



Departamento de
Ingeniería Mecánica
Universidad de Concepción

Modelado de la dinámica de satélites como cuerpos rígidos mediante redes neuronales artificiales para aplicaciones de control.

Rodolfo Ignacio Fernández Basly

Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción
para optar al título profesional de Ingeniero Civil Aeroespacial

Profesor guía:
Dr.-Ing. Félix Leaman W.
PhD. Bernardo Hernández V.

Julio 2026
Concepción, Chile

©2026 Rodolfo Ignacio Fernández Basly

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento

Agradecimientos

El desarrollo y conclusión de este trabajo es gracias al apoyo y guía de todas las personas que me acompañaron en este camino.

Quiero agradecer a mi grupo de amigos, con los que nunca faltaron las salidas llenas de risas y quienes me enseñaron la importancia de compartir todo con seres queridos.

Especialmente quiero agradecer a mi amigo Lusho, con el que hicimos todos y cada uno de los trabajos grupales, de alguna manera incluso los que eran al azar. Agradecerte porque me enseñaste a vivir en esta nueva ciudad, porque sea lo que sea me apañaste con una sonrisa y porque siempre me inspiró para sacar lo mejor de mí y hacer un buen trabajo.

Agradecerle también a mi amigo Marco, la primera persona con la que me junté cuando llegué a Concepción y con la que nunca nos separamos. Tu hiciste lo que fue un inicio solo, en tardes muertas de la risa, salidas a cualquier parte y juntas para cocinar cosas dudosas. Gracias por todo tu apoyo y por todo lo que nunca dudaste en compartir conmigo.

Agradecerle al Feltex, por su amistad sincera y por las risas en el laboratorio. Gracias por siempre apoyarme en los desafíos del laboratorio, por siempre aprender y trabajar juntos, mientras nos reíamos, y por nunca abandonarme ni siquiera en los retos que nos llegaban.

Le agradezco a la vida, por conocer a mi amorcillo Tini. Te agradezco por todo el apoyo que me has dado desde el primer momento, porque siempre me encaminas y me despejas las dudas con tu amor, porque aunque no te des cuenta eres la fuerza para sobrepasar cualquier obstáculo. Y muchas gracias por enseñarme de la mano la belleza de esta ciudad, y las cosas bonitas de la naturaleza y la humanidad.

Finalmente, agradecer al profesor Bernardo y a Don José, los que me soportaron y guiaron en este camino. Los que nunca me cerraron la puerta y me apoyaron en cada idea que tuve. Gracias por enseñarme la importancia del orden, del trabajo, de ser buena persona y por encaminarme a (espero) ser un gran profesional.

*Dedicada a mi madre,
la persona más buena en el mundo
quien desde siempre se ha esforzado por nosotros
quien me ha apoyado, amado y guiado en cada momento de mi vida,
a quien le debo toda la felicidad que tengo.*

*Y a mis abuelos,
quienes me educaron y acompañaron,
los que fueron los grandes pilares de mi hogar,
espero estén orgullosos de este logro*

Resumen

En la actualidad, el ambiente espacial es uno de los recursos más explotados, aumentando cada año las misiones que se llevan a cabo en el. Estas misiones demandan un alto nivel de exactitud y precisión a la hora de controlar las naves espaciales. Sin embargo, en las naves donde se tienen grandes apéndices flexibles como antenas, paneles solares, velas solares, entre otros, se ha evidenciado que los algoritmos de control actuales no son capaces de mantener un correcto funcionamiento debido a las vibraciones inducidas por los apéndices. Esto dado que los modelos generalmente utilizados corresponden a modelos rígidos o flexibles simplificados. Es aquí, donde las redes neuronales artificiales (ANN) se presentan como una buena opción para aproximar su dinámica de una manera eficiente. En este trabajo se aborda este desafío al desarrollar un modelo en base a redes neuronales artificiales de la dinámica de cuerpo rígido de un cubesat-6U como primer acercamiento y estudio de su capacidad de trabajo con estos sistemas para su posterior implementación como modelo de dinámica flexible.

La metodología contempló la implementación de una estructura de simulación de alta fidelidad en el software Basilisk. Estas simulaciones se llevaron a cabo para una órbita baja (LEO) y contemplaron las cuatro perturbaciones ambientales más significativas: arrastre atmosférico, presión por radiación solar, gradiente de gravedad y campo electromagnético terrestre. Además, se incorporaron tres ruedas de reacción (RW) como controladores de actitud, las cuales mediante señales pseudoaleatorias (PRBS) excitaron el sistema, asegurando la captura de su dinámica. Se realizaron 2744 simulaciones con condiciones iniciales de velocidad angular y acciones de control diferentes. Con los datos extraídos de las simulaciones se procedió al entrenamiento de diferentes arquitecturas de redes Perceptrón Multicapa (MLP), las cuales se sometieron a tres procesos de discriminación que determinaron el impacto de cada variable en el desempeño de las redes, entregando cuatro modelos funcionales.

Los resultados indican que la dinámica interna de las RW constituye la mayor fuente de complejidad no lineal del sistema. Además, se demostró que incrementar el número de variables de entrada no se traduce necesariamente en mejores predicciones; de hecho, la exclusión de variables como los cuaterniones o inercias constantes benefició el ajuste. El modelo con mejor rendimiento global (ANN C) requirió únicamente tres variables de entrada: la velocidad angular del satélite (ω_{sat}), las acciones de control (u_{ctrl}) y la velocidad angular de las RW (ω_{rw}). Este modelo logró un Error Relativo Medio (MRE) de 2.14% tanto en la predicción del instante siguiente como en ventanas recursivas, requiriendo tan solo 62.691 parámetros entrenables y 123.904 operaciones de punto flotante (Flop).

Abstract

The space environment is currently one of the most utilized resources, with the number of missions there increasing annually. These missions demand a high level of accuracy and precision in spacecraft control. However, for spacecraft with large flexible appendages such as antennas, solar panels, and solar sails, existing control algorithms have proven incapable of maintaining proper operation due to the vibrations induced by these components. This limitation arises because the models typically employed are either rigid-body models or simplified flexible models. Artificial Neural Networks (ANNs) offer a promising solution for efficiently approximating this dynamic behavior. This study addresses this challenge by developing an ANN-based model of the rigid-body dynamics of a cubesat-6U, acting as an initial approach to evaluating the technology’s suitability for such systems.

The methodology involved implementing a high-fidelity simulation framework using Basilisk software. Simulations were made for a Low Earth Orbit (LEO) scenario and were added the four most significant environmental perturbations: atmospheric drag, solar radiation pressure, gravity gradient, and the Earth’s electromagnetic field. Additionally, three reaction wheels (RW) were incorporated as attitude actuators; these excited the system using pseudo-random binary sequences (PRBS) to ensure the capture of its dynamic behavior. A total of 2,744 simulations were performed, varying initial angular velocity conditions and control inputs. Data extracted from these simulations were used to train various Multilayer Perceptron (MLP) network architectures; after three stage discrimination process, these converged into four functional models.

The results indicate that the internal dynamics of the reaction wheels are the primary source of nonlinear complexity within the system. Furthermore, the study demonstrated that increasing the number of input variables does not necessarily lead to improved predictions; actually, excluding variables such as quaternions or constant inertia values improved the model fit. The model with the best overall performance (ANN C) required only three input variables: the satellite’s angular velocity (ω_{sat}), the control actions (u_{ctrl}), and the angular velocity of the reaction wheels (ω_{rw}). This model achieved a Mean Relative Error (MRE) of 2.14% for both single-step-ahead prediction and recursive windows, requiring only 62,691 trainable parameters and 123,904 floating-point operations (FLOPs).

Contenidos

Lista de Figuras	viii
Lista de Tablas	xi
Nomenclatura	xii
1 Introducción	1
1.1 Contexto	1
1.2 Objetivos	4
1.3 Metodología	5
1.4 Descripción del trabajo	5
2 Marco Teórico	7
2.1 Marcos de Referencia	7
2.2 Cuaterniones	8
2.3 Órbita baja y sus perturbaciones	9
2.4 Dinámica de Actitud en Órbita	11
2.5 Redes Neuronales Artificiales	13
2.5.1 Arquitectura Perceptrón Multicapa (MLP)	14
2.5.2 Funciones de Activación	14
2.5.3 Entrenamiento y Optimización	15
2.5.4 Métricas de desempeño	15
3 Simulaciones	17
3.1 Cubesat y parámetros físicos	17
3.2 Condiciones de Simulación	18
3.3 Definición de acciones de control y condiciones iniciales	20
3.4 Extracción y procesamiento de datos	22
4 Entrenamiento de la red neuronal	24
4.1 Análisis de complejidad del sistema para una red neuronal artificial	25
4.2 Determinación de entradas	27
4.3 Entrenamiento con entradas de pasos anteriores	29
4.4 Definición de métricas de desempeño basadas en requerimientos de control	31
5 Resultados	33
5.1 Errores de exactitud	33
5.2 Tamaño y tiempo de predicción	34
6 Discusión	37
7 Conclusiones	40
Anexo A Carta Gantt	45
Anexo B Figuras Primera Discriminación	46
Anexo C Figuras Segunda Discriminación	47
Anexo D Figuras Tercera Discriminación	49

Anexo E Figuras Resultados Velocidades angulares	50
Anexo F Figuras Resultados Cuaterniones	51
Anexo G Figuras Resultados Velocidades angulares en ventanas	52
Anexo H Figuras Resultados Cuaterniones en ventanas	53
Anexo I Figuras Resultados entrenamiento progresivo	54

Lista de Figuras

Figura 2.1	Marco de Referencia Inercial	7
Figura 2.2	Marco de Referencia LVLH	8
Figura 2.3	Gráficos de Actitud vs Tiempo de los 3 escenarios simulados.	13
Figura 2.4	Arquitectura de un MLP	14
Figura 3.1	Modelos de cubesat-6U.	17
Figura 3.2	Orbita Baja utilizada para la simulación.	19
Figura 4.1	Gráficos de correlación entre los valores reales y las predicciones de $\Delta\omega_{x,sat}$ de las diferentes ANN	26
Figura 4.2	Gráficos de correlación entre los valores reales y las predicciones de $\Delta\omega_{x,sat}$ de las diferentes Red Neuronal Artificial (ANN) presentadas en la Tabla 4.3.	28
Figura 4.3	Gráficos de correlación entre los valores reales y las predicciones de $\Delta\omega_{x,sat}$ de las diferentes ANN.	30
Figura 5.1	Gráficos de correlación entre los valores reales y las predicciones de las variables $\Delta\omega_{x,sat}$ y q_0 de las diferentes ANN presentadas en la Tabla 5.1.	35
Figura 5.2	Gráficos de predicción recurrente para ventanas de 1s de las variables reales y predichas de $\omega_{x,sat}$ y q_0 de las diferentes ANN presentadas en la Tabla 5.1.	36
Figura B.1	Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de una red entrenada con simulaciones sin perturbaciones ni ruedas de reacción	46
Figura B.2	Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de una red entrenada con simulaciones con perturbaciones y sin ruedas de reacción	46
Figura B.3	Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de una red entrenada con simulaciones sin perturbaciones y con ruedas de reacción	46
Figura B.4	Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de una red entrenada con simulaciones con perturbaciones y ruedas de reacción	46
Figura C.1	Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{A}$	47
Figura C.2	Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{B}$	47
Figura C.3	Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{C}$	47
Figura C.4	Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{D}$	48
Figura C.5	Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{E}$	48

Figura D.1	Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{C}_2$	49
Figura D.2	Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{D}_2$	49
Figura D.3	Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{F}_2$	49
Figura E.1	Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{C}$ al ser entrenada con 2000 simulaciones	50
Figura E.2	Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{F}$ al ser entrenada con 2000 simulaciones	50
Figura E.3	Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{C}_2$ al ser entrenada con 2000 simulaciones	50
Figura E.4	Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{F}_2$ al ser entrenada con 2000 simulaciones	50
Figura F.1	Gráficos de correlación entre las variables predichas y reales de cada componente del cuaternion, de la ANN $\mathbb{C}$ al ser entrenada con 2000 simulaciones	51
Figura F.2	Gráficos de correlación entre las variables predichas y reales de cada componente del cuaternion, de la ANN $\mathbb{F}$ al ser entrenada con 2000 simulaciones	51
Figura F.3	Gráficos de correlación entre las variables predichas y reales de cada componente del cuaternion, de la ANN $\mathbb{C}_2$ al ser entrenada con 2000 simulaciones	51
Figura F.4	Gráficos de correlación entre las variables predichas y reales de cada componente del cuaternion, de la ANN $\mathbb{F}_2$ al ser entrenada con 2000 simulaciones	51
Figura G.1	Gráficos de predicción recurrente para ventanas de 1s de las variables reales y predichas de cada componente del vector ω_{sat} , para la ANN $\mathbb{C}$	52
Figura G.2	Gráficos de predicción recurrente para ventanas de 1s de las variables reales y predichas de cada componente del vector ω_{sat} , para la ANN $\mathbb{F}$	52
Figura G.3	Gráficos de predicción recurrente para ventanas de 1s de las variables reales y predichas de cada componente del vector ω_{sat} , para la ANN $\mathbb{C}_2$	52
Figura G.4	Gráficos de predicción recurrente para ventanas de 1s de las variables reales y predichas de cada componente del vector ω_{sat} , para la ANN $\mathbb{F}_2$	52
Figura H.1	Gráficos de predicción recurrente para ventanas de 1s de las variables reales y predichas de cada componente del cuaternion, para la ANN $\mathbb{C}$	53
Figura H.2	Gráficos de predicción recurrente para ventanas de 1s de las variables reales y predichas de cada componente del cuaternion, para la ANN $\mathbb{F}$	53
Figura H.3	Gráficos de predicción recurrente para ventanas de 1s de las variables reales y predichas de cada componente del cuaternion, para la ANN $\mathbb{F}_2$	53
Figura H.4	Gráficos de predicción recurrente para ventanas de 1s de las variables reales y predichas de cada componente del cuaternion, para la ANN $\mathbb{F}_2$	53
Figura I.1	Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{C}$ al ser entrenada con 5 simulaciones	54

Figura I.2	Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{F}$ al ser entrenada con 5 simulaciones	54
Figura I.3	Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{C}_2$ al ser entrenada con 5 simulaciones	54
Figura I.4	Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{F}_2$ al ser entrenada con 5 simulaciones	54
Figura I.5	Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{C}$ al ser entrenada con 50 simulaciones	55
Figura I.6	Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{F}$ al ser entrenada con 50 simulaciones	55
Figura I.7	Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{C}_2$ al ser entrenada con 50 simulaciones	55
Figura I.8	Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{F}_2$ al ser entrenada con 50 simulaciones	55
Figura I.9	Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{C}$ al ser entrenada con 500 simulaciones	56
Figura I.10	Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{F}$ al ser entrenada con 500 simulaciones	56
Figura I.11	Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{C}_2$ al ser entrenada con 500 simulaciones	56
Figura I.12	Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{F}_2$ al ser entrenada con 500 simulaciones	56

Lista de Tablas

Tabla 2.1	Datos de las simulaciones.	12
Tabla 3.1	Parámetros Físicos cubesat-6U.	18
Tabla 4.1	Conjunto de hiperparámetros para la búsqueda aleatoria.	24
Tabla 4.2	Arquitecturas de Simulación.	25
Tabla 4.3	Conjuntos de entradas para cada ANN.	27
Tabla 4.4	Conjunto de entradas para cada ANN.	29
Tabla 4.5	Métricas de desempeño para evaluación de modelos en base a ANN. . . .	32
Tabla 5.1	Arquitectura Oculta Final de cada ANN.	34
Tabla 5.2	Parámetros entrenables y cantidad de Flop para cada ANN.	34
Tabla 6.1	Resumen de resultados.	37

Nomenclatura

ADRC	Control Activo con Rechazo de Perturbaciones
ANCF	Formulación de Coordenadas Nodales Absolutas
ANN	Red Neuronal Artificial
CM	Centro de Masa
CP	Centro de Presión
DNN	Red Neuronal Profunda
ESO	Observador de Estado Extendido
FAS	Sistemas Completamente Actuados
FEM	Método de Elementos Finitos
Flop	Operaciones de punto flotante
GCI	Geocentric Inertial Frame
GMM	Método de Modos Globales
JAXA	Agencia de Exploración Aeroespacial de Japón
LEO	Low Earth Orbit
LVLH	Local Vertical-Local Horizontal
MAE	<i>Mean Absolute Error</i>
MLP	<i>Multi Layer Perceptron</i>
MPC	Control Predictivo basado en Modelos
MRE	<i>Mean Relative Error</i>
MSE	Mean Squared Error
PD	Proporcional-Derivativo
PID	Proporcional-Integral-Derivativo
PINN	Red Neuronal Físicamente Informada
PRBS	Pseudo Random Binary Signal
RW	Ruedas de Reacción
SRP	Solar Radiation Pressure
SRTM	<i>Shuttle Radar Topography Mission</i>
TMD	Amortiguadores de Masas Sintonizados

1 Introducción

1.1 Contexto

En la industria espacial, cada año son más las misiones que son llevadas a cabo con el fin de observar la Tierra, mejorar las comunicaciones, estudiar el espacio exterior e incluso intentar llegar a otros planetas [1]. Estas misiones demandan un alto nivel de exactitud y precisión a la hora de controlar la actitud de las naves espaciales. Sin embargo, se ha evidenciado que en los satélites donde se cuenta con grandes apéndices conectados al cuerpo principal, como antenas o paneles solares, entre otros, los sistemas de control no son capaces de mantener un correcto funcionamiento debido a las vibraciones inducidas por estas estructuras. Un ejemplo de esto se encuentra en la importante misión del telescopio espacial Hubble, donde la precisión del apuntado se vio comprometida debido a vibraciones provenientes de los paneles solares del satélite [2] o en la misión del observatorio espacial de dinámica solar, donde también se vio comprometida la calidad de las imágenes [3]. Adicionalmente, la necesidad de nuevas tecnologías que puedan expandir la frontera espacial, va de la mano con el diseño de estructuras flexibles en las naves espaciales. Por ejemplo, la misión IKAROS de la Agencia de Exploración Aeroespacial de Japón (JAXA) desarrolló e implementó una vela solar de 20 metros de largo para verificar los efectos de la presión de radiación solar, y su impacto en la navegación y control [4], dado su potencial como sistema propulsivo en el espacio.

Generalmente, para modelar la dinámica de estos satélites se utilizan herramientas analíticas y numéricas basadas en la mecánica clásica, aplicadas mediante tres principales enfoques: movimiento rígido, movimiento rígido+flexible y movimiento flexible [5]. El enfoque de movimiento rígido es el más común de los enfoques. Este asume que la posición relativa entre las partículas es constante, lo cual deriva en un modelo simple y de bajo costo computacional, pero que no reproduce las vibraciones internas del cuerpo. Por otro lado, el enfoque de movimiento rígido+flexible divide el problema en dos, primero abordándolo como un cuerpo rígido y luego modelando por separado las partes flexibles bajo el campo de fuerza resultante. El problema con esto es que la interacción entre el movimiento flexible y el rígido no es considerada. Por último, en el enfoque de movimiento flexible sí se tienen en cuenta las interconexiones entre las variables que definen la dinámica rígida y flexible, resolviendo el problema en simultáneo, pero a costa de un elevado costo computacional.

En base a estos enfoques se han desarrollado distintos métodos de modelamiento, siendo uno de los más utilizados el Método de Modos Globales (GMM), el cual permite desarrollar un modelo dinámico analítico de orden reducido [6]. Primero se obtienen las ecuaciones de movimiento para los apéndices flexibles y el cuerpo rígido central por separado, para luego obtener las frecuencias naturales y modos de vibrar tras derivar y aplicar condiciones de borde a las ecuaciones características. Así se obtiene un conjunto de ecuaciones dinámicas desacopladas de orden reducido. Sin embargo, este método depende mucho del tipo de estructura flexible que se tenga, ya que para ciertas estructuras puede ser complejo obtener los modos de vibrar. Otra forma es a través del Método de Elementos Finitos (FEM) [6], el cual consiste en dividir toda la estructura en pequeños elementos interconectados de los cuales se resuelven las ecuaciones de movimiento de cada uno para así obtener una fiel aproximación del comportamiento dinámico de todo el cuerpo. El problema es que el método converge en un modelo de alta dimensión lo cual complica su aplicación en el diseño de controladores.

La importancia del desarrollo de estos modelos radica en que son la base para el diseño de controladores que puedan evitar y suprimir las vibraciones en la estructura. Para abordar

este desafío se ha propuesto utilizar controladores de la familia del control robusto [7]. Estos son diseñados especialmente para mantener la estabilidad y el desempeño del sistema aún frente a incertidumbres o perturbaciones no modeladas. Para ello utilizan métodos como el H_∞ o la μ -síntesis los cuales garantizan que el sistema de control pueda responder frente a las perturbaciones inducidas por las vibraciones y cumplir su misión principal. En [8] se propuso un sistema de control de actitud robusto el cual logra realizar maniobras rápidas y de alta precisión, al mismo tiempo que suprime las vibraciones. Para ello se basa en la teoría de Sistemas Completamente Actuados (FAS) y Control Activo con Rechazo de Perturbaciones (ADRC). Primero se transforma el modelo dinámico no-lineal del vehículo espacial en un sistema completamente actuado, el cual mantiene la estructura del satélite a la vez que reduce la complejidad del modelo dinámico. Luego, implementa el control activo con rechazo de perturbaciones a través de un Observador de Estado Extendido (ESO), el cual estima en tiempo real la magnitud de las perturbaciones (vibraciones) las cuales son utilizadas para cancelarlas activamente. Así se asegura la supresión de vibraciones sin comprometer las expectativas de la misión.

Otras estrategias desarrolladas para suprimir las vibraciones son a través de la adición de elementos de supresión pasiva o activa. La supresión pasiva consiste en diseñar la estructura con elementos que disipen la energía vibratoria de manera natural, como los Amortiguadores de Masas Sintonizados (TMD), analizados por [9] para su implementación en satélites. Tras un análisis teórico, se concluye que al incluir TMD bajo las ruedas de reacción, su frecuencia de resonancia se desplaza disminuyendo las microvibraciones del sistema. Además, se deja abierta la investigación sobre su utilización en conexiones con apéndices flexibles. En contraste la supresión activa consiste en añadir sensores distribuidos por la estructura, para recopilar información de las vibraciones en tiempo real y poder suprimirlas mediante actuadores. Shengyu Lin [10] simuló el comportamiento del satélite de la misión *Shuttle Radar Topography Mission* (SRTM) el cual contaba con un gran mástil incorporado al cuerpo central del satélite. El autor demostró que al añadir actuadores piezoeléctricos estratégicamente distribuidos en la estructura mediante la técnica Gramiana, junto con un controlador de retroalimentación directa de velocidad, se lograba reducir en un 70% el desplazamiento de la punta del mástil en 0.5s.

Si bien los métodos y estrategias presentados en muchos casos han demostrado un buen rendimiento teórico, en la realidad se tienen muchas limitaciones que dificultan implementarlos en las misiones. El primer problema se presenta en la creación de un modelo dinámico el cual sea preciso y manejable al mismo tiempo, para el diseño de controladores [11]. Los modelos de alta fidelidad obtenidos mediante FEM resultan en miles de ecuaciones de movimiento, lo que se traduce en un mayor tiempo de simulación y carga computacional; si se quisiera desarrollar un controlador que se base en un modelo FEM, se tendría un vector de estado de un tamaño incapaz de realizar cálculos y trabajar con él. En el estudio de mejora del telescopio espacial Hubble incluso se encontraron inconsistencias que los obligaron a dedicar más horas en corregirlos y adaptar los modelos [2]. Por otro lado, el uso de técnicas de reducción de orden puede llevar a la pérdida de dinámicas claves para el control de las vibraciones [7]. Además, los modelos lineales frecuentemente utilizados en controladores desarrollados en base al método H_∞ no son capaces de capturar las no-linealidades geométricas que introducen los apéndices flexibles ni las incertidumbres en parámetros del modelo, como el cambio de la matriz de inercia debido al consumo de combustible. Los controladores robustos, pensados para lidiar con estas incertidumbres y perturbaciones necesitan de una excitación constante para estimar parámetros de control, lo cual puede incluso inducir a más vibraciones. Finalmente la utilización de aisladores pasivos y activos suele verse reducida a

casos muy específicos [9]. Los aisladores pasivos están limitados a trabajar en una banda de frecuencia estrecha y para pequeñas vibraciones, por lo que cuando se enfrentan a componentes con características que varían en el tiempo, como la rigidez en los rodamientos, pierden efectividad. Por otro lado, los controladores activos si bien pueden manejar una gran cantidad de tipos de vibraciones, en distintas frecuencias y amplitudes, aumentan la complejidad del sistema, tamaño y peso, lo cual es una gran penalización en la industria espacial.

