tiempo, por ejemplo porque dependan de la densidad atmosférica, la cual a su vez puede depender de otros parámetros como la altura, requiere introducir unas ecuaciones que describan la variación de esos coeficientes, dificultando al algoritmo la identificación de las ecuaciones. Por ejemplo, a lo largo de este proyecto se ha definido en el caso D-2 el sistema (5.67) que debe obtener el algoritmo, junto a los coeficientes constantes dependientes de la densidad  $A_8$ ,  $A_9$  y  $A_{10}$  determinados por las ecuaciones (5.64), (5.65) y (5.66), respectivamente. Si se considerara, por ejemplo, la densidad variable, sería necesario incluir la expresión que determina la densidad atmosférica según la altura. Considerando el modelo de atmósfera estándar internacional [23]

$$T_e(h) = T_{e_0} + \alpha(h - h_0)$$

$$\rho(h) = \rho_0 \left( \frac{T_e(h)}{T_{e_0}} \right)^{-1 - \frac{g}{\alpha R}},$$

donde  $g$  es la gravedad terrestre y  $\rho_0$ ,  $h_0$ ,  $T_{e_0}$  y  $\alpha$  son valores densidad, altura, temperatura y gradiente térmico fijados según la zona de la atmósfera.

Como vemos aparece otra incógnita que sería la altura, por lo que es necesario añadir al sistema la tercera ecuación dinámica (2.6), donde  $\dot{z}_e = -\dot{h}$ .

En este caso hemos pasado de tener un sistema de dos ecuaciones con dos incógnitas a un sistema con cinco ecuaciones y cinco incógnitas:  $T$ ,  $\gamma$ ,  $\rho$ ,  $h$  y la temperatura  $T_e$ , lo cual introduce mayor complejidad al algoritmo a la hora de recuperar estas ecuaciones a partir de sus trayectorias.

- **Evaluar el comportamiento de SINDy ante trayectorias de aeronaves más complejas.** En este caso se pretende evaluar el desempeño del algoritmo ante trayectorias de aeronaves más realistas. Éstas pueden incluir términos más complejos

en sus ecuaciones, como pueden ser términos exponenciales o trigonométricos que dependan de varias variables. Por ejemplo, puede ser interesante tratar de obtener las ecuaciones que modelan el despegue y aterrizaje de una aeronave en la fase de rotación y rodadura con el tren principal en el suelo. En estos casos es necesario considerar la ecuación de momentos, la aceleración angular  $\frac{d^2\theta}{dt^2}$ , el momento de inercia de la aeronave, los coeficientes de rozamiento de las ruedas con la pista y las fuerzas normales de dichas ruedas.

Otro ejemplo más común consistiría en incluir la presencia de viento en las ecuaciones. En concreto, es necesario tener en cuenta la ecuación del momento cinético particularizada para un sistema no inercial (en ejes cuerpo o en ejes viento)

$$\vec{F} = m\left(\frac{\partial \vec{V}_g}{\partial t} + \vec{\omega} \times \vec{V}_g\right),$$

donde el término  $\vec{\omega} \times \vec{V}_g$ , modela la rotación del sistema de referencia respecto de un sistema inercial (horizonte local),  $\vec{V}_g$  hace referencia a la velocidad absoluta del centro de masas del avión respecto a un sistema de ejes inerciales,  $m$  es la masa del avión,  $\vec{F}$  la resultante de fuerzas exteriores y  $\omega$  la velocidad angular absoluta del avión.

En este caso, la componente del viento se encuentra en el término de la velocidad absoluta  $\vec{V}_g$  de manera que

$$\vec{V}_g = \vec{V} + \vec{V}_w,$$

donde  $\vec{V}$  es la velocidad aerodinámica del avión y  $\vec{V}_w$  es el vector que modela el viento.

A continuación, para alimentar al algoritmo simplemente sería necesario utilizar datos de trayectorias con viento y medir este vector para incluirlo en la matriz  $\mathbf{X}$  y analizar cómo de precisa es la aproximación que proporciona SINDy de la ecuación que modela el viento.

- **Utilización de datos de trayectorias reales** de una aeronave en lugar de sintéticos para la obtención de las ecuaciones que rigen su movimiento. De esta manera se

podrá validar el comportamiento del algoritmo ante datos reales que, además, pueden requerir un proceso previo de limpieza. Asimismo, y dado que los datos reales son de naturaleza ruidosa, puede ser interesante aplicar la técnica propuesta en [9] para utilizar una versión más robusta del algoritmo SINDy ante datos con ruido. Gracias a este trabajo se podrían establecer las bases para utilizar SINDy en un entorno real.