En la actualidad, existe una creciente tendencia en emplear enfoques basados en aprendizaje automático para modelar la dinámica de estructuras flexibles. Por ejemplo, Tomas Slimak [12] se centra en el uso de una ANN para predecir la evolución temporal de sistemas flexibles, reduciendo el alto costo computacional que implican los métodos tradicionales. Slimak desarrolla un esquema que combina dos ANN: una encargada de aproximar las ecuaciones de movimiento en función de posiciones y velocidades, y otra que actúa como integrador numérico para propagar los estados en el tiempo. Este método fue validado en el caso de una viga en voladizo modelada con elementos finitos bajo la Formulación de Coordenadas Nodales Absolutas (ANCF), mostrando que la aproximación logra capturar adecuadamente el comportamiento dinámico, incluso frente a datos con ruido y en escenarios fuera del rango de entrenamiento.

De forma análoga, en el área del modelamiento de robots blandos, caracterizados por su flexibilidad y grados de libertad infinitos, se utilizan diferentes modelos de aprendizaje automático para abordar tareas como el control dinámico, alcance de objetivos, seguimiento de trayectorias y navegación, las cuales son un desafío debido a la no-linealidad de los robots blandos. Según Zixi Chen [13] destacan tres enfoques del aprendizaje automático: los modelos estadísticos, las ANN y el aprendizaje por refuerzo. Aplicar estos enfoques presentan ventajas como la representación de características reales de los robots, a diferencia de los modelos físicos que están basados en simplificaciones e hipótesis, y la capacidad para aproximar funciones no-lineales. En [14] se propuso y evaluó en un manipulador robótico flexible de dos eslabones un método de aprendizaje en red mejorado para la evaluación de la fiabilidad del movimiento en mecanismos flexibles. Los resultados mostraron que este método tiene precisión del 99.9% y una alta eficiencia al reducir el tiempo de cómputo de 100 simulaciones de 29.4s a 0.14s.

Al profundizar en el área satelital se destacan dos trabajos, el del autor Carlo Cena [15] y de Seongji Han [16]. Carlo Cena utiliza el Control Predictivo basado en Modelos (MPC) para controlar la actitud de un satélite. En este caso, el controlador utiliza una Red Neuronal Físicamente Informada (PINN) para predecir el comportamiento que tendrá el satélite N pasos de tiempo en el futuro, según una acción de control que se le aplique. Así, según los requerimientos de la acción que se quiere llevar a cabo, el controlador puede evaluar cual es la forma más eficiente de llegar a la referencia. Dentro del marco del control, las PINN son especialmente prácticas dado que utilizan una función de costo no solo basada en datos, sino que también en leyes físicas, lo cual se traduce en respuestas con sentido físico y replicables por el controlador, a diferencia de otras ANN que solo entregan el valor que minimiza el error. A través del simulador Basilisk, Carlo Cena adquiere datos sobre la dinámica rígida de un satélite y, mediante una función de costos híbrida basada en datos y la ley de conservación del momento angular, entrena una PINN y la implementa en un MPC. Su trabajo demuestra que la inclusión de la PINN al controlador, otorga mayor estabilidad y robustez al sistema al reducir el error de rendimiento de estabilidad en un 16.67%. Por otro lado, Seongji Han utiliza una Red Neuronal Profunda (DNN) para modelar el proceso de despliegue de un panel solar en un satélite como un sistema de multicuerpos, obteniendo

una predicción de las posiciones y respuestas como la tensión y deformación. Este modelo demostró una precisión del 99.9% en sus variables de salida y respuestas más rápidas que métodos tradicionales al tener un tiempo de cómputo de 32.6 ms frente a 57 ms.

Es así como las técnicas de aprendizaje automático aplicadas a estructuras flexibles muestran potencial para reducir el coste computacional y capturar no linealidades, pero hay aspectos críticos a tener en cuenta; estos estudios no consideran el acoplamiento con algoritmos de control de actitud ni efectos espaciales (variación de inercia o interacción con ruedas de reacción). Además, muchos de estos trabajos no son lo bastante eficientes para cumplir con los límites físicos con los que cuentan los sistemas satelitales, ni permiten un análisis de estabilidad y márgenes necesario para el diseño de controladores. Esta brecha es especialmente crítica si quiere converger en una solución práctica, ya que sin modelos que capturen con precisión la dinámica y que además sean eficientes, los controladores no podrán identificar ni cancelar las fuentes reales de vibración. Por lo anterior, se propone diseñar un modelo basado en redes neuronales artificiales para la dinámica de un satélite como cuerpo rígido, pero orientado a aplicaciones de control. De esta forma se busca evaluar, tanto la capacidad de las ANN para trabajar con estos sistemas aeroespaciales, como su futura complejización para su uso en sistemas flexibles.

El desarrollo de esta propuesta puede llegar a tener repercusiones en el ámbito económico y tecnológico. Al desarrollar un método de modelado basado en simulaciones y ANN, evitando la inversión de horas hombre en la formulación analítica de un modelo, se reducen tiempos y costos.

1.2 Objetivos

Objetivo General

Desarrollar modelo en base a redes neuronales artificiales para la dinámica de actitud de un cubesat-6U en órbita baja orientado a aplicaciones de control

Objetivos Específicos

- **OE1:** Implementar modelación dinámica de un cubesat como cuerpo rígido basado en principios físicos que obtenga los ángulos de Euler ante un caso de perturbación externa tipo torque.
- **OE2:** Definir cualitativamente requerimientos y métricas de desempeño para el modelo dinámico basado en redes neuronales artificiales para control de actitud.
- **OE3:** Generar un conjunto de datos en el simulador Basilisk para la dinámica de actitud de un satélite como cuerpo rígido considerando perturbaciones ambientales y actuadores internos.
- **OE4:** Entrenar modelos en base a redes neuronales artificiales para propagar la actitud del satélite en función de las condiciones iniciales, perturbaciones externas y actuadores internos.
- **OE5:** Evaluar el desempeño del modelo seleccionado en base a un nuevo conjunto de datos de pruebas.

1.3 Metodología

Para cumplir con el objetivo principal de la investigación, el trabajo se estructura en tres fases: simulación, entrenamiento de ANN y evaluación.

Primera fase: Simulación

Para poder entrenar una ANN se necesita un conjunto de datos que sean capaz de captar y describir la dinámica de actitud de un satélite. Sin embargo, dada la ausencia de un modelo tangible en el cual realizar ensayos y mediciones, se emplea el entorno de simulación Basilisk [17]. Este corresponde a una librería gratuita de Python, donde se pueden realizar simulaciones de alta fidelidad.

La investigación se realiza en base al modelo de cubesat-6U, M6P NanoAvionics, el cual se programa con sus paneles solares desplegados con el fin de captar la dinámica de esta configuración. Además, para lograr una alta fidelidad y teniendo en cuenta que los cubesats se suelen utilizar en misiones de órbita baja, se agregan las siguientes cuatro perturbaciones ambientales:

- Arrastre atmosférico
- Presión por radiación solar
- Efecto del gradiente de gravedad
- Campo electromagnético de la tierra

Segunda fase: Entrenamiento de ANN

Para el proceso de entrenamiento, se utilizan las librerías gratuitas de aprendizaje automático Scikit-Learn [18] y Tensorflow [19]. Así se asegura continuidad en el lenguaje de programación Python y se aprovecha su capacidad de manejar grandes cantidades de datos.

De la librería Scikit-Learn, se utiliza la estructura de ANN llamada Perceptrón Multicapa. Esta estructura se destaca por su capacidad para aproximar problemas no-lineales, gracias al uso de funciones de activación como la tangente hiperbólica. Por otro lado, la determinación de las variables de entradas, así como también los criterios de evaluación para la construcción de la función de costo, se fundamentan a través de una revisión bibliográfica de las necesidades y funcionamiento de los controladores diseñados actualmente para la tarea de control de actitud de un satélite, asegurando que la ANN contenga características relevantes para el control.

Tercera fase: Evaluación

Un problema de los modelos desarrollados en base a datos, es el sobreajuste que pueden desarrollar a los datos utilizados para su entrenamiento, afectando así su capacidad de predicción. Para evaluar el modelo se estudia el error de predicción frente a un nuevo conjunto de datos que no haya sido utilizado en el entrenamiento. Esto permite cuantificar la fidelidad de la ANN como un sustituto confiable para los modelos algebraicos.

1.4 Descripción del trabajo

El plan de trabajo, ver ANEXO A, se proyecta para el periodo de tiempo entre la semana del 08 de Marzo al 10 de Agosto, dividido en lapsos de tiempo de una semana. Para el cumplimiento de cada objetivo específico, y en base a la metodología presentada, se plantean distintas actividades que siguen un orden lógico para facilitar el estudio y aplicación a medida

que se desarrolla la investigación. Finalmente, se contemplan semanas de desarrollo de los hitos del proyecto, los cuales se presentan como informes de avance entregables al comité evaluador.

OE1:

Para cumplir con este objetivo específico primero se establecen las ecuaciones del movimiento que rigen la actitud del cubesat. Además, se identifican las perturbaciones y los actuadores que formarán parte del sistema, y que se deben incorporar en las ecuaciones. Una vez establecidas las ecuaciones del movimiento, se determinan los parámetros físicos del cubesat (masa, volumen, inercia, etc) y los valores de las perturbaciones. Luego se integran las ecuaciones del movimiento mediante métodos numéricos y se analiza la fidelidad de la simulación.

OE2:

Las constantes que caracterizan a un controlador dependen del objetivo de control y la forma en la que se requiere lograr este objetivo. Teniendo en cuenta que se busca diseñar un controlador para un cubesat, mediante búsqueda bibliográfica se determina y cuantifica cuáles son los parámetros más importantes que debe seguir un controlador bajo este tipo de misiones. Además, de determinar cómo estos parámetros se traducen en métricas de desempeño que debe cumplir el modelo de ANN para aportar en el diseño del controlador.

OE3:

Para obtener datos sobre la dinámica del cubesat, se utiliza el simulador Basilisk. Dentro del simulador se programan las perturbaciones, características físicas del satélite y actuadores (Ruedas de Reacción (RW)) que estarán presentes en la misión. Además, se deben programar las acciones de control y el momento en el que se llevarán a cabo, junto con las variables de interés que se deben almacenar durante toda la simulación.

OE4:

Para el entrenamiento y selección de hiperparámetros de la ANN, primero se construye una función de costo acorde a las métricas de desempeño anteriormente impuestas. Luego, durante el entrenamiento de la ANN, se utilizan herramientas de ajuste de hiperparámetros, como la búsqueda de malla, para determinar la configuración que minimice la función de costos.

OE5:

Para el objetivo específico cinco, se genera un nuevo conjunto de datos, que no haya participado en el entrenamiento de la ANN, con el cual se evalúa el desempeño de esta. Además, se analiza si es posible y como mejorar el resultado obtenido al realizar nuevas iteraciones.

2 Marco Teórico

2.1 Marcos de Referencia

Para describir la posición, orientación o el movimiento de un cuerpo, se requiere definir previamente un marco de referencia, el cual se establece mediante un origen y la orientación de sus ejes de coordenadas [20]. Dentro del área espacial, la selección de un marco de referencia adecuado es de suma importancia; dado que según la aplicación, una correcta elección permite simplificar la resolución de ecuaciones de movimiento o facilitar el trabajo de estrategias de control de actitud. A continuación se presentan tres marcos de referencias convencionales y utilizados en el desarrollo de este proyecto.

Geocentric Inertial Frame

El Geocentric Inertial Frame (GCI) es un marco de referencia casi inercial que tiene su origen en el centro de la tierra y sus ejes fijos en las estrellas. Se define como un sistema casi inercial, ya que su origen mantiene una aceleración lineal en su órbita alrededor del sol, a diferencia de los sistemas inerciales definidos por una aceleración nula y sin rotaciones en el tiempo. Sin embargo, los efectos de dicha aceleración son despreciables para el estudio de la dinámica de actitud.

La orientación de este sistema de referencia se representa mediante la triada $\{i_1, i_2, i_3\}$. Como se ilustra en la Figura 2.1, el eje i_3 , está alineado con el polo norte promedio de la Tierra, mientras que el eje i_1 , apunta al equinoccio vernal promedio, y por último el eje i_2 , se define mediante el producto cruz $i_2 = i_3 \times i_1$, completando el sistema coordenado.

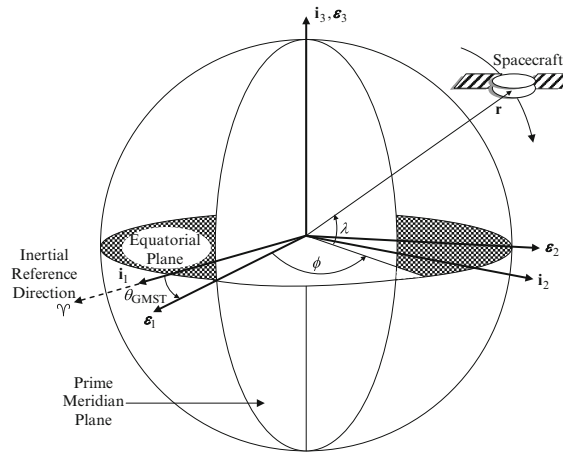


Figura 2.1 Marco de Referencia Inercial [20], $\{i_1, i_2, i_3\}$.

Marco de Referencia Fijo al Cuerpo

Un marco de referencia fijo al cuerpo corresponde a uno con su origen fijo en algún punto del cuerpo, generalmente al Centro de Masa (CM), y con sus 3 ejes, $\{x, y, z\}$ rotando conjuntamente con el cuerpo respecto al marco inercial. Es una práctica estándar orientar los ejes de este marco de referencia en las mismas direcciones que los ejes principales de inercia del cuerpo. Mediante esta alineación, se obtiene una matriz de inercias diagonal, lo cual simplifica las ecuaciones de Euler y reduce la carga computacional requerida para propagar la actitud.

Local Vertical - Local Horizontal

Para misiones espaciales donde el objetivo principal es el apuntado hacia la superficie de la Tierra, existe un marco de referencia que es especialmente útil, el Local Vertical-Local Horizontal (LVLH). A diferencia de los marcos inerciales, tiene un origen móvil que acompaña al satélite en su desplazamiento a través de la órbita. La orientación de este marco se representa mediante la triada $\{o_1, o_2, o_3\}$. El eje o_3 está alineado con el vector nadir, el cual apunta en todo momento directo al centro de la Tierra. El eje o_2 apunta en el sentido contrario a la velocidad angular de la órbita, y el eje o_1 es tangente a la órbita en el mismo sentido que el desplazamiento, tal como se observa en la Figura 2.2.

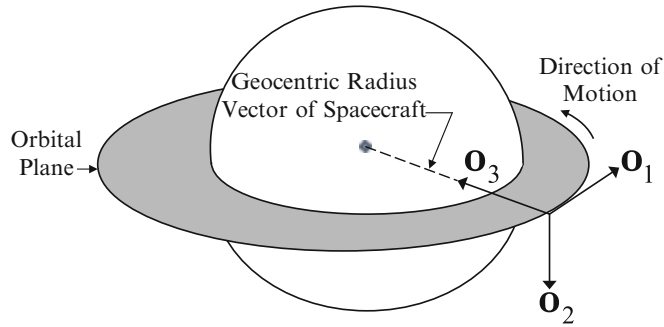


Figura 2.2 Marco de Referencia LVLH [20], $\{o_1, o_2, o_3\}$.

2.2 Cuaterniones

Para representar las rotaciones de los cuerpos se suele utilizar los ángulos de Euler o los ángulos Yaw-Pitch-Roll. Sin embargo, en el espacio donde es común tener rotaciones completas en cualquier eje, estos sistemas sufren de singularidades cinemáticas. Cuando un cuerpo se acerca a estas singularidades, algunas derivadas de la velocidad angular tienden a infinito, lo cual imposibilita la integración numérica y propagación de su dinámica en el tiempo.

Para solucionar este problema, se propuso un nuevo sistema llamado cuaterniones. Este está basado en el teorema de rotación de Euler, el cual postula que cualquier par de sistemas de coordenadas cartesianas están relacionados por una rotación única alrededor de una sola línea que pasa por su origen común [21]. Así, se puede obtener por ejemplo, el marco de referencia fijado al cuerpo, $\{x, y, z\}$, desde el marco de referencia inercial, $\{X, Y, Z\}$ mediante una rotación de magnitud θ alrededor del eje de Euler denotado por el vector unitario $\hat{u}$, como se presenta en el siguiente conjunto de ecuaciones.

$$\begin{aligned}\hat{i} &= \cos(\theta)\hat{I} + (1 - \cos(\theta))(\hat{u} \cdot \hat{I}) + \sin(\theta)\hat{u} \times \hat{I} \\ \hat{j} &= \cos(\theta)\hat{J} + (1 - \cos(\theta))(\hat{u} \cdot \hat{J}) + \sin(\theta)\hat{u} \times \hat{J} \\ \hat{k} &= \cos(\theta)\hat{K} + (1 - \cos(\theta))(\hat{u} \cdot \hat{K}) + \sin(\theta)\hat{u} \times \hat{K}\end{aligned}$$

Matemáticamente, un cuaternion $\hat{q}$ es un vector de cuatro componentes que agrupa la parte vectorial q_1, q_2, q_3 asociada al eje de Euler, y otra parte escalar q_4 asociada a la rotación en base a dicho eje.

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}$$

La parte vectorial corresponde a las componentes del eje de Euler escaladas por $\sin(\theta/2)$, mientras que la parte escalar corresponde al $\cos(\theta/2)$. Además, con el fin de representar orientaciones puras, sin alterar la escala del objeto, los cuaterniones deben cumplir la restricción de norma unitaria $\|q\| = 1$.

Para propagar como cambia la actitud de un cuerpo en el tiempo, se relaciona la velocidad angular del cuerpo $\omega = [\omega_x, \omega_y, \omega_z]^T$ con la tasa de cambio del cuaternion. Esta relación se expresa mediante la ecuación diferencial

$$\dot{q} = \frac{1}{2}\Omega(w)q \quad (1)$$

La cual expresada en su versión matricial corresponde al sistema de ecuaciones diferenciales:

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}$$

2.3 Órbita baja y sus perturbaciones

La región espacial Low Earth Orbit (LEO), comprendida entre los $160km$ a los $2000km$ de altitud [22], es la zona más cercana a la Tierra donde se puede orbitar y concentra la mayor densidad de misiones satelitales. Esta región es de particular interés estratégico para las misiones espaciales, ya que por su cercanía a la superficie terrestre se reducen los costos de poner satélites en órbita, además de que mantienen una baja latencia en las telecomunicación y permiten realizar observación terrestre con una alta resolución.

No obstante, esta región se caracteriza por la presencia de perturbaciones ambientales significativas que afectan tanto la órbita como la actitud de los satélites. Por ejemplo, debido a la baja altitud de la órbita aún existen remanentes de la atmósfera terrestre que interactúan con el satélite generando arrastre mientras orbitan, provocando la continua disminución de velocidad y altura. También si existe algún dipolo magnético en el satélite, el campo electromagnético de la tierra afectará en gran medida la actitud de este. A continuación, se describen las cuatro perturbaciones ambientales predominantes en esta región, junto con las ecuaciones que las rigen y sus efectos en el satélite.

Arrastre Atmosférico

Aunque el 99.9% de la atmósfera de la Tierra se concentra por debajo de los $100km$ de altitud, la densidad residual presente en LEO en conjunto con las altas velocidades orbitales, es suficiente para generar una fuerza aerodinámica de arrastre. Esta perturbación es capaz de reducir la velocidad del cuerpo y con ello su altitud [21]. La fuerza de arrastre sobre un cuerpo se puede modelar como

$$F_{drag} = \frac{1}{2}\rho v_{rel}^2 C_D A$$

donde ρ corresponde a la densidad de la atmósfera local, v_{rel} la velocidad del satélite relativa a la atmósfera, C_D el coeficiente de arrastre aerodinámico y A es el área del satélite que enfrenta a la atmósfera.

Es importante destacar que parámetros como la densidad varían significativamente según la posición orbital, mientras que el área expuesta depende directamente de la orientación instantánea del satélite. Por otro lado, el arrastre no es una fuerza que solo influya en la órbita del satélite, sino que también en la dinámica actitud. La fuerza de arrastre es distinta dependiendo el tipo de superficie, ángulo de incidencia con el viento, entre otras cosas; estas diferencias resultan en que el Centro de Presión (CP) no coincida exactamente con el CM. Esta diferencia en la posición genera un brazo de palanca que se traduce en un torque sobre el satélite.

Gradiente de Gravedad

La atracción gravitacional entre dos cuerpos, no actúa solo sobre sus centros de masa, sino que actúa sobre cada una de sus partículas. Esto, sumado a que la intensidad de la fuerza es inversamente proporcional al cuadrado de la distancia, se traduce en un campo gravitatorio que no es estrictamente uniforme sobre los cuerpos, sino que afectará en mayor medida a las partículas que se encuentren a menos separadas. En los satélites, las partes de este que se encuentren más cercanos a la Tierra experimentan una mayor atracción que las demás. Esta diferencia de fuerzas, genera un torque perturbador el cual busca de manera natural alinear el eje principal de menor inercia con el vector Nadir.

Al considerar un cuerpo rígido y utilizar el sistema de referencia LVLH en conjunto con los ángulos yaw = ϕ , pitch = θ y roll = ψ , para describir la actitud del satélite relativo al sistema LVLH, se puede calcular el torque generado mediante

$$M_{GG} = \frac{3\mu}{r^3} \begin{bmatrix} (I_3 - I_2)\cos^2(\theta)\cos(\psi)\sin(\psi) \\ (I_3 - I_1)\cos(\theta)\sin(\theta)\cos(\psi) \\ (I_1 - I_2)\cos(\theta)\sin(\theta)\sin(\psi) \end{bmatrix}$$

donde μ corresponde al parámetro gravitatorio de la Tierra e I_n el momento de inercia alrededor del centro de masa del satélite [20].

Presión por radiación solar

Los fotones son partículas sin masa que viajan a la velocidad de la luz y que conforman todo el espectro de radiación electromagnética. A pesar de no tener masa, su energía y cantidad de momento son distintos de cero, por lo que al impactar en la superficie de los cuerpos transfieren este momento y lo perturban, fenómeno conocido como presión por radiación solar [21]. En el espacio, el Sol es la principal fuente de radiación electromagnética y entre mayor es la cercanía con el mismo, mayor es la intensidad. Si se asume un modelo bala de cañón para el cuerpo del satélite, y se desprecia la variación de distancia con el sol, dado que el diametro de la orbita del satélite es despreciable con respecto a la distancia entre el Sol y la Tierra, se puede calcular la fuerza que genera esta perturbación sobre el cuerpo como:

$$F_{SRP} = -\nu \frac{S}{c} C_r A_{sc} \hat{u}$$

donde $\hat{u}$ es el vector unitario que apunta desde el satélite hacia el Sol, A_{sc} es el área de absorción del satélite (en este caso πR^2), ν es la función sombra, C_r , S la constante solar y c la velocidad de la luz.

Esta corresponde a una forma muy simplificada de calcular la intensidad de esta perturbación. En modelos más complejos, el área expuesta a la luz y su ángulo de incidencia juega un papel fundamental y no es definida solo como la superficie total del cuerpo. Por esto, conocer la orientación del satélite y posición del sol es de suma importancia en misiones espaciales. Además, esta fuerza no actúa solo en el centro de masa del satélite, sino que se distribuye por toda su superficie, generando un torque que afecta a la actitud del satélite.

Torque Electromagnético

La perturbación por torque electromagnético se produce debido a la interacción entre el momento dipolar magnético residual en satélites, generado por componentes electrónicos o bucles de corriente, y el campo magnético de la Tierra. Esta interacción genera un par de fuerzas que intenta alinear el dipolo del satélite con el campo magnético local, al igual que una brújula se alinea con el polo norte. Este torque se determina mediante el producto cruz:

$$M_{mag} = m \times B$$

donde m corresponde al momento dipolar magnético residual del satélite y B al vector del campo magnético local de la tierra [20]. Es fundamental destacar que el vector B va a depender de la posición que tenga el satélite en el momento. Por lo anterior, para poder propagar esta perturbación junto con las anteriores, no solo hay que centrarse en la actitud sino que en el desplazamiento del satélite.

Para propagar el movimiento del satélite en una órbita circular, se debe resolver el problema de los dos cuerpos, donde solo se considera la interacción entre el satélite y la Tierra. Así, se tiene que la aceleración del satélite esta dada por

$$\ddot{r} = -\frac{\mu}{||r||^3}r \quad (2)$$

donde r es el vector posición del satélite y μ el parámetro gravitacional de la Tierra.

2.4 Dinámica de Actitud en Órbita

La evolución temporal de la rotación de un satélite se describe a través de las ecuaciones de Euler, las cuales relacionan la variación del momento angular total con los torques externos aplicados en el cuerpo. Si el origen del marco de referencia fijo al cuerpo coincide con el CM del cuerpo, la ecuación de movimiento rotacional expresada en este mismo marco toma la siguiente forma:

$$I \frac{d\omega}{dt} + \omega \times H_G = M_T \quad (3)$$

donde I corresponde al tensor de inercia del satélite, ω a la velocidad angular del satélite respecto al marco inercial, $H_G = I\omega$ al momento angular del satélite y M_T corresponde al momento total externo ejercido sobre el satélite compuesto por los momentos de perturbación externos antes visto. Para un caso de estudio simplificado, M_T se define como la diferencia entre una perturbación externa $M_{environment}$ y un momento de control $M_{control}$

$$M_T = M_{ambiental} - M_{control}$$

A partir de las ecuaciones (1) y (3) se implementa un entorno de simulación para la propagación de la dinámica de actitud. La actitud del satélite se propaga mediante cuaterniones integrando la ecuación (1) para evadir singularidades, mientras que las variaciones en su velocidad angular se propagan mediante la ecuación (3).

Estas simulaciones se configuran tomando como referencia el modelo de cubesat-6U “M6P” de NanoAvionics [23]. Para este vehículo, el tensor de inercia principal, expresado en el marco fijo al cuerpo, se define como:

$$I_{6U} = \begin{bmatrix} 70.61 & 0 & 0 \\ 0 & 155.00 & 0 \\ 0 & 0 & 105.64 \end{bmatrix}_{xyz} \cdot 10^{-3} \text{ kg} \cdot \text{m}^2$$

Se establece una orientación inicial del marco del cuerpo ($\{x, y, z\}$) rotada 45° alrededor de los ejes x e y con respecto al marco inercial ($\{X, Y, Z\}$), lo cual se representa mediante el cuaternión $q_0 = [-0.1464 \ 0.3536 \ 0.3536 \ 0.8536]^T$. Con el propósito de validar la fidelidad física del propagador este se enfrenta a tres escenarios distintos cuyos parámetros se especifican en la Tabla 2.1.

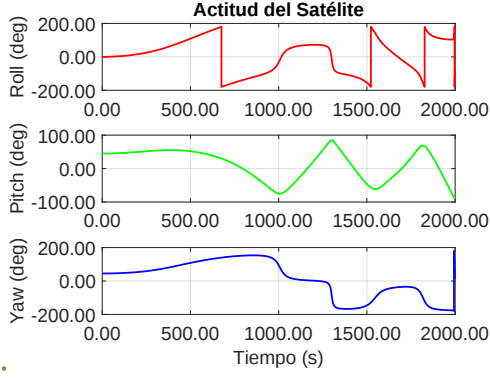
- **Escenario 1:** Se ejerce un torque ambiental constante al satélite, el cual al ser externo al satélite se expresa en el marco de referencia inercial.
- **Escenario 2:** Se ejerce solo un torque de control, el cual al ser parte del sistema del satélite, se expresa en el marco de referencia del cuerpo.
- **Escenario 3:** Se inicia con una condición inicial de velocidad angular distinta de 0

Tabla 2.1 Datos de las simulaciones.

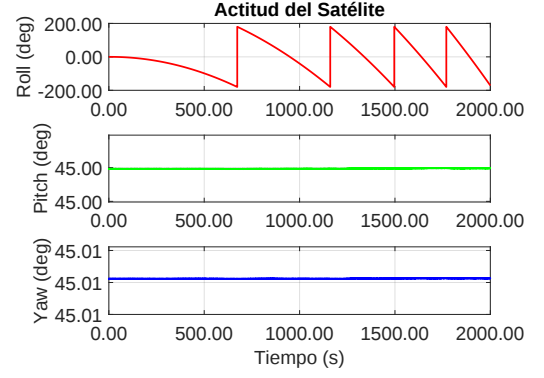
Simulación	$[\omega_0]_{xyz} \frac{\text{rad}}{\text{s}}$	$[M_{\text{ambiental}}]_{XYZ} [Nm]$	$[M_{\text{control}}]_{xyz} [Nm]$
1	$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \cdot 10^{-6}$	$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$
2	$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \cdot 10^{-6}$
3	$\begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \cdot 10^{-4}$	$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$

En la Figura 2.3 se presentan los resultados de la propagación de actitud del satélite para los tres escenarios propuestos. Para el primer escenario, donde se aplicó un torque ambiental externo constante expresado en el marco inercial, se observa que el satélite experimenta rotaciones acopladas en sus tres ejes (ver Figura 2.3(a)). Esto ocurre debido a que la orientación del marco del cuerpo cambia continuamente con respecto a la orientación del marco inercial, lo cual se traduce en que el torque ambiental se proyecta en componentes variables sobre los ejes de inercia del cuerpo a medida que el satélite gira. Por esto, las aceleraciones angulares que experimenta el satélite no son constantes y resultan en un comportamiento angular sin un patrón fijo, a diferencia de los otros 2 escenarios.

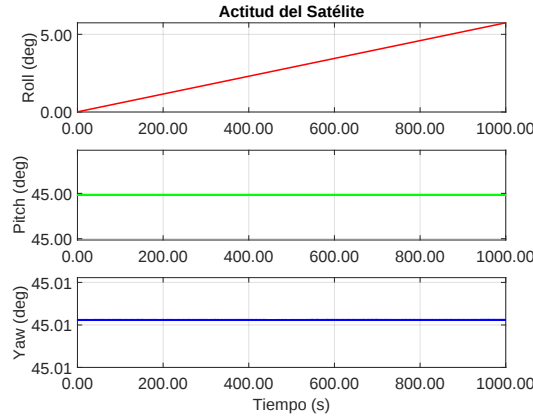
En el segundo escenario, ver Figura 2.3(b), se observa que el satélite rota exclusivamente alrededor del eje x (roll), mientras que los demás ejes permanecen constantes. A diferencia del caso anterior, el torque de control se define y fija solidariamente al marco de referencia del satélite. Al estar anclado a un eje principal de inercia, la perturbación afecta únicamente a dicho eje sin importar la actitud. Por otro lado, como el torque aplicado es constante, se induce una aceleración angular constante en el eje x , lo que a su vez provoca un aumento lineal de la velocidad angular. Este aumento lineal de la velocidad angular provoca que el satélite cumpla una rotación cada vez en menor tiempo, lo cual en el gráfico se visualiza mediante saltos en intervalos de tiempo cada vez más cortos. Cabe destacar, que estos saltos, visibles también en la Figura 2.3(a), se deben al rango en el que se representan los ángulos de actitud, en este caso $[-180, 180]$ grados. Debido a esto, cuando la rotación del satélite supera cualquiera de los límites, su valor inicial se reinicia en el límite opuesto, ya que tener una rotación de 180° es equivalente a una de -180° .



(a) Escenario Número 1.



(b) Escenario Número 2.



(c) Escenario Número 3.

Figura 2.3 Gráficos de Actitud vs Tiempo de los 3 escenarios simulados.

Finalmente, en el tercer escenario, ver Figura 2.3(c), se observa la evolución del ángulo de roll ante una velocidad angular inicial constante en un entorno sin torques externos. Como resultado, el ángulo de roll aumenta con una pendiente constante a lo largo del tiempo. Además, al no haber perturbaciones externas ni formas de disipación de energía, el momento angular se conserva y se mantiene una rotación uniforme.

2.5 Redes Neuronales Artificiales

Las ANN son modelos computacionales basados en la estructura de un cerebro humano [24]. Principalmente se utilizan para procesar información y aprender patrones complejos a partir de un conjunto de datos. Al igual que en los humanos, su unidad estructural básica se llama neurona, y son las encargadas de recibir un conjunto de entradas (x) a las cuales le asignan un peso (w) y suman en conjunto de un término adicional llamado sesgo (b). Finalmente al resultado de esta operación lineal se le aplica una función matemática llamada función de activación (f), que define la salida de la neurona

$$\hat{y} = f\left(\sum_{i=1}^n w_i x_i + b\right).$$

2.5.1 Arquitectura Perceptrón Multicapa (MLP)

Dentro de las distintas arquitecturas que existen de ANN, *Multi Layer Perceptron* (MLP) es una de las más fundamentales. El MLP se destaca por su capacidad de aproximar sistemas no lineales mediante el uso sucesivo de un conjunto de regresiones lineales y funciones de activación no lineales aplicadas a través de diferentes etapas [18].

Como se observa en la Figura 2.4, estructuralmente un MLP se organiza en capas secuenciales:

- **Capa de entrada:** Es la capa que recibe los datos sin procesar. El número de neuronas en esta capa corresponde exactamente al número de variables de entrada del sistema.
- **Capas ocultas:** Son las capas intermedias donde ocurre el procesamiento principal y la extracción de características no lineales. Una red puede tener múltiples capas ocultas conectadas secuencialmente.
- **Capa de salida:** Es la capa final que entrega la predicción o estimación del modelo, donde su cantidad de neuronas corresponde a la dimensión de la variable objetivo.

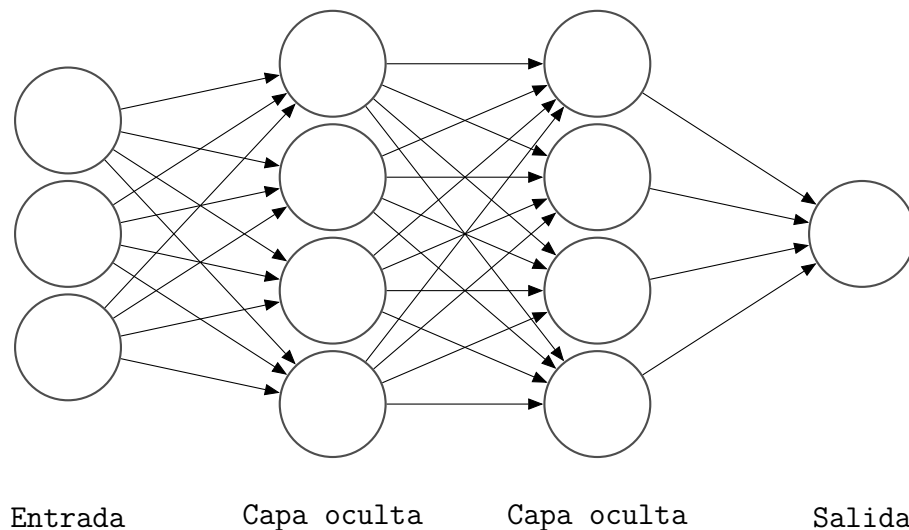


Figura 2.4 Arquitectura de un MLP

2.5.2 Funciones de Activación

Las funciones de activación son los componentes matemáticos cruciales que se aplican a la salida de las neuronas en las capas ocultas. Su función principal es permitirle a la red neuronal aprender y aproximar dinámicas complejas y no lineales. Para el diseño y búsqueda de hiperparámetros de redes neuronales aplicadas se suelen evaluar diversas funciones de activación, entre las que destacan [19]:

- **Tanh (Tangente Hiperbólica):** Sirve para mapear los valores de entrada a un rango entre -1 y 1, centrando los datos alrededor del cero.
- **ReLU (Rectified Linear Unit):** Es una función definida a tramos que retorna directamente la entrada si es positiva, o cero si es negativa. Es computacionalmente eficiente y ayuda a mitigar el problema del desvanecimiento del gradiente.
- **ELU (Exponential Linear Unit):** Es similar a ReLU para valores positivos, pero utiliza una curva exponencial suave para valores negativos, lo que permite valores

medios de activación más cercanos a cero.

- **SiLU (Swish):** Definida como $f(x) = x \cdot \sigma(x)$ (donde σ es la función sigmoide), es una función suave y no monótona que ha demostrado mejorar el rendimiento en redes profundas.

2.5.3 Entrenamiento y Optimización

El proceso de entrenamiento de una ANN consiste en un algoritmo iterativo mediante el cual la red ajusta sus pesos y sesgos internos para que sus predicciones se acerquen lo más posible a los valores reales. Para cuantificar el error durante el entrenamiento y guiar la optimización de los parámetros, se utiliza una función de costo, donde la más utilizada corresponde al Mean Squared Error (MSE) [18]

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2,$$

donde y corresponden a los valores reales, $\hat{y}$ a las predicciones de la red y n al número total de observaciones. El algoritmo de entrenamiento minimiza este MSE a través de métodos de gradiente descendente y retropropagación. Un recorrido completo de todo el conjunto de entrenamiento a través de la red constituye una época; sin embargo, debido a la naturaleza iterativa del algoritmo, una sola época raras veces es suficiente para alcanzar la convergencia, por lo que el proceso se repite durante múltiples épocas hasta minimizar el error.

Durante el proceso de desarrollo de una ANN, además de los pesos existen otros parámetros que no dependen de los datos y se deben ajustar manualmente, estos son los hiperparámetros. A diferencia de los parámetros internos, que son aprendidos automáticamente por el algoritmo de optimización mediante la minimización de la función de costo, los hiperparámetros se definen con anterioridad y determinan tanto la arquitectura del modelo como la dinámica del propio entrenamiento. Entre los más relevantes se encuentran la tasa de aprendizaje (learning rate), que controla el tamaño de los pasos en la dirección del gradiente descendente; el número de épocas y el tamaño de lote (batch size); así como variables estructurales tales como la cantidad de capas ocultas, el número de neuronas por capa y las funciones de activación. La correcta configuración de estos hiperparámetros resulta fundamental para garantizar la convergencia del modelo y evitar problemas como el subajuste o el sobreajuste.

El subajuste y sobreajuste son fenómenos que pueden ocurrir durante el proceso de optimización de los hiperparámetros, e indican un mal desempeño de la ANN. El subajuste ocurre cuando el modelo carece de la complejidad suficiente para capturar los patrones subyacentes de los datos, presentando un error elevado tanto en el entrenamiento como en la evaluación. Por el contrario, el sobreajuste se produce cuando la red memoriza el ruido y las particularidades específicas del conjunto de entrenamiento, perdiendo su capacidad de generalización frente a datos no vistos. Para monitorear el desempeño del modelo y prevenir estos problemas, los datos disponibles se dividen típicamente en un conjunto de entrenamiento, utilizado para el ajuste de los pesos, y un conjunto de validación, el cual evalúa de forma independiente la precisión del modelo durante las épocas y permite detener el entrenamiento o seleccionar los hiperparámetros óptimos.

2.5.4 Métricas de desempeño

Para evaluar la calidad de las estimaciones y comparar la capacidad de generalización de distintas arquitecturas de ANN, es necesario cuantificar el error entre los valores observados

y las predicciones utilizando un conjunto de validación o prueba. Si bien el MSE se utiliza habitualmente como función de costo durante el entrenamiento, en la etapa de evaluación se recurre a un conjunto complementario de métricas que permiten interpretar el error desde diferentes perspectivas.

El coeficiente de determinación (R^2) evalúa la proporción de la varianza total de la variable objetivo que es explicada por el modelo ANN. Su valor máximo es uno e indica una predicción perfecta, mientras que un valor de cero indica que el modelo no es mejor que ocupar una estimación constante del promedio de los datos. Su fórmula tiene la siguiente forma

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}.$$

Por otro lado, si las magnitudes tienen significado físico, se suele utilizar el *Mean Absolute Error* (MAE). Este mide, mediante un promedio simple, el promedio de las magnitudes de los errores individuales sin considerar su signo. Al usar un promedio simple, resulta menos sensible a valores atípicos y ofrece una medida representativa del error promedio del modelo:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|.$$

Una forma de cuantificar el error con un porcentaje es a través del *Mean Relative Error* (MRE). Este cuantifica la magnitud del error absoluto en proporción del valor real observado, entregando una medida adimensional de la precisión del modelo. Resulta particularmente útil para comparar el desempeño entre variables con distintas órdenes de magnitud o escalas:

$$\text{MRE} = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{|y_i|} \cdot 100 \, \%.$$

Cuando la variable objetivo del modelo no es un escalar sino un vector, la métrica se extiende para evaluar un error relativo general en todas las dimensiones. Para ello, se promedia el MRE calculado sobre cada una de las componentes del vector:

$$\text{MRE}^Y = \frac{1}{M} \sum_{j=1}^M \left(\frac{1}{N} \sum_{i=1}^N \frac{|Y_{i,j} - \hat{Y}_{i,j}|}{|Y_{i,j}|} \right) \cdot 100 \, \%, \quad (4)$$

donde Y representa el vector objetivo, M corresponde al número de sus componentes y N es el número total de muestras observadas.

Finalmente, en arquitecturas o configuraciones orientadas a realizar predicciones recurrentes, es fundamental evaluar cómo evoluciona la degradación del error en el tiempo. Para ello se generaliza el error agrupando el análisis en ventanas de predicción:

$$\text{MRE}_{v,N}^Y = \frac{1}{M} \sum_{j=1}^M \left[\frac{1}{W} \sum_{w=1}^W \left(\frac{1}{N} \sum_{i=1}^N \frac{|Y_{w,i,j} - \hat{Y}_{w,i,j}|}{|Y_{w,i,j}|} \right) \right] \cdot 100 \, \%, \quad (5)$$

donde W indica el número total de ventanas de predicción evaluadas, N representa la cantidad de pasos o puntos estimados de forma recurrente en cada ventana y j cada una de las M componentes del vector de salida.

3 Simulaciones

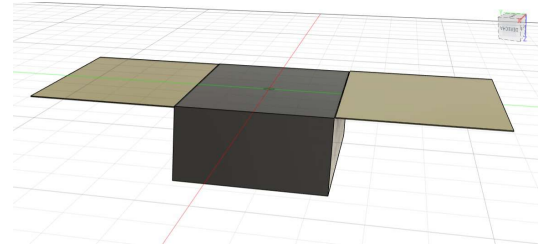
Basilisk es un entorno de simulación desarrollado como una librería de Python que permite simular el comportamiento dinámico complejo entre cuerpos celestes y todo tipo de naves espaciales. Este se destaca principalmente por su modularidad y capacidad de configuración, lo que permite ejecutar simulaciones de satélites en órbita con alta fidelidad. En este capítulo, se presenta el desarrollo y configuración de la arquitectura de simulación utilizada para la recopilación del conjunto de datos, que posteriormente servirá para el entrenamiento de las ANN.

3.1 Cubesat y parámetros físicos

Para la ejecución de las simulaciones se selecciona el modelo de cubesat-6U “M6P” de NanoAvionics [23]. Este modelo se configura dentro del simulador con sus dos paneles solares desplegados, ver 3.1(a), los cuales se consideran como apéndices extendidos y cuyos parámetros de masa y geometría se encuentran integrados dentro de la matriz de inercia total del cubesat. Es importante destacar que se considera que la masa total del cubesat está uniformemente distribuida por todo su volumen y su geometría es simétrica, por lo que presenta una matriz de inercias diagonal, como se observa en la Tabla 3.1.



(a) Modelo Real Nanoavionics.



(b) Modelo Representativo simplificado.

Figura 3.1 Modelos de cubesat-6U.

Propiedades Aerodinámicas

Para el cálculo de la fuerza de arrastre atmosférico, se considera un acabado superficial homogéneo en toda la superficie del satélite. Esto se traduce en un coeficiente de arrastre aerodinámico constante al cual se le da un valor de $C_D = 2.2$ según el estándar presentado en [25].

Propiedades Ópticas

Para la Solar Radiation Pressure (SRP) el modelo se programa con dos superficies diferentes: los paneles solares y la del cuerpo central del cubesat. Debido a la diferencia en materiales

y funciones, los coeficientes de reflexión especular C_{sp} y reflexión difusa C_{df} varían significativamente entre ambas partes. Como se observa en la Tabla 3.1, la reflectividad total $C_r = C_{sp} + C_{df}$ de los paneles solares es tres veces menor que la del cuerpo principal. Esto, ya que el diseño de los paneles solares busca minimizar la reflexión y maximizar la absorción de fotones incidentes para la generación de energía.

Propiedades Magnéticas

Finalmente, otro aspecto crítico en la fabricación de un satélite para LEO, es la interacción con el campo electromagnético de la Tierra. Generalmente, el campo magnético residual de los componentes internos suele ser opacado por magnetorques instalados en el satélite, pero en este caso de estudio no se consideran dichos actuadores. Por esto, el satélite interactúa únicamente con el campo magnético terrestre mediante el campo magnético residual del satélite, y cuyos valores son entregados por el proveedor y especificados en la Tabla 3.1.

Tabla 3.1 Parámetros Físicos cubesat-6U.

Parámetro	Valor
Inercia Satélite [$kg \cdot m^2$]	$\begin{bmatrix} 70.61 & 0 & 0 \\ 0 & 155.00 & 0 \\ 0 & 0 & 105.64 \end{bmatrix}_{xyz} \cdot 10^{-3}$
Inercia RW [$kg \cdot m^2$]	$\begin{bmatrix} 1.469 & 0 & 0 \\ 0 & 1.469 & 0 \\ 0 & 0 & 2.938 \end{bmatrix}_{rw} \cdot 10^{-5} \quad kg \cdot m^2$
Volumen del cuerpo [m^3]	$0.3 \cdot 0.2 \cdot 0.1$
Superficie Paneles Solares [m^2]	$0.3 \cdot 0.2$
Coefficiente de arrastre C_D [—]	2.2
Campo Residual [Am^2]	$[0.01 \quad 0.05 \quad 0.01]_{xyz}$
C_{sp} Cuerpo [—]	0.5
C_{df} Cuerpo [—]	0.1
C_{sp} Paneles [—]	0.1
C_{df} Paneles [—]	0.1

3.2 Condiciones de Simulación

El comportamiento del cubesat se pone a prueba en una órbita circular baja, identificada por los siguientes parámetros de Kepler y visible en la Figura 3.2. Esta órbita es el estándar para misiones de satélites de este tipo.

- Semieje mayor: 7371km
- Excentricidad: 0.0001
- Inclinação: 33.3°
- Nodo Ascendente: 48.2°
- Argumento del Perigeo: 347.8°
- Anomalía Verdadera: 85.3°

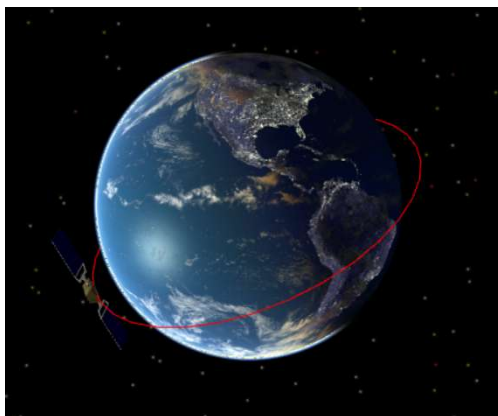


Figura 3.2 Órbita Baja utilizada para la simulación.

Además, se programan las perturbaciones de arrastre atmosférico, SRP, gradiente de gravedad y campo electromagnético de la Tierra, que son las perturbaciones predominantes en esta región.