En ese sentido se podrían utilizar los datos de trayectorias proporcionados en [12]. En concreto, recoge datos de trayectorias de aeronaves entre el 19 de septiembre del 2020 y el 23 de abril del 2021 en la zona del aeropuerto de Pittsburgh-Butler, Pensilvania, Estados Unidos. Asimismo, también se proporciona información meteorológica que incluye los datos sobre el viento (con lo que probablemente sería necesario utilizar la metodología comentada en el punto anterior).

# Bibliografía

## Libros

- [1] J. H. Ahlberg y col. *The Theory of Splines and Their Applications: Mathematics in Science and Engineering: A Series of Monographs and Textbooks, Vol. 38.* v. 38. Elsevier Science, 2016. ISBN: 9781483222950. URL: <https://books.google.es/books?id=3bZ1DAAAQBAJ>.
- [2] S. A. Billings. *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains.* USA: John Wiley y Sons, 2013. ISBN: 9781119943594.
- [3] M. W. Hirsch, S. Smale y R. L. Devaney. *Differential equations, dynamical systems, and an introduction to chaos.* Estados Unidos: Academic Press (edition 2), 2003. ISBN: 9780123497031.
- [4] T. E. W. Khattak y P. Buhler. *Big Data Fundamentals: Concepts, Drivers and Techniques.* USA: Prentice Hall, 2016. ISBN: 9780134291079.
- [5] M. Á. Gómez Tierno, M. Pérez Cortés y C. Puentes Márquez. *Mecánica de vuelo.* España: Garceta grupo editorial, 2012. ISBN: 9788415452010.

## Artículos científicos

- [6] J. Bongard y H. Lipson. «Automated reverse engineering of nonlinear dynamical systems». En: *Proceedings of the National Academy of Sciences* 104.24 (2007), págs. 9943-9948. ISSN: 0027-8424. DOI: [10.1073/pnas.0609476104](https://doi.org/10.1073/pnas.0609476104). eprint: <https://doi.org/10.1073/pnas.0609476104>.

- [//www.pnas.org/content/104/24/9943.full.pdf](https://www.pnas.org/content/104/24/9943.full.pdf). URL: <https://www.pnas.org/content/104/24/9943>.
- [7] S. L. Brunton, J. L. Proctor y J. N. Kutz. «Discovering governing equations from data by sparse identification of nonlinear dynamical systems». En: *Proceedings of the National Academy of Sciences* 113.15 (2016), págs. 3932-3937. ISSN: 0027-8424. DOI: [10.1073/pnas.1517384113](https://doi.org/10.1073/pnas.1517384113). URL: <https://www.pnas.org/content/113/15/3932.full.pdf>.
- [8] K. P. Champion y col. «A Unified Sparse Optimization Framework to Learn Parsimonious Physics-Informed Models From Data». En: *IEEE Access* 8 (2020), págs. 169259-169271. DOI: [10.1109/ACCESS.2020.3023625](https://doi.org/10.1109/ACCESS.2020.3023625).
- [9] K. Kaheman, J. N. Kutz y S. L. Brunton. «SINDy-PI: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics». En: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 476.2242 (oct. de 2020), pág. 20200279. DOI: [10.1098/rspa.2020.0279](https://doi.org/10.1098/rspa.2020.0279).
- [10] I. G. Kevrekidis y col. «Equation-free, coarse-grained multiscale computation: enabling microscopic simulators to perform system-level analysis». En: *Commun. Math. Sci* 1.4 (2003), págs. 715-762.
- [11] L. Pan y col. «A New Three-Scroll Unified Chaotic System Coined». En: *International Journal of Nonlinear Science* 10 (2010), págs. 462-474. URL: <http://www.internonlinearscience.org/upload/papers/20110618025420887.pdf>.
- [12] J. Patrikar y col. «TrajAir: A General Aviation Trajectory Dataset». En: (jun. de 2021). DOI: [10.1184/R1/14866251.v1](https://doi.org/10.1184/R1/14866251.v1). URL: [https://kilthub.cmu.edu/articles/dataset/TrajAir\\_A\\_General\\_Aviation\\_Trajectory\\_Dataset/14866251](https://kilthub.cmu.edu/articles/dataset/TrajAir_A_General_Aviation_Trajectory_Dataset/14866251).
- [13] M. Raissi, P. Perdikaris y G.E. Karniadakis. «Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations». En: *Journal of Computational Physics* 378 (2019), págs. 686-707. ISSN: 0021-9991. DOI: <https://doi.org/10.1016/j.jcp.2018.10.045>. URL: <https://www.sciencedirect.com/science/article/pii/S0021999118307125>.