Programación arrastre atmosférico

Para programar el arrastre atmosférico, se utiliza el módulo “facetDragDynamicEffector”. El funcionamiento de este módulo se basa en calcular el CP de la fuerza de arrastre total sobre el satélite mediante la suma de las fuerzas generadas en cada área programada del satélite. Si no coincide el CP con el CM, entonces esta fuerza se traduce en un torque externo sobre el cuerpo. Para realizar este cálculo, aparte de los parámetros físicos del satélite, el módulo necesita información instantánea del cuerpo y su geometría 3D. La geometría del cubesat se programa construyéndolo cara a cara, para ello se entrega cada vector que va desde el CM hacia el centro de cada cara que se quiera añadir, junto con el vector normal a la superficie de la cara indicando la dirección en la que mira y son válidas sus propiedades físicas. Por otro lado, debe conocer la posición del satélite, ya que esta se utiliza para estimar las características de la atmósfera local, y la orientación para estimar que caras y en que ángulo incide el flujo de viento.

Programación Presión por Radiación Solar

La **SRP** se implementa con el módulo “facetSRPDynamicEffector”. Al igual que en la fuerza de arrastre, se calcula el efecto de la incidencia de la luz en cada superficie del satélite y con ello el torque generado por la diferencia en la ubicación del CP y CM. Es importante destacar que si bien la SRP esta condicionada, en parte por la posición del satélite y la sombra de la Tierra, y debido a la duración de la maniobra, se considera que el satélite se encuentra en todo momento iluminado por el Sol, siendo de vital importancia el vector posición entre el Sol y el satélite.

Programación Torque por campo electromagnético

Para el torque por el campo electromagnético de la Tierra es necesario implementar dos módulos, el “magneticFieldWMM” para replicar el campo electromagnético de la Tierra, y el “MtbEffector” para implementar el dipolo magnético residual en el cubesat. Para replicar el campo de la Tierra, el simulador cuenta con una base de datos del comportamiento de

este en un periodo de 4 años. Por otro lado, el segundo módulo se utiliza para instalar un imán en el satélite, el cual constantemente, y según la posición y orientación del satélite, interactúa e intenta alinearse con el campo local de la Tierra. De esta interacción se calcula el torque generado y se aplica sobre el satélite.

Programación Gradiente de Gravedad

En el caso del gradiente de gravedad, este se incorpora mediante el módulo “GravityGradientEffector”. Este módulo utiliza la información de la posición inercial del satélite, así como también su orientación y matriz de inercias. Con esta información se calcula como el campo gravitatorio afecta a todo el cuerpo y el torque que este genera.

3.3 Definición de acciones de control y condiciones iniciales

En el área de la identificación de sistemas, la recolección de datos requiere de un proceso activo donde se debe excitar el sistema para obtener la mayor cantidad de información dinámica posible [26]. El objetivo primordial de una señal de entrada (excitación) es garantizar que el conjunto de datos resultante sea máximamente informativo, proporcionando las condiciones necesarias para caracterizar la dinámica oculta del sistema y discriminar sin error entre los diferentes modelos candidatos. Para obtener la información necesaria, las acciones de control encargadas de excitar el sistema deben cumplir con dos propiedades fundamentales: persistencia y orden de la excitación. Una señal se define como persistentemente excitante de orden n si su espectro de potencia es estrictamente mayor a cero en al menos n puntos distintos. Además, el orden de la persistencia de la excitación debe ser igual o superior al orden del sistema que se desea identificar.

Para cumplir con dichas propiedades, existen distintos modelos de entradas ya diseñados y que se pueden extrapolar a distintos sistemas. En [26] se recomienda el uso de las Pseudo Random Binary Signal (PRBS). Una PRBS es una señal binaria, periódica y determinista, con propiedades espectrales similares a las del ruido blanco. Estas señales se generan a través de polinomios característicos de orden n , donde el periodo máximo de la señal depende de este orden, siendo su longitud máxima $M = 2^n - 1$ bits.

Para obtener el orden del sistema y así generar una PRBS adecuada, es necesario analizar el espacio estados del sistema.

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (6)$$

donde $x(t)$ corresponde al vector de estados y $u(t)$ al vector de control. Al representar un modelo simple de dinámica de actitud, construido en base a las ecuaciones (1) y (3) en el espacio estados, se definen el vector de estados y de control como:

$$x(t) = \begin{bmatrix} w \\ q \end{bmatrix} = \begin{bmatrix} w_x \\ w_y \\ w_z \\ q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \quad u(t) = [M_{control}] = \begin{bmatrix} M_{x,control} \\ M_{y,control} \\ M_{z,control} \end{bmatrix}.$$

Se puede observar que el vector de estado cuenta con siete componentes y el vector de control con tres componentes. Por lo tanto, para asegurar una excitación completa del sistema, se necesita una PRBS de orden al menos $n = 10$.

Con la librería “scipy.signal” de Python, se generan diferentes secuencias PRBS que son independientes entre sí mediante los polinomios característicos de grado 10:

- $P_1 = x^{10} + x^9 + x^8 + x^5 + 1$
- $P_2 = x^{10} + x^9 + x^7 + x^6 + 1$
- $P_3 = x^{10} + x^9 + x^5 + x^2 + 1$
- $P_4 = x^{10} + x^9 + x^4 + x^2 + 1$
- $P_5 = x^{10} + x^8 + x^7 + x^2 + 1$
- $P_6 = x^{10} + x^8 + x^5 + x^4 + 1$

Así, se generan seis secuencias PRBS de una longitud de señal de $M = 1023$ bits. Además, dado que el satélite posee tres ejes de control y para obtener mayor variabilidad en los datos, se seleccionan dos PRBS para cada eje de actuación.

El control de actitud del satélite se efectúa mediante tres RW, modelo “RW0” de NanoAvionics [23], dispuestas una en cada eje principal de inercia del satélite. Estas ruedas ya se encuentran integradas dentro del catálogo interno de Basilisk, por lo que solo se seleccionan y posicionan en el satélite. Luego, para aplicar las secuencias PRBS a través de las RW, los valores binarios (0 y 1) se mapean a una señal de control de voltaje que alimenta su motor interno. Esta señal se configura como una onda cuadrada con una amplitud de $\pm 5V$, la cual el controlador interno de la rueda traduce en el torque electromecánico correspondiente aplicado al sistema. Otro aspecto importante en implementación de las PRBS, es que el cambio de las RW no puede ser instantáneo, ya que sino el tiempo puede ser insuficiente para que cada activación perturbe de manera visible el sistema. Para estos casos se le aplica un tiempo de mantenimiento a cada bit de la PRBS. Según las especificaciones de la rueda de reacción, al aplicar el torque máximo de forma constante sobre esta, se tarda $6.25s$ en llegar a su velocidad angular máxima. Teniendo en cuenta que la librería inicia las secuencia de PRBS en un vector de orden $n = 10$ lleno de 1, cada bit debe tener un tiempo de mantenimiento inferior a $0.625s$ para que no llegue a su máxima velocidad y luego mantenga su funcionamiento promedio cercano a su límite, por lo que se mantiene un tiempo de estabilización de $0.5s$ por bit. Así, las señales PRBS de entrada para cada RW tendrán la forma de una señal cuadrada de amplitud $\pm 5V$ y un ancho de pulso de $0.5s$

Finalmente, se determinan las condiciones iniciales de velocidad angular y actitud en la simulación. Un aspecto crítico a tener en cuenta para esto, es la capacidad máxima de almacenamiento de momento angular de las RW. Esto se traduce en que a cierta velocidad angular del satélite, RW no serán capaces de reducirla y controlar así su actitud. Para determinar las velocidades angulares máxima del satélite en cada eje, estas se calculan a partir de la inercia del cuerpo y el momento angular máximo de la RW ($h_{rw} = 0.020$ Nms):

$$\begin{aligned}
 I\omega &= h_{rw} \\
 \omega_{xx} &= \frac{h_{rw}}{I_{xx}} = \frac{0.020}{0.07061} = 0.2832 rad/s \\
 \omega_{yy} &= \frac{h_{rw}}{I_{yy}} = \frac{0.020}{0.155} = 0.1290 rad/s \\
 \omega_{zz} &= \frac{h_{rw}}{I_{zz}} = \frac{0.020}{0.10564} = 0.1893 rad/s.
 \end{aligned}$$

Para entrenar la red neuronal con un modelo que represente tanto el régimen controlable como el régimen saturado, se realiza un barrido de 7 puntos equidistantes en el rango de $-1.5\omega_{max}$ a $1.5\omega_{max}$ para cada eje de inercia. En cuanto a la actitud, la condición inicial se mantiene constante en el cuaternión identidad ($q_0 = [0, 0, 0, 1]^T$), asumiendo que el satélite inicia alineado con el sistema de referencia inercial.

$$N_{sim} = 7(\omega_x) \cdot 7(\omega_y) \cdot 7(\omega_z) \cdot 2(PRBS_x) \cdot 2(PRBS_y) \cdot 2(PRBS_z) = 2744.$$

Finalmente, la cantidad de simulaciones totales necesarias para cubrir todo el espacio de estados y combinaciones de excitación se determina por el producto del barrido de velocidades angulares y las dos señales PRBS asignadas a cada actuador.

3.4 Extracción y procesamiento de datos

Finalmente se determinan las variables que se deben extraer de las simulaciones para su posterior uso en el entrenamiento de las ANN. Para evaluar las variables que pueden aportar en el entrenamiento, se basa en la ecuación de movimiento rotacional de un satélite con ruedas de reacción incluidas. Expresada en el marco de referencia del cuerpo, esta ecuación tiene la forma de

$$I \frac{d\omega_{sat}}{dt} + \omega_{sat} \times (I\omega_{sat} + J\omega_{rw}) = M_{ambiental} \quad (7)$$

donde I y J corresponden al tensor de inercias del satélite y las RW respectivamente, ω_{sat} y ω_{rw} a las velocidades angulares del satélite y RW, y $M_{ambiental}$ a la suma de los torques ambientales externos que perturban el satélite. En esta ecuación se destacan 2 términos principales, el momento angular del satélite y RW, y los torques de perturbación ambiental. En base a lo anterior se propone extraer variables que aporten en el cálculo de estos términos.

Con respecto a las perturbaciones, la actitud del satélite (representada por cuaterniones) participa en el cálculo de todas las perturbaciones dentro del simulador, lo cual la convierte en la variable que más podría aportar a la ANN para captar las perturbaciones del sistema. Sin embargo, las variables de posición y velocidad del satélite también aportan en el cálculo de determinadas perturbaciones. La posición es importante para calcular tanto el vector del campo electromagnético de la Tierra, como las propiedades de la atmósfera. Por otro lado, la velocidad es necesaria para estimar la fuerza del arrastre atmosférico. Así, se opta por incluir estas 3 variables como posibles entradas de la ANN.

Para el cálculo de cada momento angular, correspondiente a los 2 términos restantes, es necesario conocer la inercia y velocidad angular de cada cuerpo, donde la velocidad angular de las ruedas de reacción dependerá del voltaje (acción de control) que se le ingrese a cada una. Así, para cada instante de la simulación, se extraen las siguientes variables:

- Cuaternion (q)
- Velocidad angular del satélite (ω_{sat})
- Acción de control (u)
- Posición Inercial (r)
- Velocidad inercial del satélite (v)
- Velocidad angular ruedas de reacción (ω_{rw})
- Inercia del satélite (I)
- Inercia de las ruedas de reacción (J)

Estas variables son extraídas de cada simulación, mediante un archivo .csv que contiene estas variables para cada instante de tiempo (0.1s) en cada fila. Sin embargo, cabe destacar que, mientras las demás variables no requieren de un previo procesamiento, se debe tener especial cuidado con los cuaterniones.

Cuando se utilizan cuaterniones unitarios para representar orientaciones en el espacio, estos presentan una ambigüedad llamada **doble recubrimiento** [20]. Esta es una propiedad geométrica la cual indica que para cualquier rotación en nuestro espacio tridimensional, existen 2 cuaterniones q y $-q$ que la representan, pero numéricamente estos residen en polos opuestos de una hiperesfera de cuatro dimensiones. Al extraer las series temporales de cuaterniones de las simulaciones, es posible que el cuaternión salte entre estas dos representaciones de un instante a otro, generando discontinuidades numéricas que luego perjudican el entrenamiento de la ANN. Para solucionar este problema y asegurar que los datos describan trayectorias rotacionales continuas, se evalúa la proximidad geométrica entre instantes consecutivos mediante el producto punto. Dado un cuaternión en el instante anterior q_{t-1} y el cuaternión actual q_t , se fuerza la continuidad aplicando la siguiente transformación algorítmica:

$$q_t = \begin{cases} q_t, & \text{si } q_t \cdot q_{t-1} \geq 0 \\ -q_t, & \text{si } q_t \cdot q_{t-1} < 0 \end{cases}.$$

Así, se garantiza una representación continua de las rotaciones del satélite la cual ya estaría lista para ser utilizada para entrenar una ANN.

4 Entrenamiento de la red neuronal

Como se introduce en la Sección 2.5, dentro de las distintas arquitecturas que existen de ANN, el MLP es una de las más básicas, pero que destaca por su capacidad de aproximar sistemas no lineales mediante un conjunto de regresiones lineales y funciones de activación no lineales en sus capas ocultas [18]. En base a lo anterior se utiliza la arquitectura de MLP para desarrollar el modelo de la dinámica de cuerpo rígido del cubesat-6U.

El objetivo del modelo es que sea capaz de predecir el siguiente estado dinámico del satélite, a partir de información actual como entrada. Como primer acercamiento al sistema, se utilizan las entradas de ω_{sat} y u_{ctrl} para predecir exclusivamente el cambio del estado $\Delta\omega_{sat}$ (variable objetivo). Dado que la cinemática de los cuaterniones, que relaciona ω_{sat} y la orientación instantánea es conocida y exacta (ecuación (1)), se omite el cuaternión como variable objetivo y se opta por propagar su orientación mediante integración numérica. Por otro lado, se define como variable objetivo el cambio del estado ya que; por un lado con el instante anterior y el cambio es posible construir el siguiente instante de tiempo, y principalmente por el comportamiento de los parámetros entrenables dentro del modelo. Dado que el tiempo de muestreo en las simulaciones es de $0.1s$ el cambio de las velocidades angulares del sistema es del orden de $10^{-4}rad/s$, correspondiendo en promedio al 0.1% de la magnitud total de la velocidad angular. Por lo anterior, cuando se entrena la ANN para predecir el siguiente estado de tiempo

$$x_{t+1} = x_t + f(x_t, u_t) \quad (8)$$

los parámetros entrenables se tenderán a ajustar al término con mayor peso, en este caso x_t que representa el 99.9% del estado siguiente, y prácticamente ignorando su variación $f(x_t, u_t)$. Por el contrario, si como variable objetivo se tiene solo la variación del estado, se restringe el entrenamiento de sus parámetros exclusivamente a la predicción del cambio.

Una vez determinada la variable objetivo, se prepara el proceso de entrenamiento de la MLP. Para ello mediante las librerías de Tensorflow [19] y Scikit-Learn [18] se desarrolla una estructura de código generalizada capaz de entrenar y buscar la mejor configuración de hiperparámetros de la MLP para cualquiera sea el conjunto de entradas de la red. Primero con la librería de Scikit-Learn se dividen los datos en variables de entrada y salida. Luego, se dividen en dos conjuntos, uno para entrenamiento de la red y otro para su validación, siendo el tamaño de este último un 30% del total de los datos. Finalmente, mediante *StandardScaler* se escalan las entradas y salidas para evitar que la diferencia en la magnitud entre las distintas entradas cargue los parámetros a favor de alguna de ellas.

Ya con los datos procesados, se programa el proceso de entrenamiento y determinación de hiperparámetros de la red con la librería de Tensorflow. La determinación de hiperparámetros se realiza mediante una búsqueda aleatoria, variando los hiperparámetros que se muestran en la Tabla 4.1 en 30 combinaciones distintas.

Tabla 4.1 Conjunto de hiperparámetros para la búsqueda aleatoria.

Hiperparámetro	Valores
Capas Ocultas	$N \in \{x \in \mathbb{N} 1 \leq x \leq 8\}$
Neuronas	$32N \in \{x \in \mathbb{N} 1 \leq x \leq 4\}$
Función de Activación	relu; tanh; elu; swish
Ratio de Aprendizaje	$10^{-N} \in \{x \in \mathbb{N} 2 \leq x \leq 4\}$

Para el proceso de entrenamiento de cada combinación de configuración, incluida la combinación de configuración finalmente seleccionada, se utilizan 500 épocas de entrenamiento, con un comando de detención temprana en caso de que la función de costos no disminuya en 15 épocas consecutivas. La función de costos utilizada para la optimización de los parámetros corresponde al MSE, el cual es el estándar dentro del entrenamiento de ANN.

4.1 Análisis de complejidad del sistema para una red neuronal artificial

Como primer acercamiento, se busca determinar qué componente del sistema afecta en mayor medida la capacidad de predicción de la red. Para ello, se compara la capacidad de predicción de cuatro ANN luego de ser entrenadas cada una con una simulación cada vez más compleja de un satélite en órbita. Como se menciona en la Sección 3.4, existen 2 términos importantes dentro de la dinámica de un satélite: las perturbaciones y el momento angular del satélite, siendo la no-linealidad de esta último, introducida por el efecto giroscópico y las RW, la más compleja. Al expandir la ecuación (7) se puede observar la no-linealidad del sistema y los términos involucrados

$$\begin{aligned} I_{xx} \frac{d\omega_x}{dt} + (I_{zz} - I_{yy})\omega_y\omega_z + \omega_y(J_{zz}\omega_{rw,z}) - \omega_z(J_{yy}\omega_{rw,y}) &= M_{ambiental,x} \\ I_{yy} \frac{d\omega_y}{dt} + (I_{xx} - I_{zz})\omega_x\omega_z + \omega_z(J_{xx}\omega_{rw,x}) - \omega_x(J_{zz}\omega_{rw,z}) &= M_{ambiental,y} \\ I_{zz} \frac{d\omega_z}{dt} + (I_{yy} - I_{xx})\omega_x\omega_y + \omega_x(J_{yy}\omega_{rw,y}) - \omega_y(J_{xx}\omega_{rw,x}) &= M_{ambiental,z}. \end{aligned}$$

Por otro lado, la dinámica aislada de cada RW establece una relación directa entre la aceleración angular de la rueda (relativa al cuerpo), la aceleración angular del satélite y el torque de control (M_{ctrl}) aplicado por el motor:

$$\begin{aligned} \frac{d\omega_{rw,x}}{dt} &= \frac{M_{ctrl,x}}{J_{xx}} - \frac{d\omega_x}{dt} \\ \frac{d\omega_{rw,y}}{dt} &= \frac{M_{ctrl,y}}{J_{yy}} - \frac{d\omega_y}{dt} \\ \frac{d\omega_{rw,z}}{dt} &= \frac{M_{ctrl,z}}{J_{zz}} - \frac{d\omega_z}{dt}. \end{aligned}$$

Para estudiar como influyen cada uno de estos términos en el desempeño de la ANN, se desarrollan cuatro escenarios de simulación con las características presentes en la Tabla 4.2, las cuales aumentan gradualmente la complejidad del sistema.

Tabla 4.2 Arquitecturas de Simulación.

Escenario	Perturbaciones Ambientales	Método de Control
Básico	Inactivas	Torque Directo
Intermedio	Activas	Torque Directo
Avanzado	Inactivas	RW
Alta Fidelidad	Activas	RW

Con estos cuatro escenarios de simulación se generan dos simulaciones de cada escenario, uno para entrenamiento y otro para evaluación. Así, se obtienen cuatro ANN distintas, cada

una asociada a una arquitectura de simulación. Para comparar la capacidad de predicción de las cuatro ANN la Figura 4.1 muestra los gráficos de correlación entre el resultado real y predicho para $\Delta\omega_{x,sat}$ de cada una de estas redes. Se presenta unicamente el desempeño de esta componente por ser representativo del comportamiento de las demás ($\{\omega_{y,sat}; \omega_{z,sat}\}$); los gráficos de correlación de cada componente se encuentran en el ANEXO B.

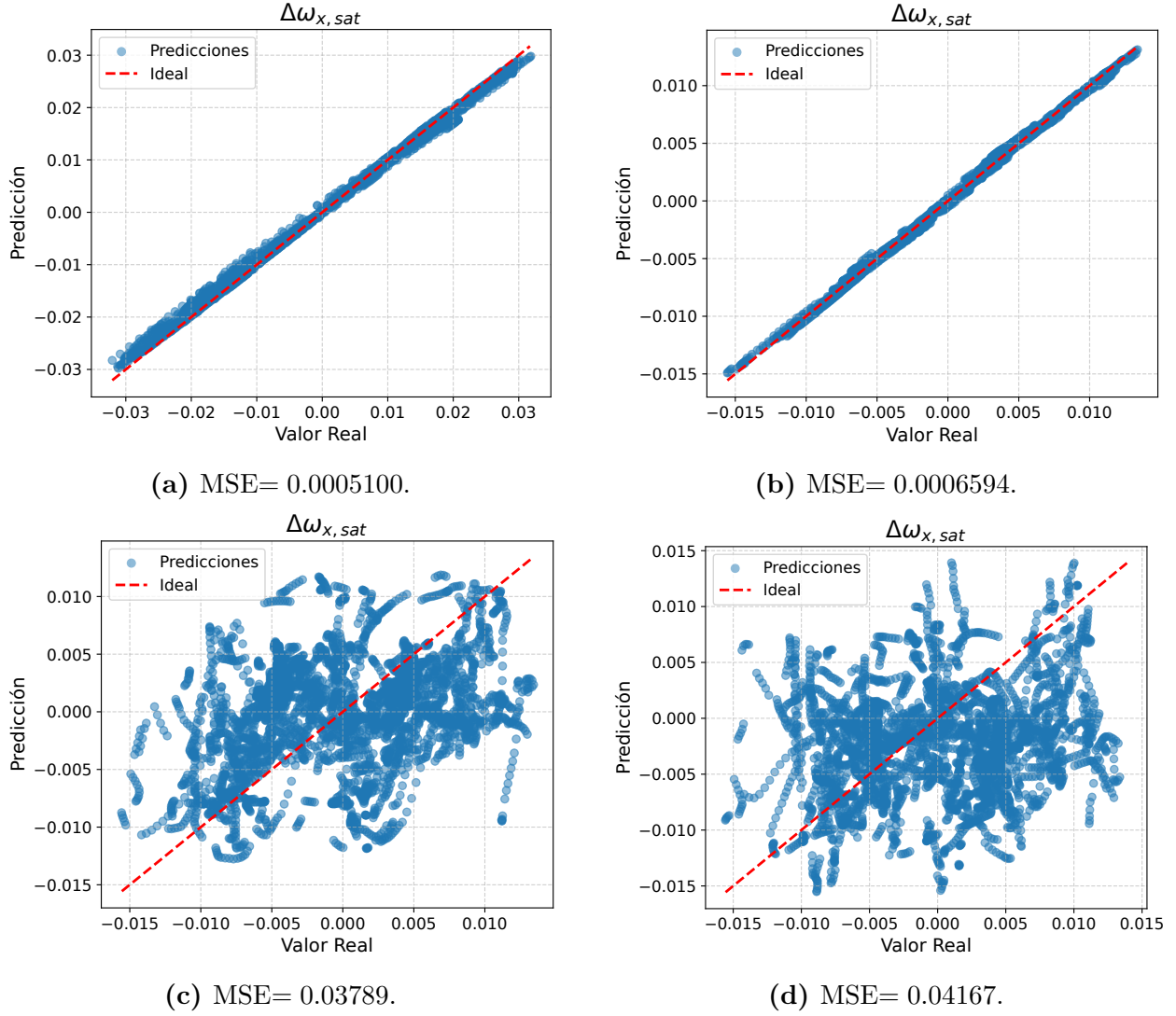


Figura 4.1 Gráficos de correlación entre los valores reales y las predicciones de $\Delta\omega_{x,sat}$ [rad/s] de las diferentes ANN. Figura (a) Escenario Básico; Figura (b) Escenario Intermedio; Figura (c) Escenario Avanzado; Figura (d) Escenario de Alta Fidelidad.

Al comparar el desempeño de cada ANN se destaca el aumento en la dispersión de los puntos entre las Figura 4.1(b) y Figura 4.1(c). Este aumento coincide con la implementación de las RW y su dinámica interna, la cual aporta en la complejización del sistema. Por otro lado, se puede observar como la inclusión de las RW degrada el rendimiento del modelo, incrementando el MSE en casi 2 ordenes de magnitud (de 0.0006 a 0.037). En contraste, la adición de perturbaciones ambientales solo representa incrementos marginales del error. Estos resultados evidencian la dificultad de la red para aproximar la dinámica interna de las RW, siendo un punto crítico que abordar en la mejora del desempeño de las redes.

4.2 Determinación de entradas

A continuación se procede a analizar el impacto de cada variable extraída de las simulaciones en el mejoramiento de la capacidad de predicción de una ANN. Para ello, y tomando como entradas básicas las utilizadas en la Sección anterior ($\{\omega_{sat}, u_{ctrl}\}$), se entrena con una simulación seis diferentes ANN, las cuales se diferencian por sus variables de entradas como se muestra en la Tabla 4.3. Las redes se entrenan con solamente una simulación, para reducir los tiempos de entrenamiento, ya que en esta parte solo se espera comparar el aporte que tiene cada conjunto de entradas en la capacidad de predicción.

Tabla 4.3 Conjuntos de entradas para cada ANN.

ANN	Entradas
A	$\{\omega_{sat}; u_{ctrl}\}$
B	$\{\omega_{sat}; u_{ctrl}; q\}$
C	$\{\omega_{sat}; u_{ctrl}; \omega_{rw}\}$
D	$\{\omega_{sat}; u_{ctrl}; r; v\}$
E	$\{\omega_{sat}; u_{ctrl}; I_{sat}; I_{rw}\}$

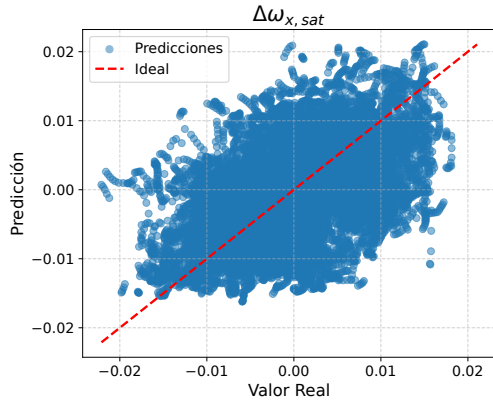
Para comparar el desempeño de cada red, se utiliza el error $MRE^{\Delta\omega_{sat}}$, el cual calcula el promedio porcentual del error del vector (variable objetivo) $\Delta\omega_{sat}$. Esta métrica de error, se calcula de igual forma para todas las redes, al evaluar su capacidad de predicción frente a seis simulaciones que no son utilizados para ningún entrenamiento ni búsqueda de hiperparámetros, permitiendo además evaluar su capacidad de generalización y evitar el sobreajuste.

En la Figura 4.2 se presentan los gráficos de correlación entre el resultado real y el predicho para $\Delta\omega_{x,sat}$ de cada una de las ANN, junto con los valores de $MRE^{\Delta\omega_{sat}}$ asociado a cada una de ella; para ver los gráficos de correlación de cada componente, revisar ANEXO C. Se evidencian dos conjuntos de entradas que mejoran el desempeño de las ANN con respecto a la ANN básica A, la C y D. La ANN C, la cual entrega información sobre el estado de las RW, es la que mayor impacto tiene sobre el desempeño, reduciendo el error en un factor de tres. Esto, concuerda con el análisis anterior, donde se evidencia que las RW son las que mayor impacto tienen en la calidad de predicción de las redes. Por otro lado, al entregarle cierta información acerca de las perturbaciones, también se logra reducir el error, pero en menor medida con respecto a la arquitectura C. Esto igualmente coincide con lo visto en la Sección 4.1, donde se evidencia que el impacto que tienen las perturbaciones es menor al de las RW.

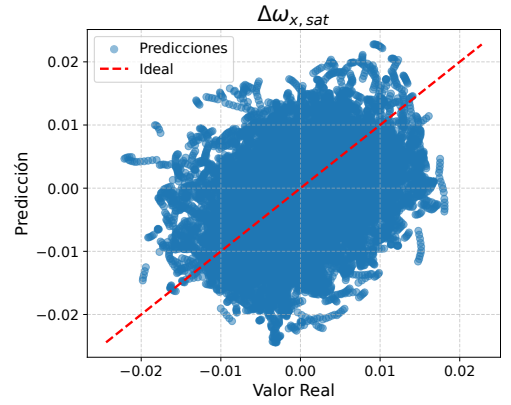
Un aspecto importante a destacar, es el empeoramiento que causa agregar los cuaterniones como entrada, visible en el rendimiento de la ANN B. La actitud del satélite, es crucial para estimar todas las perturbaciones que lo afectan, sin embargo, este no tiene un impacto positivo en la reducción del error como en el caso de la red D. Esto se puede deber a las restricciones y, el espacio en el que se encuentran los cuaterniones; como se mencionó anteriormente, estos cuaterniones deben mantener una norma igual 1, pero la red al realizar cálculos internos puede afectar esta norma y obtener cuaterniones sin un sentido físico, combinado con esto se tiene el hecho de que los cuaterniones corresponden a un espacio de 4 dimensiones donde la combinación lineal de dos de ellos no resulta en un cuaternion válido con el que se puedan realizar predicciones.

Finalmente, con respecto a la ANN $\mathbb{E}$, se aprecia que esta mantiene un error similar al de la ANN $\mathbb{A}$. Esto ocurre por el hecho de que, aunque las inercias sean una pieza clave en la propagación del estado del satélite, para una ANN tener una entrada que no cambia en ninguna iteración se traduce como una variable sin efectos en el sistema, lo cual no le ayuda a relacionar entradas con salidas.

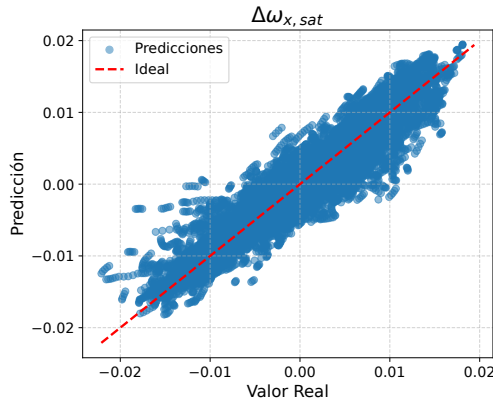
Como las redes $\mathbb{C}$ y $\mathbb{D}$ son las únicas que muestran mejoras con respecto a $\mathbb{A}$, estas se seleccionan para continuar con el estudio de reducción de error de predicción.



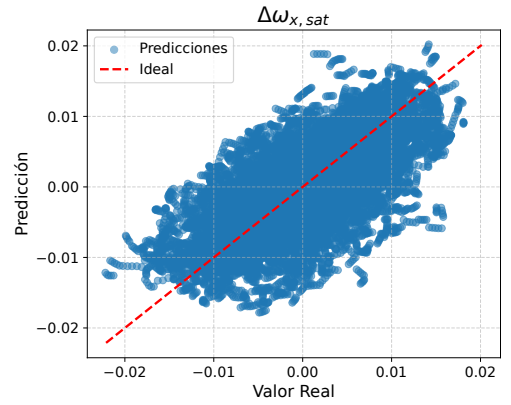
(a) ANN $\mathbb{A}$: $\text{MRE}^{\Delta\omega_{sat}} = 623\%$.



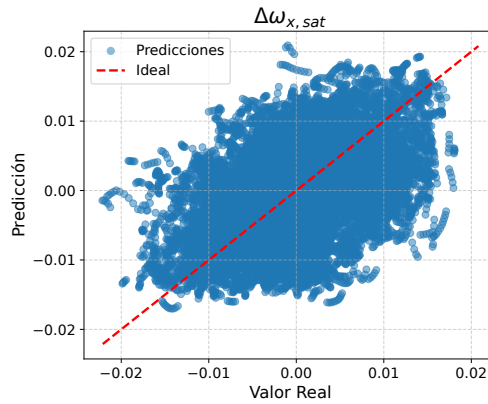
(b) ANN $\mathbb{B}$: $\text{MRE}^{\Delta\omega_{sat}} = 745\%$.



(c) ANN $\mathbb{C}$: $\text{MRE}^{\Delta\omega_{sat}} = 191\%$.



(d) ANN $\mathbb{D}$: $\text{MRE}^{\Delta\omega_{sat}} = 512\%$.



(e) ANN $\mathbb{E}$: $\text{MRE}^{\Delta\omega_{sat}} = 652\%$.

Figura 4.2 Gráficos de correlación entre los valores reales y las predicciones de $\Delta\omega_{x,sat}$ de las diferentes ANN presentadas en la Tabla 4.3.

4.3 Entrenamiento con entradas de pasos anteriores

La estimación del siguiente estado de un sistema, no solo se puede obtener mediante el estado actual de este, sino que también se puede estimar en base a estados anteriores del sistema. La ecuación (8) muestra como calcular el estado de un sistema en un instante $t + 1$ mediante el instante t y su variación. Esto también se puede aplicar para el cálculo de instantes anteriores

$$x_{t-1} = x_{t-2} + f(x_{t-2}, u_{t-2}),$$

y así se podría obtener el siguiente instante de tiempo en mediante información de los estados anteriores

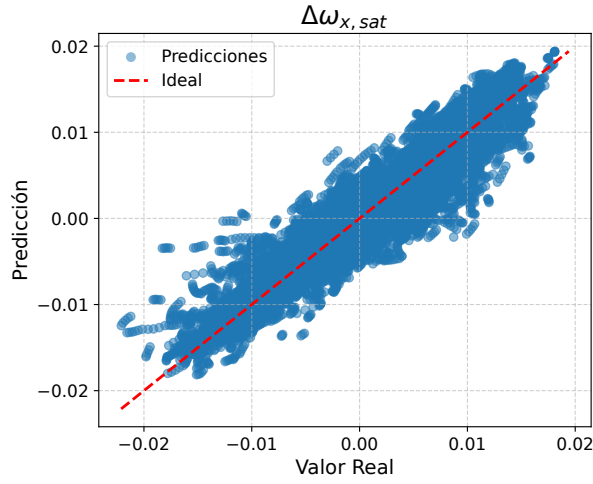
$$x_{t+1} = x_{t-2} + f(x_{t-2}, u_{t-2}) + f(x_{t-1}, u_{t-1}) + f(x_t, u_t).$$

En base a lo anterior se propone entregar como nuevas entradas al sistema, las variables seleccionadas anteriormente junto con sus valores en N instantes de tiempo anteriores. Esto se realiza para las redes $\mathbb{C}$ y $\mathbb{D}$, y para una nueva arquitectura correspondiente a la combinación de entradas de las mismas. Para el entrenamiento de las redes se opta por la configuración con $N = 3$ pasos anteriores, ya que tras realizar pruebas con $N = \{3, 10, 20\}$ se evidencia un estancamiento en la mejora en la capacidad de predicción luego de los 3 pasos hacia atrás. Así, las nuevas arquitecturas se presentan en la Tabla 4.4.

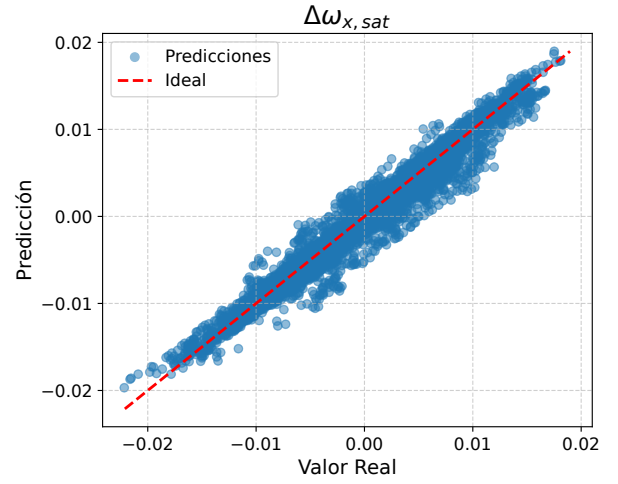
Tabla 4.4 Conjunto de entradas para cada ANN.

ANN	Entradas
$\mathbb{C}$	$\{\omega_{sat}; u_{crl}; \omega_{rw}\}$
$\mathbb{C}_2$	$\{\omega_{sat}; u_{crl}; \omega_{rw}\}_{k-3:k}$
$\mathbb{D}$	$\{\omega_{sat}; u_{crl}; r; v\}$
$\mathbb{D}_2$	$\{\omega_{sat}; u_{crl}; r; v\}_{k-3:k}$
$\mathbb{F}$	$\{\omega_{sat}; u_{crl}; \omega_{rw}; r; v\}$
$\mathbb{F}_2$	$\{\omega_{sat}; u_{crl}; \omega_{rw}; r; v\}_{k-3:k}$

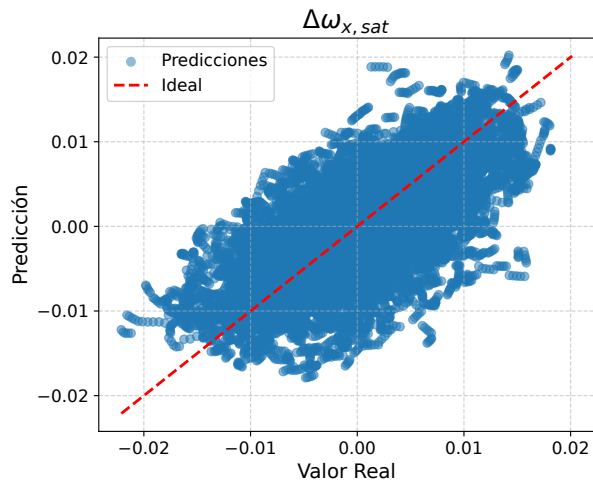
Al evaluar su capacidad de predicción, se obtienen los gráficos de correlación y $MRE^{\Delta\omega_{sat}}$ visibles en la Figura 4.3; para ver los gráficos de correlación de cada componente, revisar ANEXO D. Se observa que en dos de los tres casos en que se entregaron los pasos de tiempo anteriores, se logra reducir el error de predicción. De este proceso de discriminación se seleccionan las 4 mejores redes: $\mathbb{C}$, $\mathbb{C}_2$, $\mathbb{F}$ e $\mathbb{F}_2$, para proceder a su evaluación final.



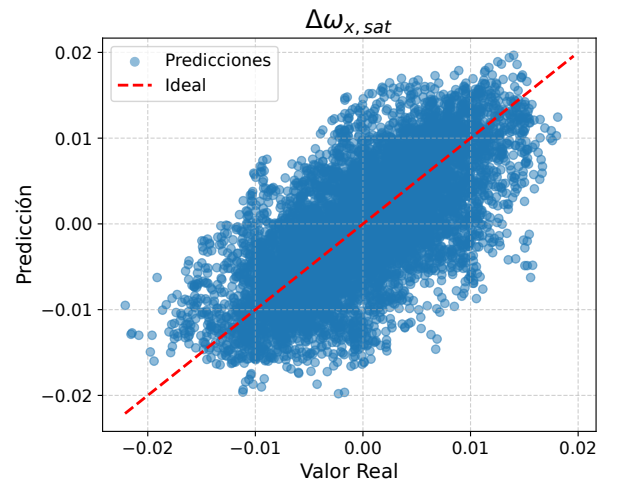
(a) ANN C: $\text{MRE}^{\Delta\omega_{sat}} = 191\%$.



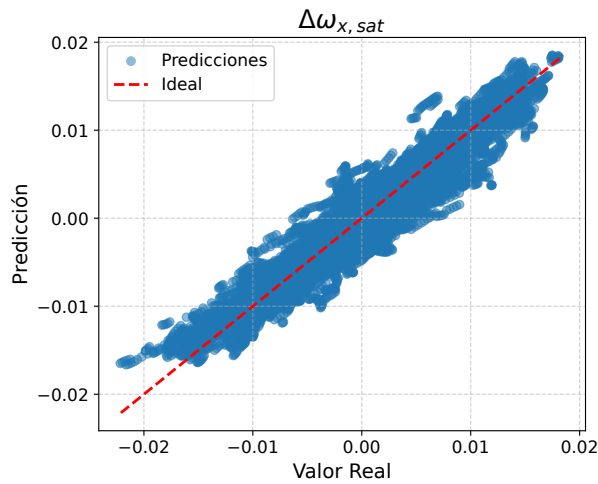
(b) ANN C_2 : $\text{MRE}^{\Delta\omega_{sat}} = 101\%$.



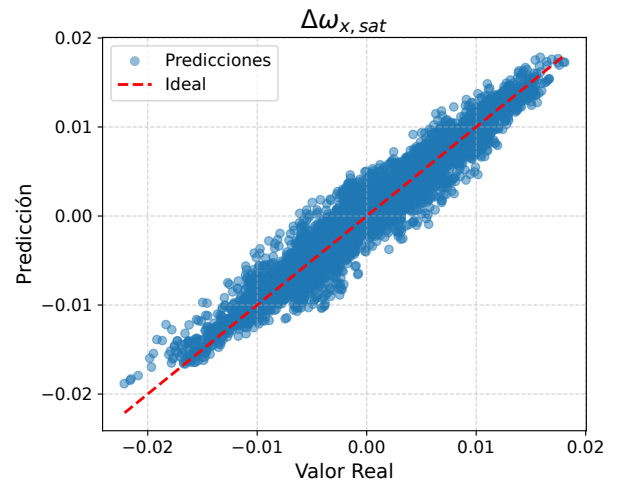
(c) ANN D: $\text{MRE}^{\Delta\omega_{sat}} = 512\%$.



(d) ANN D_2 : $\text{MRE}^{\Delta\omega_{sat}} = 592\%$.



(e) ANN F: $\text{MRE}^{\Delta\omega_{sat}} = 169\%$.



(f) ANN F_2 : $\text{MRE}^{\Delta\omega_{sat}} = 117\%$.

Figura 4.3 Gráficos de correlación entre los valores reales y las predicciones de $\Delta\omega_{x,sat}$ de las diferentes ANN.

4.4 Definición de métricas de desempeño basadas en requerimientos de control

Dentro del desarrollo de algoritmos de control, los modelos dinámicos son de vital importancia; ya sea para ajustar los parámetros de un controlador o como parte misma de su arquitectura, la calidad y objetivo del modelo son clave para desarrollar un controlador óptimo. En esta sección se destacan tres controladores de actitud que se han propuesto desde el año 2022 hasta la actualidad, y como se traducen sus requerimientos funcionales en métricas de desempeños evaluables para los modelos dinámicos basados en ANN.

En 2023, con la motivación de superar el desempeño de los controladores Proporcional-Integral-Derivativo (PID) para satélites y reducir el consumo de energía, el autor Alansaari [27] propuso utilizar un controlador Proporcional-Derivativo (PD) de estructura variable. Este nuevo controlador se basa en alternar entre dos conjuntos de ganancias para un controlador PD según la posición en el plano de fase del estado del satélite. Para sintonizar las ganancias de los controladores, se usó un modelo algebraico simplificado del funcionamiento de un satélite rígido, además de una aproximación de la dinámica de los actuadores, para aumentar la fidelidad del modelo. Al comparar el desempeño de un controlador PD tradicional, con el del nuevo controlador, se obtuvo una mejora significativa en el rendimiento transitorio y en el consumo de energía, logrando el objetivo propuesto.

Otro trabajo que se debe destacar es el de Esit [28], el cual propone aprovechar la capacidad de los controladores MPC para demostrar que se puede estabilizar un satélite usando solo magnetorques, lo cual reduce los costos del satélite. Los controladores MPC tienen integrado un modelo dinámico del sistema, el cual utilizan para predecir la respuesta que tendrá y elegir el camino óptimo para llegar al objetivo. En este caso, Esit utiliza un modelo de espacio de estados linealizado y variante en el tiempo, que además de tener en cuenta el campo magnético de la Tierra, incluye el gradiente de gravedad y las limitaciones físicas de los magnetorques. Los resultados demostraron la viabilidad de usar solo magnetorques para estabilizar el satélite aún cuando se partía de un estado de rotación descontrolada.

Finalmente, en 2025 el autor Huang [29] propone un nuevo controlador MPC modificado con el objetivo de aumentar su precisión, a la vez que reducir su carga computacional y consumo de energía. Basado en un modelo de espacio estados linealizado, Huang implementa un mecanismo de autoatención, el cual determina el momento en el que se debe calcular el control basado en el estado actual, y un horizonte de predicción, el cual ajusta de manera dinámica los pasos de tiempo a predecir. Así, se logra reducir la carga computacional en un 74.8% y evitar cálculos innecesarios.

Al igual que en los trabajos presentados, la mayoría de los controladores propuestos se basan en modificar algoritmos de control PID y MPC para reducir el costo computacional y de energía, a la vez que mantener o aumentar su precisión. Sin embargo, se encuentran ligados al modelo dinámico que ocupen para diseñar el controlador, en el caso del PID, o integren dentro de su arquitectura, como en los MPC. Estos modelos, suelen estar simplificados y en muchos casos no modelar todas las perturbaciones externas ni dinámica de los actuadores en ellos. Por lo anterior, se proponen los criterios de evaluación para los modelos basados en ANN, presentados en la Tabla 4.5.

Como se puede observar en la Tabla 4.5 a cada criterio de desempeño, se le asigna una métrica cuantitativa para la evaluación directa de cada modelo. Para el criterio de exactitud se sigue utilizando la métrica de MRE^Y para comparar el desempeño todos los modelos. Sin embargo, para el criterio de exactitud en N pasos de tiempo se utiliza la métrica de $MRE_{v,N}^Y$.

Así se puede comparar el error en predicción recurrente para cada modelo.

En el caso de los criterios de tiempo de predicción y tamaño de la red se utilizan métricas las cuales sean independientes del hardware utilizado y que posibiliten la comparación con cualquier otro tipo de arquitectura de ANN. Por lo anterior se opta por utilizar la cantidad de Operaciones de punto flotante (Flop) que debe hacer la ANN para realizar una predicción, y para el tamaño de la red se mide en cantidad de parámetros entrenables, los cuales están directamente ligados tanto al tamaño como al tipo de red, permitiendo incluso realizar comparaciones entre distintos tipos de ANN. Para obtener los parámetros entrenables de la red, se utiliza la función *summary()* de Tensorflow la cual se basa en la ecuación (9); para la cantidad de FLOPs se utiliza la ecuación (10), presentada por [30]

$$\text{Parámetros Entrenables} = N_{\text{entradas}} \cdot N_{\text{salidas}} + N_{\text{salidas}} \quad (9)$$

$$\text{Flop} = 2 \cdot N_{\text{entradas}} \cdot N_{\text{salidas}} \quad (10)$$

donde para cada capa N_{entradas} es el número de neuronas de entrada y N_{salidas} la cantidad de neuronas de salida.

Por último, también se calculan los errores de exactitud asociados a la predicción de los cuaterniones. Dado que la red se solo utiliza para predecir y construir los estados de velocidad angular, es importante analizar como los errores asociados a ello se propagan luego en la determinación de la actitud. Para ello, mediante un integrador numérico se utilizan las velocidades angulares construidas en base a la predicción de la red para propagar los cuaterniones y calcular sus errores MRE^Y y $\text{MRE}_{v,N}^Y$.

Tabla 4.5 Métricas de desempeño para evaluación de modelos en base a ANN.