- [14] C. Remani. «Numerical Methods for Solving Systems of Nonlinear Equations». En: 2013. URL: <https://www.lakeheadu.ca/sites/default/files/uploads/77/docs/RemaniFinal.pdf>.
- [15] A. Savitzky y M. J. E. Golay. «Smoothing and Differentiation of Data by Simplified Least Squares Procedures.» En: *Analytical Chemistry* 36.8 (1964), págs. 1627-1639. DOI: [10.1021/ac60214a047](https://doi.org/10.1021/ac60214a047). URL: <https://doi.org/10.1021/ac60214a047>.
- [16] M. Schmidt y H. Lipson. «Distilling free-form natural laws from experimental data». En: *science* 324.5923 (2009), págs. 81-85. DOI: [10.1126/science.1165893](https://doi.org/10.1126/science.1165893). URL: <https://www.science.org/doi/abs/10.1126/science.1165893>.
- [17] M. Schmidt y col. «Automated refinement and inference of analytical models for metabolic networks». En: *Physical biology* 8.5 (2011), pág. 055011. DOI: [10.1088/1478-3975/8/5/055011](https://doi.org/10.1088/1478-3975/8/5/055011).
- [18] B. de Silva y col. «PySINDy: A Python package for the sparse identification of nonlinear dynamical systems from data». En: *Journal of Open Source Software* 5.49 (2020), pág. 2104. DOI: [10.21105/joss.02104](https://doi.org/10.21105/joss.02104). URL: <https://doi.org/10.21105/joss.02104>.
- [19] P. Zheng y col. «A Unified Framework for Sparse Relaxed Regularized Regression: SR3». En: *IEEE Access* 7 (2019), págs. 1404-1423. DOI: [10.1109/ACCESS.2018.2886528](https://doi.org/10.1109/ACCESS.2018.2886528).

## Otras fuentes

- [20] J. L. Arroyo Barrigüete. *Introducción a los sistemas dinámicos*. Fecha de consulta 25 de enero de 2022. URL: [https://frdelpino.es/investigacion/wp-content/uploads/2020/02/EC013-00-04-Capitulo\\_04.pdf](https://frdelpino.es/investigacion/wp-content/uploads/2020/02/EC013-00-04-Capitulo_04.pdf).
- [21] Britannica. *Chaos theory*. Fecha de consulta 6 de marzo de 2022. URL: <https://www.britannica.com/science/chaos-theory>.
- [22] J. C. Ponce Campuzano. *Strange Attractors*. Fecha de consulta 25 de enero de 2022. URL: <https://www.dynamicmath.xyz/strange-attractors/>.

- [23] M. Cavcar. *The International Standard Atmosphere (ISA)*. Universidad de Anadolu. Fecha de consulta 8 de abril de 2022. URL: <http://fisicaatmo.at.fcen.uba.ar/practicas/ISAweb.pdf>.
- [24] L. Camacho Delgado. *Breve introducción a los atractores caóticos*. Universidad de Córdoba. Fecha de consulta 25 de enero de 2022. URL: <https://www.uco.es/organiza/departamentos/quimica-fisica/quimica-fisica/MC/QC2.htm>.
- [25] R. R. Rivera Durón. *El mundo torcido de los sistemas dinámicos caóticos*. Fecha de consulta 25 de enero de 2022. URL: <http://www.uaslp.mx/Comunicacion-Social/Documents/Divulgacion/Revista/Quince/225/225-04.pdf>.
- [26] Escuela superior de matemáticas de Barcelona. *Dynamical systems and differential equations*. Fecha de consulta 23 de enero de 2022. URL: <https://bgsmath.cat/dynamical-systems-2/>.
- [27] G. Gordon y R. Tibshirani. *Optimization: Lecture 2, August 30*. Curso de optimización en otoño de la Universidad Carnegie Mellon. Fecha de consulta 28 de enero de 2022. URL: [https://www.cs.cmu.edu/~ggordon/10725-F12/scribes/10725\\_Lecture2.pdf](https://www.cs.cmu.edu/~ggordon/10725-F12/scribes/10725_Lecture2.pdf).
- [28] S. He. *Introduction to dynamical system modelling*. Fecha de consulta 23 de enero de 2022. URL: [https://www.cs.bham.ac.uk/~szh/teaching/matlabmodeling/Lecture2\\_body.pdf](https://www.cs.bham.ac.uk/~szh/teaching/matlabmodeling/Lecture2_body.pdf).
- [29] J. D. Hunter. *Lorenz Attractor*. Fecha de consulta 7 de enero de 2022. URL: [https://matplotlib.org/3.1.0/gallery/mplot3d/lorenz\\_attractor.html](https://matplotlib.org/3.1.0/gallery/mplot3d/lorenz_attractor.html).
- [30] Instituto Nacional de Técnica Aeroespacial. *Área de Mecánica de Vuelo: Funciones*. Fecha de consulta 26 de enero de 2022. URL: <https://www.inta.es/MecanicaDeVuelo/es/inicio/funciones/#>.
- [31] E. C. Leskow. *Sistema de referencia*. Fecha de consulta 26 de enero de 2022. URL: <https://concepto.de/sistema-de-referencia/>.
- [32] Ministerio de Educación. *Definición de sistema de referencia*. Fecha de consulta 26 de enero de 2022. URL: [http://newton.cnice.mec.es/materiales\\_didacticos/sist\\_ref/sist\\_ref.html](http://newton.cnice.mec.es/materiales_didacticos/sist_ref/sist_ref.html).