Criterio	Métrica	Justificación
Exactitud	MRE^Y	En el caso de los controladores PID, la exactitud y fidelidad son clave para una buena sintonización de parámetros y posterior desempeño del controlador
Exactitud en N pasos de tiempo	$\text{MRE}_{v,N}^Y$	Los controladores MPC utilizan los modelos para predecir la respuesta del sistema y determinar el mejor camino hacia el equilibrio, por lo que la red debe mantener un error bajo aún pasados N pasos de tiempo
Tiempo de predicción	FLOPs	En los controladores MPC se toman decisiones en el instante tras predecir el comportamiento del modelo, por lo que el tiempo de convergencia del modelo es crucial para el desempeño del controlador
Tamaño de la red	Parámetros entrenables	Una de las principales motivaciones era reducir el costo computacional de los controladores, si la red es demasiado pesada y se ocupa un controlador que deba incluir el modelo, se aumentará el costo computacional asociado a implementarlo

5 Resultados

En esta sección se presentan las cuatro ANN finales obtenidas del entrenamiento de las cuatro mejores redes seleccionadas en la Sección 4.3. Con el fin de llevar registro de la evolución del desempeño de predicción, y para evitar gastar recursos en caso de que una red demostrara empeoramiento, estas fueron entrenadas de manera progresiva con: 5, 50, 500 y, por limitaciones de hardware, 2000 de las 2744 simulaciones seleccionadas al azar. También se presentan los resultados de las redes al ser evaluadas con las métricas de desempeño determinadas en la Sección 4.4. Los gráficos que se muestran corresponden al desempeño en la predicción de una componente del vector ω_{sat} de cada red, y como este desempeño influye luego en el calculo y propagación del cuaternión. Es importante estudiar el desempeño en la predicción de los cuaterniones, ya que los algoritmos de control se diseñan para controlar la orientación del satélite. Para ver el desempeño en cada componente de los vectores revisar los ANEXOS E, F, G, H e I.

5.1 Errores de exactitud

En la columna izquierda de la Figura 5.1 se presentan los gráficos de correlación para la variable $\Delta\omega_{x,sat}$, mientras que en la columna derecha los gráficos de correlación para la componente q_0 de los cuaterniones. Además, bajo cada gráfico se presenta el resultado numérico del error $MRE^{\Delta\omega_{sat}}$ y MRE^q que cuantifica el error en la predicción de los cuaterniones.

Con respecto al error $MRE^{\Delta\omega_{sat}}$, se puede observar que la ANN $\mathbb{C}$ es la que tiene menor error con un valor de 2.14%, seguida de su versión entrenada con pasos anteriores (ANN $\mathbb{C}_2$) que tiene un error de 2.86%. Lo mismo ocurre entre las ANN $\mathbb{F}$ e $\mathbb{F}_2$, donde la red entrenada sin pasos de tiempo anteriores es la que presenta menor error de las 2, pero su otra versión se encuentra solo a 5 centésimas de diferencia. Para el error MRE^q se observa que las redes entrenadas con pasos anteriores son las que presentan el menor error con un valor de 0.14%, mientras que las otras redes se encuentran 3 centésimas por encima.

Por otro lado, en la Figura 5.2 se presentan la evolución temporal de velocidad angular del satélite en x , $\omega_{x,sat}$ y de la componente q_0 del cuaternion, para un espacio de simulación acotado entre los 150s y 170s; esto para las variables predichas y reales. Las predicciones recursivas se realizan para ventanas de 1s (10 instantes de tiempo) las cuales se presentan separadas por líneas punteadas azules que indican el inicio de una ventana y el punto desde el cual se empieza a propagar la predicción recursiva en cada caso. Además bajo cada gráfico se presentan los errores asociados a la utilización de la red en la predicción recurrente de la variación de la velocidad angular $MRE_{v,10}^{\Delta\omega_{sat}}$, y al calculo de los cuaterniones en base a dichas predicciones recurrentes $MRE_{v,10}^q$.

En el caso del error $MRE_{v,10}^{\Delta\omega_{sat}}$ se puede observar que las ANN mantienen la misma jerarquía de desempeño que en el caso del error $MRE^{\Delta\omega_{sat}}$, pero con la magnitud de sus errores mayor o igual al de este. En los gráficos se puede observar que la única de las redes que tiende a divergir del valor real corresponde a la ANN $\mathbb{F}_2$, la cual es la que mantiene el $MRE_{v,10}^{\Delta\omega_{sat}}$ de mayor magnitud. También se presenta el error $MRE_{v,10}^{\omega_{sat}}$ que corresponde al calculo del error porcentual promedio de las ventanas, pero del estado completo del satélite.

Para todas las ANN este error es inferior al 0.5% y demuestra como entrenar las redes para predecir la variación del estado tiene un impacto positivo al reducir el error del estado completo, pues se evidencia que para todo $MRE_{v,10}^{\omega_{sat}} < MRE_{v,10}^{\Delta\omega_{sat}}$. Con respecto a $MRE_{v,10}^q$ se observa que al igual que en el caso de $MRE_{v,10}^{\Delta\omega_{sat}}$ sus valores son siempre mayores a los

de MRE^q , siendo el menor de todos el de la ANN $\mathbb{F}$. Por otro lado, al compararlo con el $MRE_{v,10}^{\Delta\omega_{sat}}$ se observa que sus valores son inferiores a este, a pesar de ser una componente propagada en base al mismo.

5.2 Tamaño y tiempo de predicción

En la Tabla 5.1, se puede observar la estructura interna de cada una de las ANN obtenida tras la búsqueda aleatoria de hiperparámetros. La estructura completa de las redes consta de la capa de entrada, con igual número de neuronas que variables de entrada, la cual se conecta con la primera capa oculta. Luego se tiene la sección de capas ocultas de la red, las cuales se muestran en la Tabla 5.1, donde la última capa oculta se conecta a la capa de salida la cual para todas las redes presenta 3 neuronas, por el tamaño del vector $\Delta\omega_{sat}$, y una función de activación lineal.

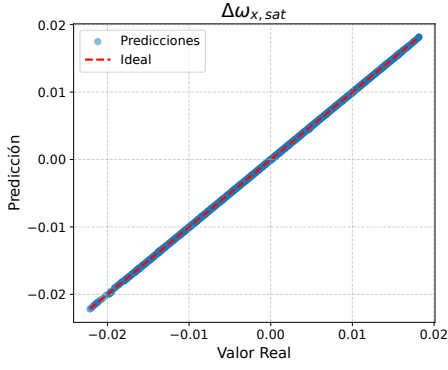
Tabla 5.1 Arquitectura Oculta Final de cada ANN.

ANN	Neuronas por Capa Oculta	Función de Activación
$\mathbb{C}$	{128, 96, 128, 96, 96, 64, 128}	{ <i>Silu, Silu, Tanh, Relu, Silu, Relu, Silu</i> }
$\mathbb{C}_2$	{128, 96, 128, 96, 96, 64, 128}	{ <i>Silu, Silu, Tanh, Relu, Silu, Relu, Silu</i> }
$\mathbb{F}$	{128, 128, 96, 128, 128, 64}	{ <i>Tanh, Elu, Silu, Tanh, Silu, Silu</i> }
$\mathbb{F}_2$	{128, 128, 96, 128, 128, 64}	{ <i>Tanh, Elu, Silu, Tanh, Silu, Silu</i> }

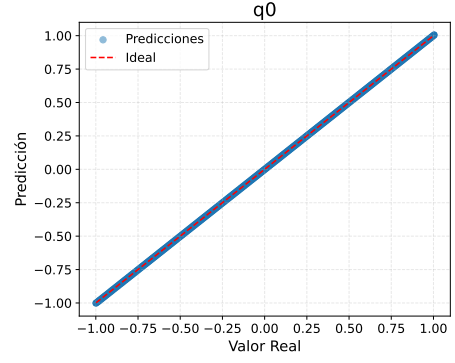
Para conocer la cantidad de FLOPs de cada red, se utiliza en conjunto la ecuación (10) y la información de la estructura de las redes presentadas en la Tabla 5.1, así se calcula la cantidad de Flop correspondientes. Además, a través del comando *summary()* de Tensorflow se extraen la cantidad de parámetros entrenables de cada red. Los resultados se presentan en la Tabla 5.2. Se puede observar que las capas con menor cantidad de parámetros entrenables y Flop corresponden a las arquitecturas de $\mathbb{C}$ y $\mathbb{C}_2$, las cuales son las que presentan el mayor número de capas ocultas.

Tabla 5.2 Parámetros entrenables y cantidad de Flop para cada ANN.

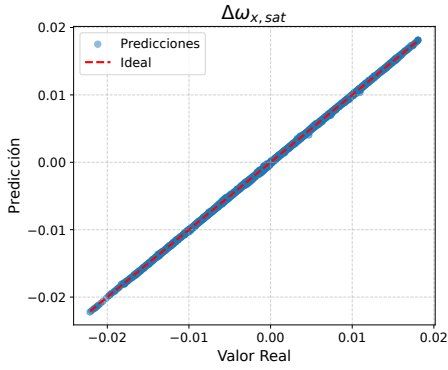
ANN	Parámetros Entrenables	Cantidad de Flop
$\mathbb{C}$	62691	127072
$\mathbb{C}_2$	64995	131680
$\mathbb{F}$	68323	138784
$\mathbb{F}_2$	72163	146464



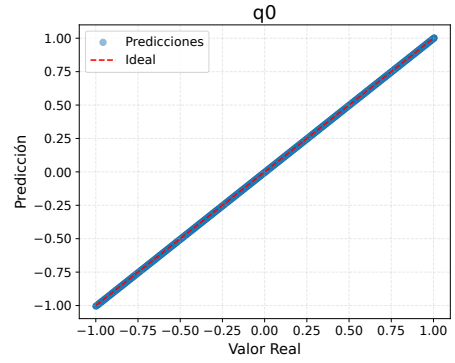
(a) ANN C: $\text{MRE}^{\Delta\omega_{sat}} = 2.14\%$.



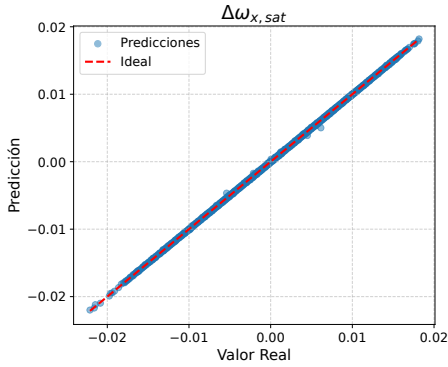
(b) ANN C: $\text{MRE}^q = 0.17\%$.



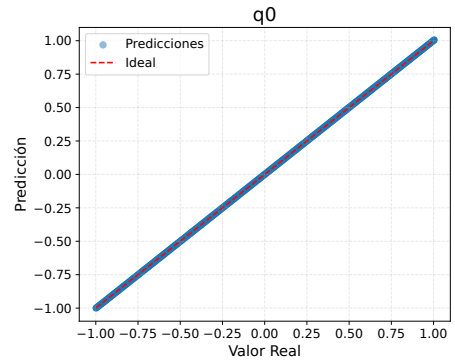
(c) ANN F: $\text{MRE}^{\Delta\omega_{sat}} = 4.55\%$.



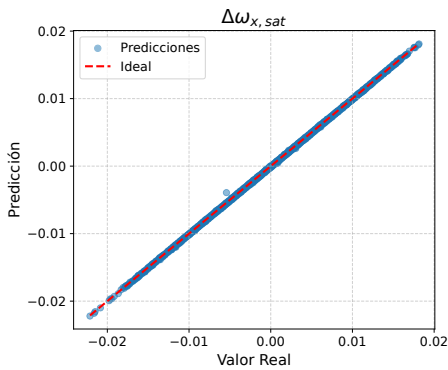
(d) ANN F: $\text{MRE}^q = 0.17\%$.



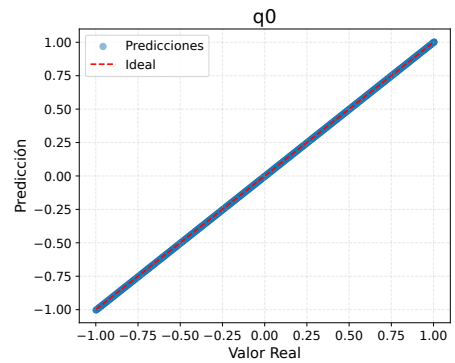
(e) ANN C₂: $\text{MRE}^{\Delta\omega_{sat}} = 2.86\%$.



(f) ANN C₂: $\text{MRE}^q = 0.14\%$.

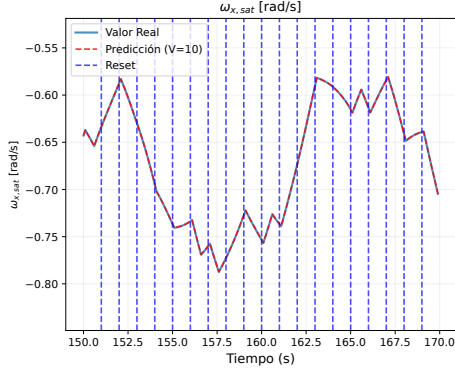


(g) ANN F₂: $\text{MRE}^{\Delta\omega_{sat}} = 4.6\%$

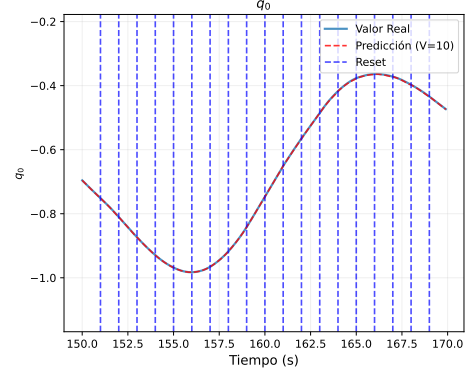


(h) ANN F₂: $\text{MRE}^q = 0.14\%$

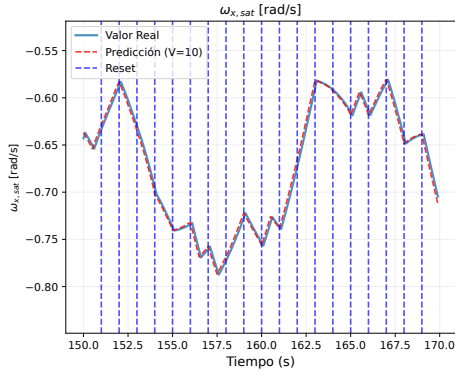
Figura 5.1 Gráficos de correlación entre los valores reales y las predicciones de las variables $\Delta\omega_{x,sat}$ y q_0 de las diferentes ANN presentadas en la Tabla 5.1.



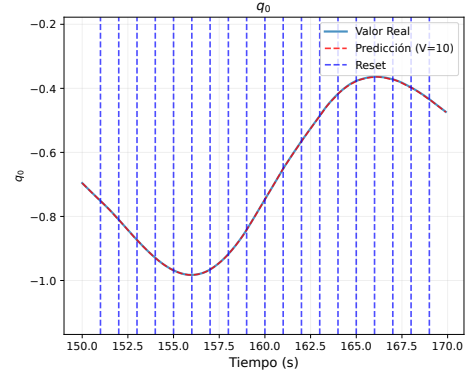
(a) ANN C: $\text{MRE}_{v,10}^{\Delta\omega_{sat}} = 2.14\%$; $\text{MRE}_{v,10}^{\omega_{sat}} = 0.14\%$.



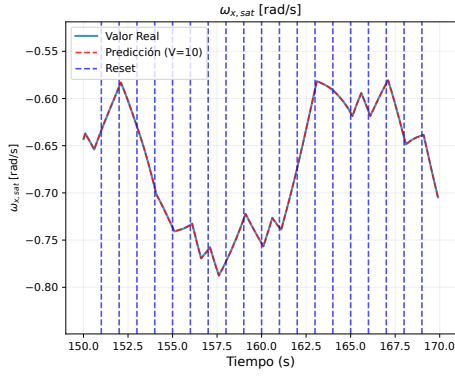
(b) ANN C: $\text{MRE}_{v,10}^q = 1.32\%$.



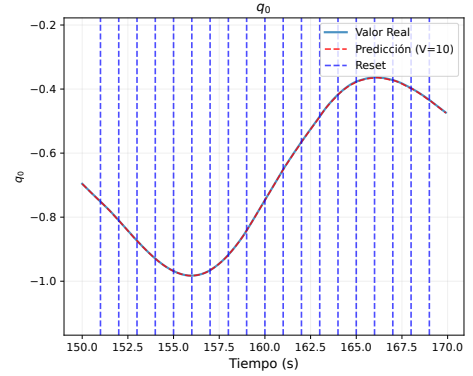
(c) ANN F: $\text{MRE}_{v,10}^{\Delta\omega_{sat}} = 4.6\%$; $\text{MRE}_{v,10}^{\omega_{sat}} = 0.27\%$.



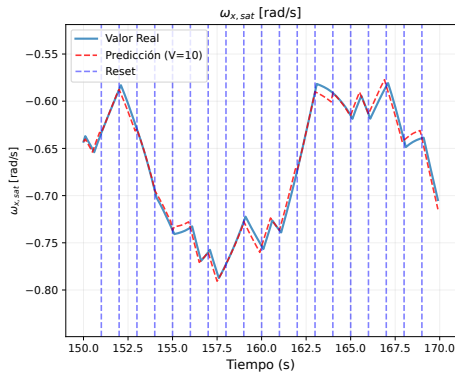
(d) ANN F: $\text{MRE}_{v,10}^q = 1.31\%$.



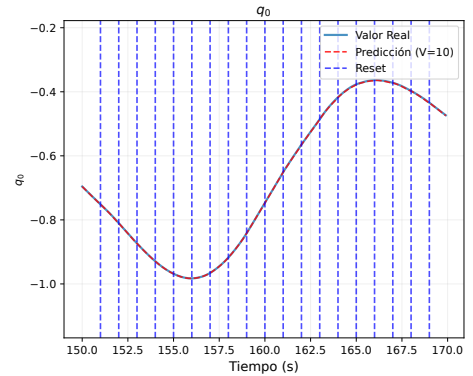
(e) ANN C₂: $\text{MRE}_{v,10}^{\Delta\omega_{sat}} = 3.8\%$; $\text{MRE}_{v,10}^{\omega_{sat}} = 0.26\%$.



(f) ANN C₂: $\text{MRE}_{v,10}^q = 1.45\%$.



(g) ANN F₂: $\text{MRE}_{v,10}^{\Delta\omega_{sat}} = 5.2\%$; $\text{MRE}_{v,10}^{\omega_{sat}} = 0.37\%$.



(h) ANN F₂: $\text{MRE}_{v,10}^q = 1.58\%$.

Figura 5.2 Gráficos de predicción recurrente para ventanas de 1s de las variables reales y predichas de $\omega_{x,sat}$ y q_0 de las diferentes ANN presentadas en la Tabla 5.1.

6 Discusión

En esta sección, se presenta la Tabla 6.1 con un resumen de los resultados de las métricas de evaluación, seguido de un análisis de la misma y comparación con otras redes MLP desarrolladas por [15].

Tabla 6.1 Resumen de resultados.

ANN	Entradas	$\text{MRE}^{\Delta\omega_{sat}}$	$\text{MRE}_{v,10}^{\Delta\omega_{sat}}$	Parámetros Entrenables	FLOPs
$\mathbb{C}$	$\{\omega_{sat}; u_{crl}; \omega_{rw}\}$	2.14%	2.14%	62691	123904
$\mathbb{C}_2$	$\{\omega_{sat}; u_{crl}; \omega_{rw}\}_{k-3:k}$	2.86%	3.80%	64995	128512
$\mathbb{F}$	$\{\omega_{sat}; u_{crl}; \omega_{rw}; r; v\}$	4.55%	4.60%	68323	135296
$\mathbb{F}_2$	$\{\omega_{sat}; u_{crl}; \omega_{rw}; r; v\}_{k-3:k}$	4.60%	5.20%	72163	142976
MLP_{4x16}	$\{\omega_{sat}; u_{crl}; \omega_{rw}; \alpha_{sat}; I_{sat}; I_{rw}\}$	19.67%	22.63%	1822	3456
MLP_{9x64}	$\{\omega_{sat}; u_{crl}; \omega_{rw}; \alpha_{sat}; I_{sat}; I_{rw}\}$	9.85%	10.52%	37214	73216

Como se menciona en la Sección 5, con respecto al error $\text{MRE}^{\Delta\omega_{sat}}$, las redes presentan un patrón donde las ANN de los pares $\{\mathbb{C}, \mathbb{C}_2\}$ y $\{\mathbb{F}, \mathbb{F}_2\}$ que no presentan las entradas de pasos anteriores, son las que tienen el menor error de cada par, y siendo la mejor de las cuatro la ANN $\mathbb{C}$. Es importante destacar este hecho, ya que, si bien se presenta la herramienta de los pasos anteriores como un método para mejorar la capacidad de predicción de las redes, se evidencia que lo que más influye son las distintas entradas (información) que se le entregan a la red y en que se ve cada una involucrada dentro del sistema. Si bien, puede ser una buena orientación revisar las ecuaciones analíticas que rigen o aproximan el comportamiento de un cuerpo para determinar estas entradas, las ANN son ajenas a estas y solo son un arreglo numérico. Este hecho podría explicar porque a pesar de que en la red $\mathbb{F}$ que además de entregarle la información más importante, $\{\omega_{rw}\}$ junto con $\{r, v\}$, no mejora sino que empeora la capacidad de predicción de la red. En adición a lo anterior, se debe destacar la estructura de las simulaciones de las cuales se extrajeron los datos. Si bien, en ellas se varió la condición inicial de ω_{sat} y se dieron diferentes secuencias de acciones de control para excitar y captar la dinámica del problema, ninguna de estas variaciones influía en las perturbaciones presentes en las simulaciones. A pesar de que se ha mencionado que la actitud del satélite juega un papel fundamental en el cálculo de las perturbaciones ambientales, también lo hacen la posición del satélite y la velocidad inercial de este. Como en ninguna simulación se varió la condición inicial de posición del satélite, todas las simulaciones se realizaron para la misma zona orbital, por lo que el comportamiento de las perturbaciones varía solo por la actitud entre las distintas simulaciones. Esto puede haber influido en el peso que tiene $\{r, v\}$ en el mejoramiento de la capacidad de predicción.

Al analizar el error $\text{MRE}_{v,10}^{\Delta\omega_{sat}}$ se destaca el hecho de que las redes mantienen la misma jerarquía de desempeño que en el caso del MRE, pero con magnitudes diferentes; al realizar la comparación, se evidencia que la magnitud de $\text{MRE}_{v,10}^{\Delta\omega_{sat}}$ es siempre mayor y en el mejor de los casos igual al de $\text{MRE}^{\Delta\omega_{sat}}$. Sin embargo, es importante destacar como el hecho de haber entrenado las redes para predecir la variación del estado, influye en el desempeño de predicción del estado completo. Mientras que el error en la predicción recursiva de la variación del estado se encuentra entre 2% y 5% al analizar el error para el estado completo se tiene que sus magnitudes son menores, siendo estas inferiores al 0.5%. La diferencia observada entre las magnitudes de los errores $\text{MRE}^{\Delta\omega_{sat}}$ y $\text{MRE}_{v,10}^{\Delta\omega_{sat}}$ radica en la estrategia

de entrenamiento que se utilizó para las redes. Al entrenar las redes utilizando la función de costos MSE para predecir la variación del estado exclusivamente del siguiente instante, se esta aplicando el método de minimización del Error de Ecuación (EE) [31]. Justamente este enfoque favorece el desempeño, en este caso de las ANN, a corto plazo, pero siendo susceptibles a la acumulación de errores al realizar predicciones recurrentes (o para simulaciones). Al retroalimentar las estimaciones de la red como entradas para propagar de manera recursiva los siguientes estados, los errores iniciales se propagan y amplifican, explicando que las magnitudes de $MRE_{v,10}^{\Delta\omega_{sat}}$ sean siempre mayores a las de $MRE^{\Delta\omega_{sat}}$. Por lo anterior, si se tiene la certeza de que el modelo se requiriera emplear en controladores MPC, se debería contemplar la transición hacia funciones de costo basadas en el Error de Salida (OE). Estas alternativas, aunque exigen algoritmos de entrenamiento más complejos como la retropropagación a través del tiempo, optimizan directamente la respuesta simulada minimizando el error sobre el horizonte completo de la predicción.

Al comparar la magnitud de los errores en la determinación de los cuaterniones con los errores de predicción de $\Delta\omega_{sat}$, se evidencia que estos son siempre menor, a pesar de que el cuaternion se propaga en base a dichas predicciones. Esto ocurre principalmente porque la propagación de los cuaterniones se realiza en base al estado completo de la velocidad angular, donde como se menciona anteriormente, el error es menor que en el caso de la predicción de la variación. Además, al propagar los cuaterniones y aplicar una normalización sobre ellos, se podría estar agregando un filtro frente al error proveniente de las predicciones. Esto se evidencia al observar las Figuras 5.2(g) y 5.2(h), donde a pesar de que diverge la velocidad angular, no lo hace el cuaternion.

Con respecto al tamaño de las redes, se evidencia que la red con menos parámetros entrenables corresponde a la ANN $\mathbb{C}$, que tiene la menor cantidad de entradas. De manera homologa, la red mas grande corresponde a la arquitectura $\mathbb{F}_2$, la cual es la que tiene la mayor cantidad de parámetros de entradas. Sin embargo, el orden de clasificación de las redes no solo se guía por la cantidad de parámetros de entrada, ya que para las arquitecturas en posiciones intermedias esta clasificación ocurre al revés, yendo primero la que tiene un mayor número de entradas. Al observar la arquitectura interna de cada red, visible en la Tabla 5.1, se puede observar que la cantidad de capas ocultas, neuronas y su ordenamiento, es igual para los pares de redes que tienen las mismas variables de entrada y su versión con pasos hacia atrás, siendo lo único que las diferencia la cantidad de neuronas de la capa de entrada, pero este cambio solo provoca un aumento del 4% y 6% en los parámetros respectivamente. Por otro lado, las 2 redes con menor cantidad de parámetros entrenables, corresponden a las redes con mayor número de capas ocultas, al contrario de lo que se esperaría. En base a todo lo anterior y con respecto a la ecuación (9), se puede afirmar que el tamaño de la red no solo se ve afectado por el número de capas y neuronas totales que tenga una red, sino que también de como se ordenan estas capas al interior de ella.

Al analizar los FLOPs, se observa que este presenta un comportamiento similar a los parámetros entrenables. Esto ocurre debido a que las 4 ANN son arquitecturas MLP, donde los FLOPs son aproximadamente el doble de los parámetros entrenables. Esto se puede demostrar al dividir la ecuación (10) por la ecuación (9)

$$P = \frac{2 \cdot N_{entradas} \cdot N_{salidas}}{N_{entradas} \cdot N_{salidas} + N_{salidas}} = \frac{2 \cdot N_{entradas} \cdot N_{salidas}}{N_{salidas}(N_{entradas} + 1)} = \frac{2 \cdot N_{entradas}}{N_{entradas} + 1}$$

de aqui se obtiene una relación P entre FLOPs y parámetros entrenables, donde si se evalúa el rango de valores que puede tomar con las entradas mínimas y máximas, se tiene un rango entre $P_{min} = 1.8$ y $P_{max} = 1.98$, el cual coincide con la relación promedio que se observa

en la Tabla 6.1 de $P = 1.979$. De igual forma es importante conocer los Flop además de los parámetros entrenables, ya que como se mencionó anteriormente, esta relación se cumple solo en el caso de los MLP, por lo que si se quiere comparar el tiempo de computo con arquitecturas diferentes no bastaría solo con ver los parámetros entrenables.

Al evaluar las cuatro métricas en conjunto, se evidencia que la ANN $\mathbb{C}$ es la que cumple mejor con los requisitos para luego ser utilizada como modelo base para el desarrollo de sistemas de control. Esta red presenta el mejor desempeño tanto en predicción del instante siguiente, como en predicción recursiva. Además esto lo logra manteniendo la arquitectura y tiempo de computo mas bajo de las 4. También es importante destacar como estos resultados se obtienen mediante el ingreso de la menor cantidad de variables dentro de las 4 ANN. En la realidad esto se traduce en una ventaja de implementación, al necesitar menos sensores o cálculos de estimación para obtener las entradas, que en comparación de las otras redes.

Finalmente, para estimar la posibilidad de ocupar esta red en un algoritmo de control, se compara con otros dos modelos de MLP, $\text{MLP}_{4 \times 16}$ y $\text{MLP}_{9 \times 64}$, desarrollados por [15] para estimar la dinámica de un satélite como cuerpo rígido y servir como modelo dentro de un controlador MPC. Al comparar la ANN $\mathbb{C}$ con estos modelos, se observa que esta es la que presenta el menor error en $\text{MRE}^{\Delta\omega_{sat}}$ y $\text{MRE}_{v,10}^{\Delta\omega_{sat}}$, pero con un tamaño y tiempo aproximadamente el doble que su red mas exacta. Sin embargo, el autor también presenta otras arquitectura de redes de mayor tamaño, las cuales mantienen un error $\text{MRE}^{\Delta\omega_{sat}}$ por debajo del 1% y son utilizadas luego en un controlador MPC. Esto indica que a pesar del tamaño de la ANN $\mathbb{C}$, esta si se podría utilizar para el desarrollo de un controlador.

7 Conclusiones

En este trabajo se desarrollaron y evaluaron cuatro ANN en base a la arquitectura MLP, que sirven como modelos de predicción de la dinámica de cuerpo rígido de un cubesat-6U en órbita baja. Para el entrenamiento de estos modelos, se implementó una arquitectura de simulación de alta fidelidad en Basilisk, de la cual se extrajeron distintas variables de interés en 2744 simulaciones distintas. Tras la evaluación de los cuatro modelos se determinó que la ANN $\mathbb{C}$ cumple con el objetivo general al presentar los valores mas bajos de las métricas de evaluación presentadas, y al compararse con la literatura demostrar ser apta para implementarse en un controlador MPC.

A partir del proceso de entrenamiento y evaluación de los modelos, se pueden destacar distintas conclusiones con respecto a la selección de las entradas, selección de variable objetivo y tamaño de la red:

- **Determinación de entradas:** Se evidenció que aumentar la cantidad de variables de entradas (información) o incluir información sobre instantes de tiempo anteriores, no siempre mejora la capacidad de predicción de una red. Así, es más importante el proceso de estudio de complejidad del sistema, y que variables contienen mayor información sobre esta, que aumentar la cantidad total de información entregada al modelo. Todo esto se respalda en el hecho de que la ANN $\mathbb{C}$, seleccionada como la mejor y que mantiene la menor cantidad de entradas ($\{\omega_{sat}; u_{ctrl}; \omega_{rw}\}$), es la que logró el mejor desempeño en predicción, con un $MRE^{\Delta\omega_{sat}} = MRE_{v,10}^{\Delta\omega_{sat}}$ de 2.14%.
- **Selección de variable objetivo:** Durante el desarrollo del trabajo, la correcta selección de una variable objetivo permitió no solo un mejor manejo de las ANN, sino que también repercutió en el desempeño de predicción de la red. El optar por predecir el cambio en el estado del satélite ($\Delta\omega_{sat}$), en vez del estado completo (ω_{sat}), le permitió a los parámetros de la red ajustarse mejor. Además, al construir el siguiente estado completo en base a la variación predicha se obtiene un error asociado a ellos menor que el de las variaciones $MRE_{v,10}^{\omega_{sat}} < MRE_{v,10}^{\Delta\omega_{sat}}$, lo cual a su vez permitió propagar los cuaterniones con un error mínimo.
- **Tamaño de la red:** Los resultados mostraron que una red más grande (con respecto a parámetros entrenables) no se relaciona necesariamente con una mejor capacidad de predicción. En los resultados se pudo evidenciar como el orden según el desempeño en predicción de cada red, era el inverso al tamaño de estas, siendo la red $\mathbb{C}$ la con mejor capacidad de predicción y con menor tamaño (62691 parámetros entrenables)

Finalmente, durante el trabajo y proceso de finalización se identificaron diferentes desafíos y limitaciones que deben ser abordadas en un futuro. El primer desafío que se identificó fue la no linealidad aportada por las RW del sistema. En el análisis progresivo del sistema, se evidenció como al agregar las RW el desempeño de la red caía bruscamente al aumentar de un MSE de 0.0006594 a 0.03789. Es importante identificar las partes críticas de un sistema, especialmente si luego se espera trabajar con un modelo flexible, donde pueden haber más componentes que lo complejicen. Una mejora que se propone para futuros trabajos es la variabilidad en las simulaciones; si bien en este caso se variaron condiciones iniciales, se omitió la posición inicial del satélite, la cual pudo haber repercutido negativamente en el aporte que tuvo la información de r y v , además de no permitir evaluar el funcionamiento del modelo para la totalidad de la órbita en la que se encuentra. Por último, una mejora crítica que se debe abordar en el funcionamiento de los modelos se relaciona con la información de

entrada en las predicciones recurrentes. En este trabajo, se asumió que toda la información futura de entrada en las predicciones recurrentes era conocida $(\omega_{rw}; r; v)$, sin embargo en la realidad estas deben ser estimadas o propagadas de alguna forma. Por lo anterior, se propone como mejora desarrollar un método de estimación de estas entradas en paralelo, y estudiar sus efectos en el desempeño de las redes.

Referencias

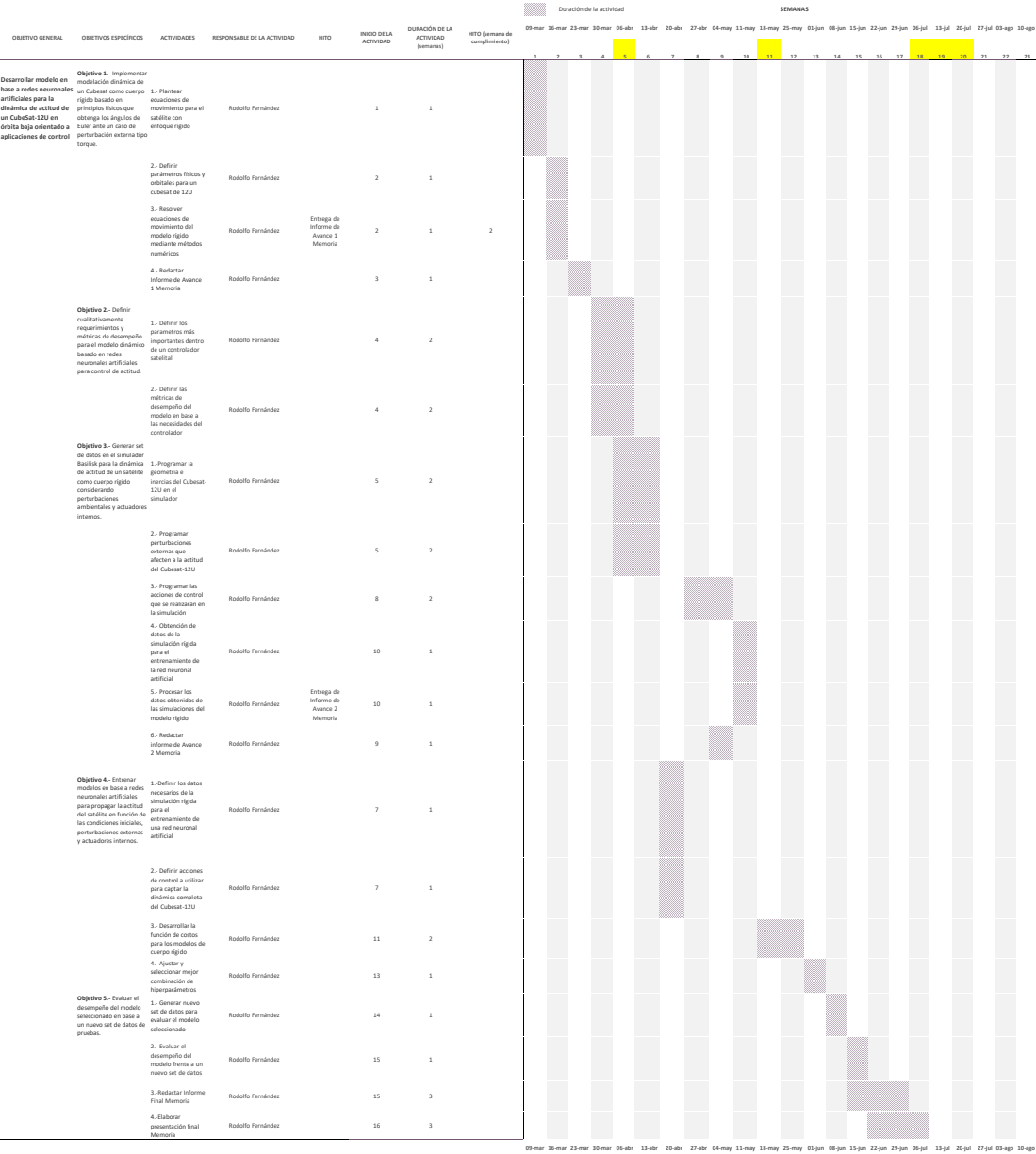
- [1] Fortune Business Insights, “Crecimiento del mercado de satélites medianos y grandes, participación, análisis de la industria y pronóstico regional, 2026-2034,” abril 2026. Última actualización: 6 de abril de 2026. Accedido: 1 de mayo de 2026.
- [2] A. P. Buckley, “Hubble space telescope pointing control system design improvement study results,” *Journal of Guidance, Control, and Dynamics*, vol. 18, no. 2, pp. 194–199, 1995.
- [3] K. L. Bourkland and K.-C. A. Liu, “Verification of the solar dynamics observatory high gain antenna pointing algorithm using flight data,” in *AIAA Guidance, Navigation, and Control Conference*, American Institute of Aeronautics and Astronautics, August 2011.
- [4] Y. Tsuda, O. Mori, R. Funase, H. Sawada, T. Yamamoto, T. Saiki, T. Endo, K. Yonekura, H. Hoshino, and J. Kawaguchi, “Achievement of ikaros — japanese deep space solar sail demonstration mission,” *Acta Astronautica*, vol. 82, pp. 183–188, 2013.
- [5] J. L. Redondo Gutiérrez, “Attitude control of flexible spacecraft: Design, implementation and evaluation of control strategies targeting flexible structures in the space domain, based in an analytical modeling of these structures,” Master’s thesis, Delft University of Technology (TUDelft), January 2019. Student number: 4570332, Project duration: May 1, 2018 – November 30, 2018.
- [6] Y. Yang, Z. Zhang, M. Liu, H. Han, J. Wei, and D. Cao, “Dynamic modeling and analysis for a large flexible spacecraft with antenna trusses and solar wings,” *Applied Mathematical Modelling*, 2025. In Press, Journal Pre-proof.
- [7] J. Park, J. D. Lee, and H. Bang, “Vibration suppression adaptive prescribed performance control for flexible spacecraft with uncertain moment of inertia,” *Acta Astronautica*, vol. 235, pp. 141–154, 2025.
- [8] D. Wang, Y. Wu, C. Yue, and S. Ma, “Fully actuated system-based active disturbance rejection saturated attitude control for flexible spacecraft,” *Journal of the Franklin Institute*, vol. 362, p. 107613, 2025. This research was supported by the National Natural Science Foundation of China under Grant No. 61973153.
- [9] Z. qiang Hu, L. Yang, Y.-S. Wang, L. Wei, and Y. Chen, “Tuned mass damper on spacecraft reaction wheel assembly,” *Applied Acoustics*, vol. 210, p. 109456, 2023.
- [10] S. Lin, Y. Zhao, and W. Ma, “Vibration control system design using active devices for large space structure during attitude maneuver,” *Journal of Vibration Engineering & Technologies*, vol. 13, p. 294, 2025.
- [11] L. Zhang, H. Ren, T. Yuan, and W. Fan, “Dynamic modeling and attitude maneuver control on $SO(3)$ for spacecraft with large flexible appendages,” *Acta Astronautica*, vol. 228, pp. 551–560, 2025.
- [12] T. Slimak, A. Zwölfer, B. Todorov, and D. Rixen, “A machine learning approach to simulate flexible body dynamics,” *Multibody System Dynamics*, vol. 64, pp. 515–541, 2025.
- [13] Z. Chen, F. Renda, A. Le Gall, L. Mocellin, M. Bernabei, T. Dangel, G. Ciuti, M. Cianchetti, and C. Stefanini, “Data-driven methods applied to soft robot mod-

- eling and control: A review,” *Submitted to IEEE*, 2024. Available at: <https://sites.google.com/view/23zcb>.
- [14] C.-W. Fei, H. Li, H. Liu, C. Lu, L. An, and L. Han, “Enhanced network learning model with intelligent operator for the motion reliability evaluation of flexible mechanism,” *Aerospace Science and Technology*, 2020. Article in press.
 - [15] C. Cena, M. Martini, and M. Chiaberge, “Learning satellite attitude dynamics with physics-informed normalising flow,” 2025. arXiv:2508.07841v2.
 - [16] S. Han, H.-S. Choi, J. Choi, J. H. Choi, and J.-G. Kim, “A DNN-based data-driven modeling employing coarse sample data for real-time flexible multibody dynamics simulations,” *Computer Methods in Applied Mechanics and Engineering*, vol. 373, p. 113480, 2021.
 - [17] P. W. Kenneally, S. Piggott, and H. Schaub, “Basilisk: A flexible, scalable and modular astrodynamics simulation framework,” *Journal of Aerospace Information Systems*, vol. 17, pp. 496–507, sep 2020.
 - [18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikitlearn: Machine learning in python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
 - [19] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattemberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015. Software available from tensorflow.org.
 - [20] F. L. Markley and J. L. Crassidis, *Fundamentals of Spacecraft Attitude Determination and Control*, vol. 33 of *Space Technology Library*. New York, NY, USA: Springer Science+Business Media, 2014.
 - [21] H. D. Curtis, *Orbital Mechanics for Engineering Students*. Elsevier Aerospace Engineering Series, Oxford, UK: Butterworth-Heinemann, fourth ed., 2020.
 - [22] D. A. Vallado and W. D. McClain, *Fundamentals of Astrodynamics and Applications*. Space Technology Library, Hawthorne, CA: Microcosm Press, 4th ed., 2013. Publicado conjuntamente por Microcosm Press y Springer.
 - [23] NanoAvionics, “6u cubesat / nanosatellite bus m6p,” 2026. Accedido: 9 de mayo de 2026.
 - [24] S. Haykin, *Neural Networks and Learning Machines*. Pearson Education, 3 ed., 2009.
 - [25] a.i. solutions, Inc., “FreeFlyer.” <https://ai-solutions.com/freelyer-astrodynamic-software/>, 2026. Space mission design, analysis, and operations software.
 - [26] L. Ljung, *System Identification: Theory for the User*. Prentice Hall Information and System Sciences Series, Upper Saddle River, NJ: Prentice Hall, 2nd ed., 1999.

- [27] A. Alansaari, I. Boiko, and S. Swei, “Variable-structure pd controller for satellite attitude: Stability analysis and design for optimal performance,” *Journal of the Franklin Institute*, vol. 360, pp. 9728–9753, 2023.
- [28] M. Esit, H. E. Soken, and C. Hajiyev, “A model predictive control based magnetorquer-only attitude control approach for a small satellite,” *IFAC-PapersOnLine*, vol. 55, no. 22, pp. 79–84, 2022.
- [29] T. Huang, J. Zhang, M. Li, Y. Shen, J. Wu, and H. Wang, “Self-triggered mpc with adaptive prediction horizon for nano-satellite attitude control system,” *Advances in Space Research*, vol. 75, pp. 2251–2270, 2025. Available online 18 October 2024.
- [30] J. Sevilla, P. Villalobos, J. F. Cerón, M. Burtell, L. Heim, A. B. N, and T. Besiroglu, “Estimating training compute of deep learning models,” *arXiv preprint arXiv:2211.02001*, 2022.
- [31] Y. Tomita, A. A. H. Damen, and P. M. J. Van den Hof, “Equation error versus output error methods,” *Ergonomics*, vol. 35, no. 5-6, pp. 551–564, 1992.

A Carta Gantt

CARTA GANTT - Modelado de la dinámica de satélites como cuerpos rígidos mediante redes neuronales artificiales para aplicaciones de control.



B Figuras Primera Discriminación

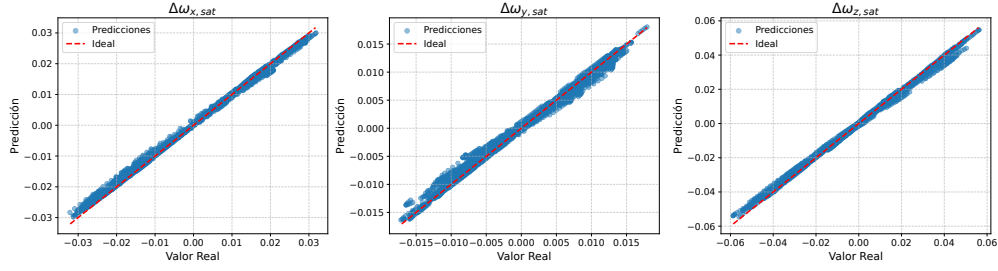


Figura B.1 Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de una red entrenada con simulaciones sin perturbaciones ni ruedas de reacción

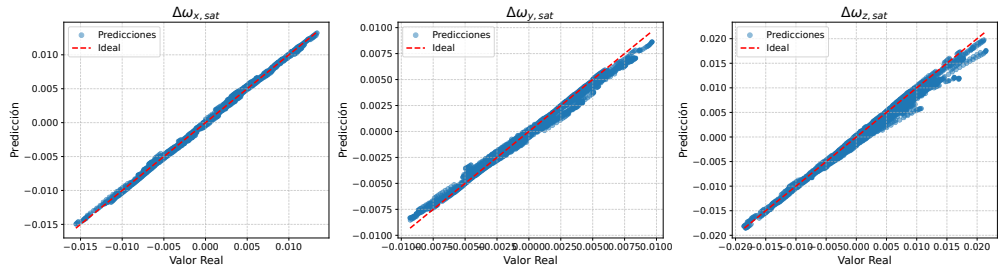


Figura B.2 Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de una red entrenada con simulaciones con perturbaciones y sin ruedas de reacción

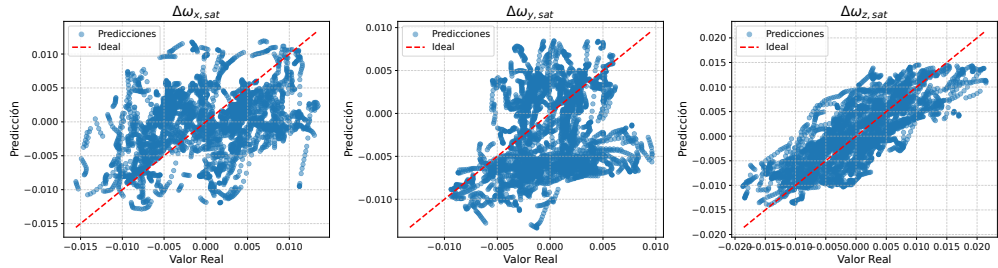


Figura B.3 Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de una red entrenada con simulaciones sin perturbaciones y con ruedas de reacción

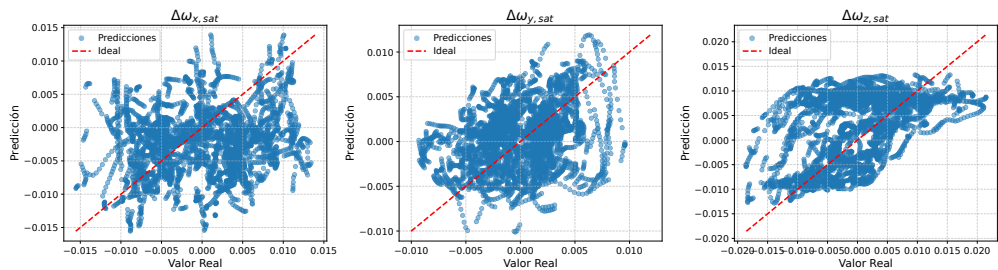


Figura B.4 Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de una red entrenada con simulaciones con perturbaciones y ruedas de reacción