- [33] C. R. Nave. *Péndulo Simple*. Fecha de consulta 23 de enero de 2022. URL: <http://hyperphysics.phy-astr.gsu.edu/hbasees/pend.html>.
- [34] C. R. Nave. *Series de Taylor*. Fecha de consulta 18 de febrero de 2022. URL: <http://hyperphysics.phy-astr.gsu.edu/hbasees/tayser.html>.
- [35] D. Q. Nykamp. *The idea of a dynamical system*. Fecha de consulta 23 de enero de 2022. URL: [https://mathinsight.org/dynamical\\_system\\_idea](https://mathinsight.org/dynamical_system_idea).
- [36] B. de Silva y col. *PySINDy*. Fecha de consulta 18 de febrero de 2022. URL: <https://pysindy.readthedocs.io/en/latest/index.html>.
- [37] R. Tibshirani y L. Wasserman. *A Closer Look at Sparse Regression*. Curso de métodos estadísticos para el aprendizaje automático en la Universidad Carnegie Mellon. Fecha de consulta 5 de marzo de 2022. URL: <https://www.stat.cmu.edu/~larry/=sml/sparsity.pdf>.
- [38] Universidad de Brown. *Introduction to Learning Dynamical Systems*. Fecha de consulta 23 de enero de 2022. URL: <https://cs.brown.edu/research/ai/dynamics/tutorial/Documents/DynamicalSystems.html>.
- [39] Universidad de Valencia. *Introducción a los modelos dinámicos*. Fecha de consulta 23 de enero de 2022. URL: <https://www.uv.es/olmos/Ecuaciones%20diferenciales.pdf>.
- [40] Universidad de Waterloo. *Dynamical Systems*. Fecha de consulta 23 de enero de 2022. URL: <https://uwaterloo.ca/applied-mathematics/future-undergraduates/what-you-can-learn-applied-mathematics/dynamical-systems>.
- [41] Universidad Politécnica de Madrid. *Breve introducción a los atractores caóticos*. Fecha de consulta 25 de enero de 2022. URL: [https://youtu.be/J6q\\_2\\_DdYfI](https://youtu.be/J6q_2_DdYfI).
- [42] Universidad Politécnica de Madrid. *MOOC Caos 4.3: El atractor de Rössler*. Fecha de consulta 25 de enero de 2022. URL: <https://youtu.be/ubgvwjQJdJg>.
- [43] upGrad. *Benefits and Advantages of Big Data and Analytics in Business*. Fecha de consulta 19 de diciembre de 2021. URL: <https://www.upgrad.com/blog/benefits-and-advantages-of-big-data-analytics-in-business/>.

- [44] H. Urrutxua y L. Cadarso. *Mathematical Modelling of Physical Systems*. Diapositivas de la asignatura de sistemas de mando y control. 2020.
- [45] Wikipedia. *Atractor de Lorenz*. Fecha de consulta 25 de enero de 2022. URL: [https://es.wikipedia.org/wiki/Atractor\\_de\\_Lorenz](https://es.wikipedia.org/wiki/Atractor_de_Lorenz).
- [46] Wikipedia. *Atractores*. Fecha de consulta 25 de enero de 2022. URL: <https://es.wikipedia.org/wiki/Atractor>.
- [47] Wikipedia. *Thomas' cyclically symmetric attractor*. Fecha de consulta 25 de enero de 2022. URL: [https://en.wikipedia.org/wiki/Thomas'\\_cyclically\\_symmetric\\_attractor](https://en.wikipedia.org/wiki/Thomas'_cyclically_symmetric_attractor).
- [48] Xataka ciencia. *Efecto mariposa y atractores (III)*. Fecha de consulta 25 de enero de 2022. URL: <https://www.xatakaciencia.com/matematicas/efecto-mariposa-y-atractores-iii>.
- [49] Xataka ciencia. *Efecto mariposa y atractores (y V)*. Fecha de consulta 25 de enero de 2022. URL: <https://www.xatakaciencia.com/matematicas/efecto-mariposa-y-atractores-y-v>.
- [50] H. Zenil y N. A. Kiani. *5.1 What is a Dynamical System?* Fecha de consulta 23 de enero de 2022. URL: <https://www.youtube.com/watch?v=6bGVtoqtWWw&t=21s>.