C Figuras Segunda Discriminación

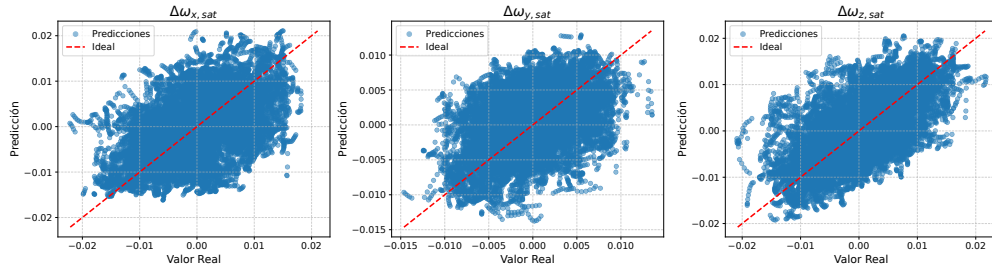


Figura C.1 Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN A

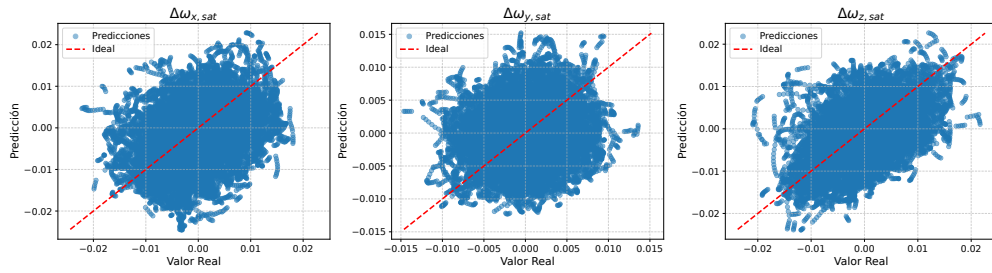


Figura C.2 Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN B

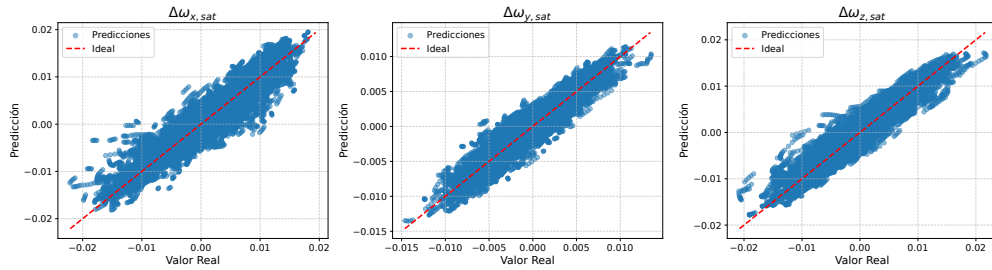


Figura C.3 Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN C

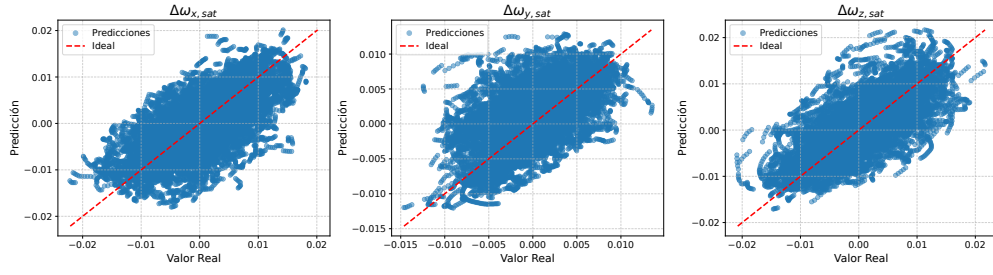


Figura C.4 Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{D}$

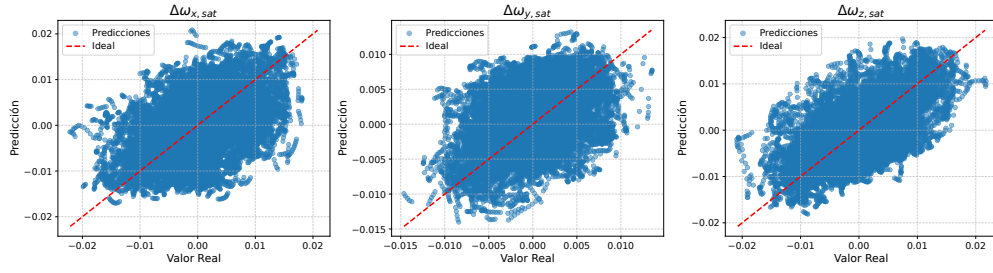


Figura C.5 Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{E}$

D Figuras Tercera Discriminación

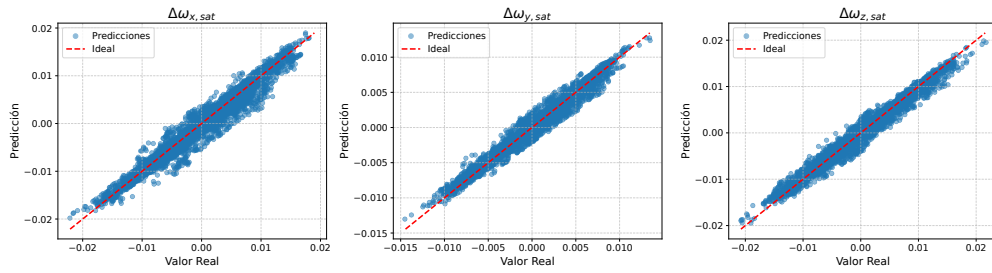


Figura D.1 Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{C}_2$

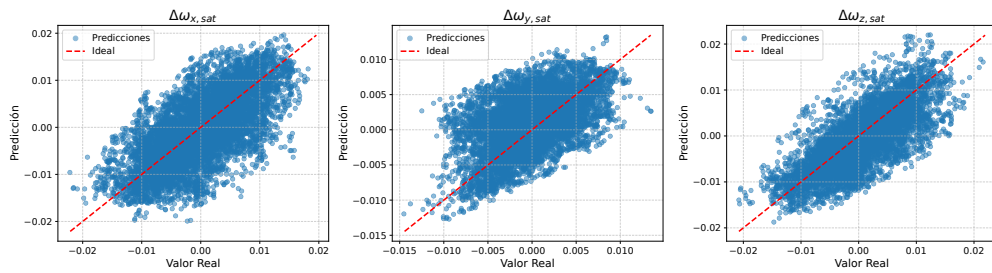


Figura D.2 Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{D}_2$

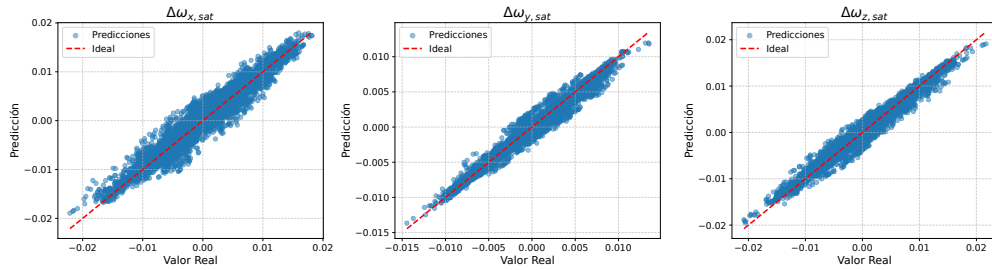


Figura D.3 Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{F}_2$

E Figuras Resultados Velocidades angulares

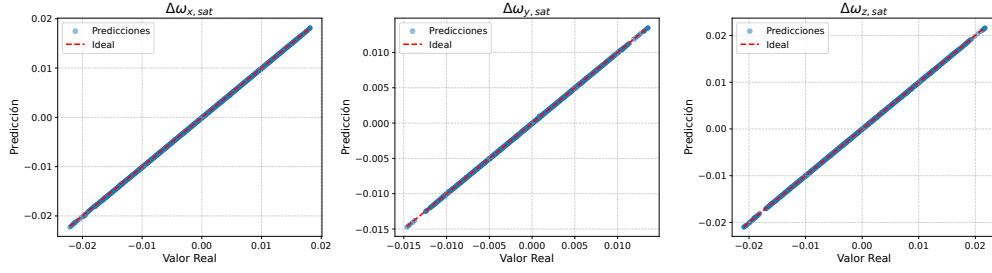


Figura E.1 Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{C}$ al ser entrenada con 2000 simulaciones

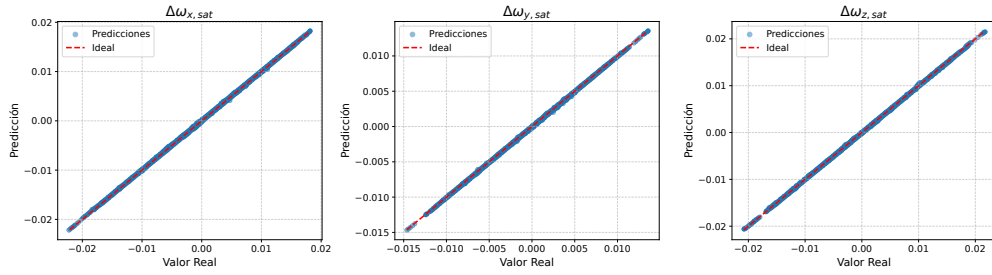


Figura E.2 Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{F}$ al ser entrenada con 2000 simulaciones

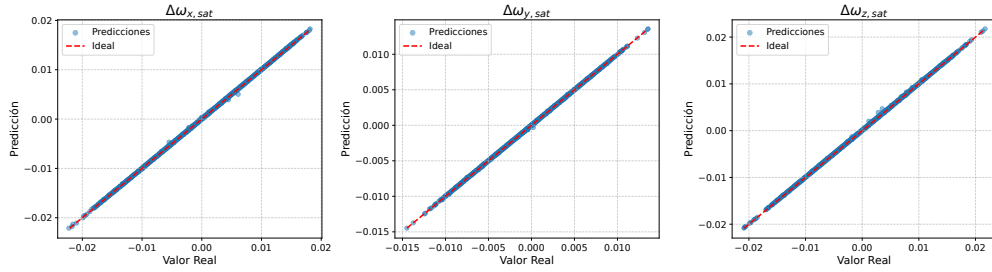


Figura E.3 Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{C}_2$ al ser entrenada con 2000 simulaciones

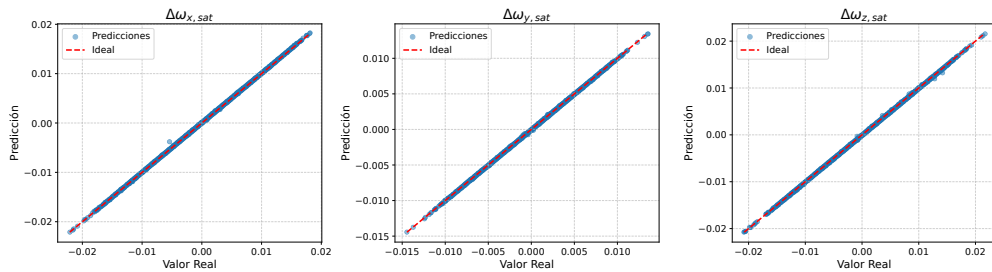


Figura E.4 Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{F}_2$ al ser entrenada con 2000 simulaciones

F Figuras Resultados Cuaterniones

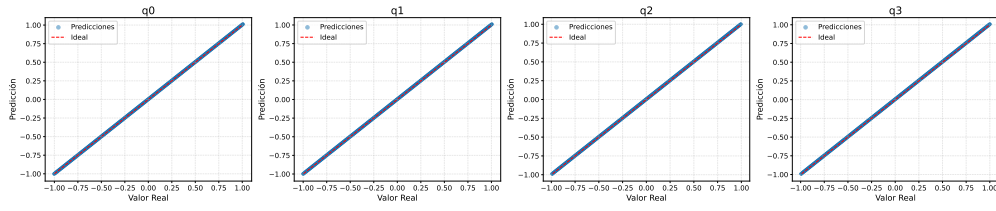


Figura F.1 Gráficos de correlación entre las variables predichas y reales de cada componente del cuaternion, de la ANN $\mathbb{C}$ al ser entrenada con 2000 simulaciones

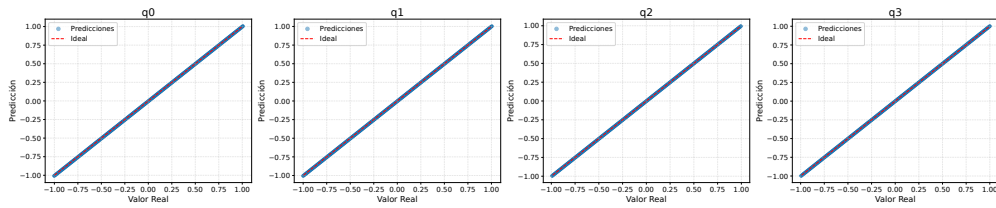


Figura F.2 Gráficos de correlación entre las variables predichas y reales de cada componente del cuaternion, de la ANN $\mathbb{F}$ al ser entrenada con 2000 simulaciones

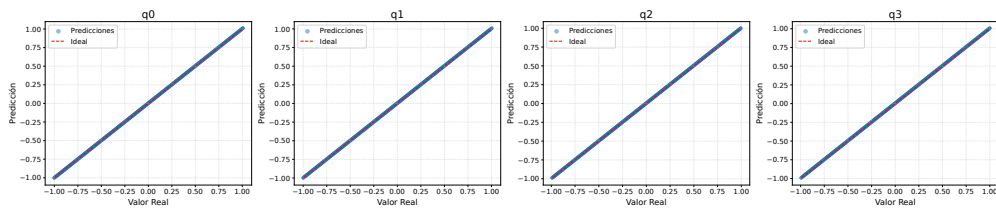


Figura F.3 Gráficos de correlación entre las variables predichas y reales de cada componente del cuaternion, de la ANN $\mathbb{C}_2$ al ser entrenada con 2000 simulaciones

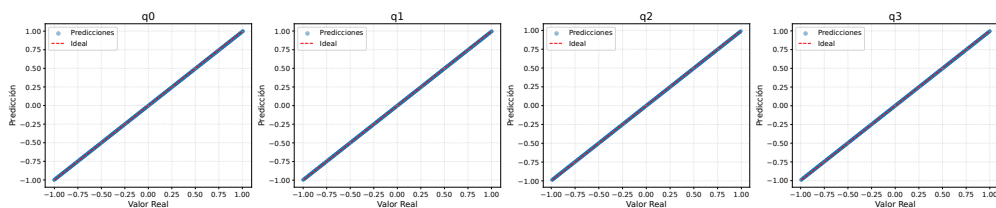


Figura F.4 Gráficos de correlación entre las variables predichas y reales de cada componente del cuaternion, de la ANN $\mathbb{F}_2$ al ser entrenada con 2000 simulaciones

G Figuras Resultados Velocidades angulares en ventanas

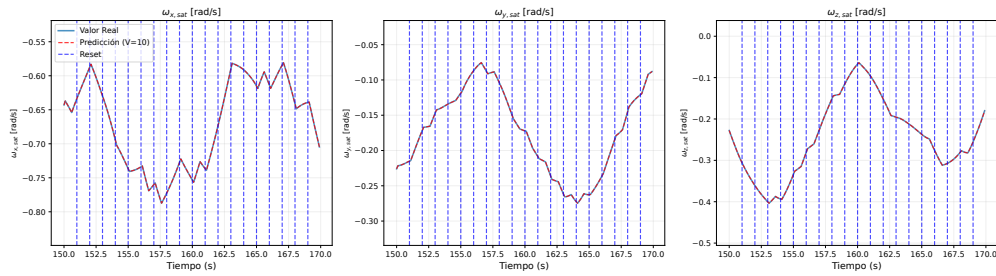


Figura G.1 Gráficos de predicción recurrente para ventanas de 1s de las variables reales y predichas de cada componente del vector ω_{sat} , para la ANN $\mathbb{C}$

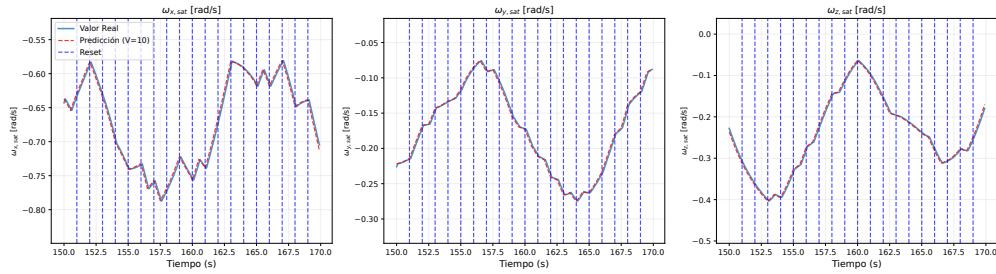


Figura G.2 Gráficos de predicción recurrente para ventanas de 1s de las variables reales y predichas de cada componente del vector ω_{sat} , para la ANN $\mathbb{F}$

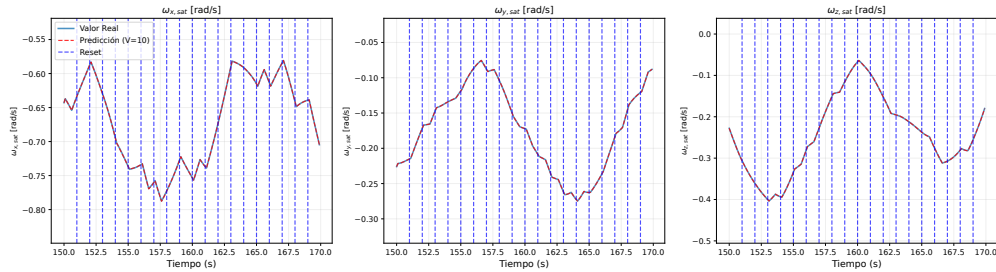


Figura G.3 Gráficos de predicción recurrente para ventanas de 1s de las variables reales y predichas de cada componente del vector ω_{sat} , para la ANN $\mathbb{C}_2$

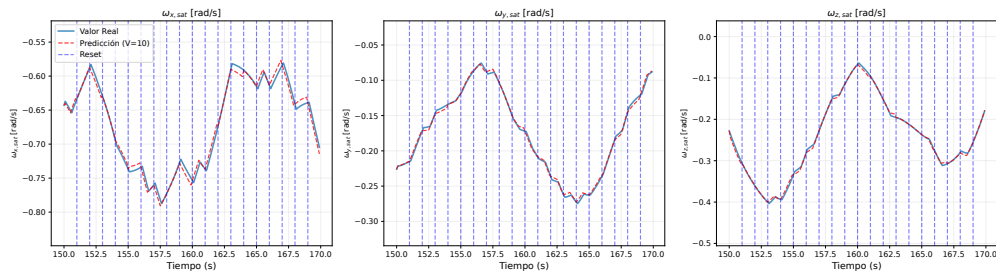


Figura G.4 Gráficos de predicción recurrente para ventanas de 1s de las variables reales y predichas de cada componente del vector ω_{sat} , para la ANN $\mathbb{F}_2$

H Figuras Resultados Cuaterniones en ventanas

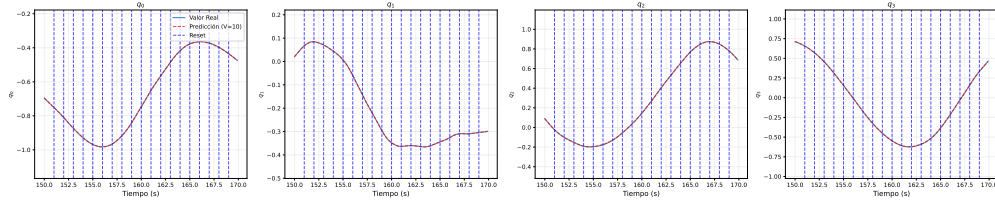


Figura H.1 Gráficos de predicción recurrente para ventanas de 1s de las variables reales y predichas de cada componente del cuaternion, para la ANN $\mathbb{C}$

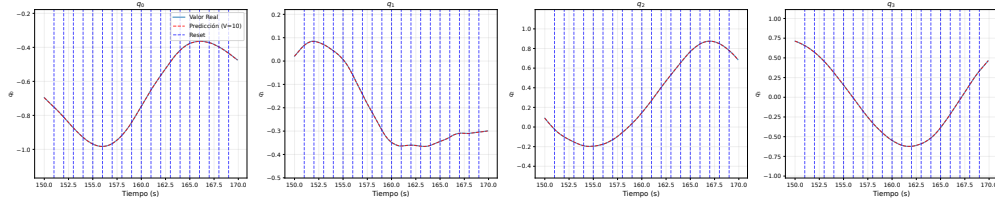


Figura H.2 Gráficos de predicción recurrente para ventanas de 1s de las variables reales y predichas de cada componente del cuaternion, para la ANN $\mathbb{F}$

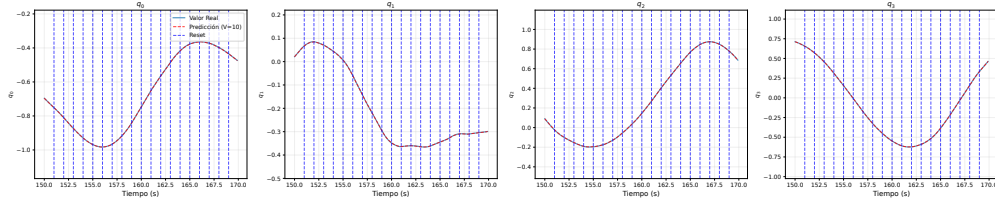


Figura H.3 Gráficos de predicción recurrente para ventanas de 1s de las variables reales y predichas de cada componente del cuaternion, para la ANN $\mathbb{F}_2$

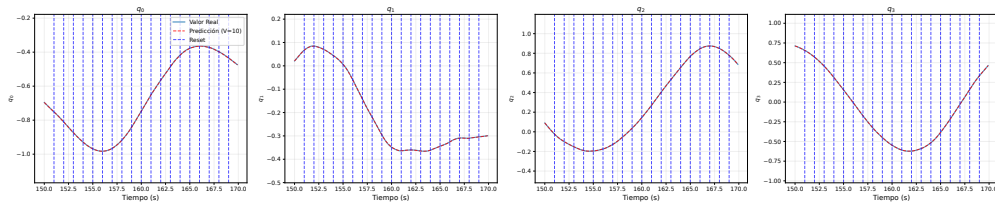


Figura H.4 Gráficos de predicción recurrente para ventanas de 1s de las variables reales y predichas de cada componente del cuaternion, para la ANN $\mathbb{F}_2$

I Figuras Resultados entrenamiento progresivo

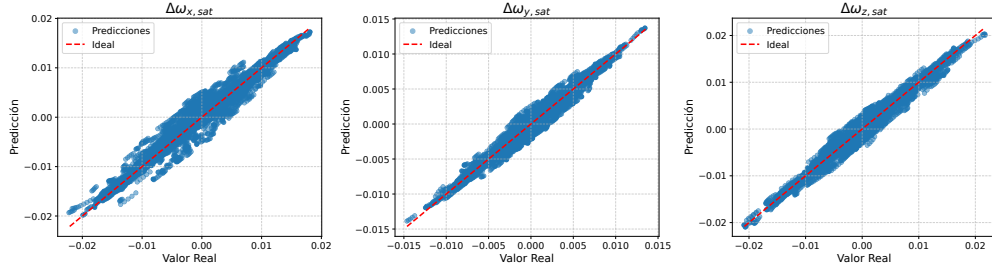


Figura I.1 Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{C}$ al ser entrenada con 5 simulaciones

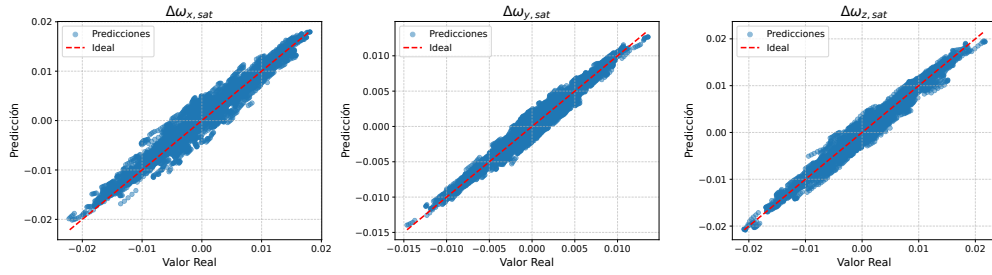


Figura I.2 Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{F}$ al ser entrenada con 5 simulaciones

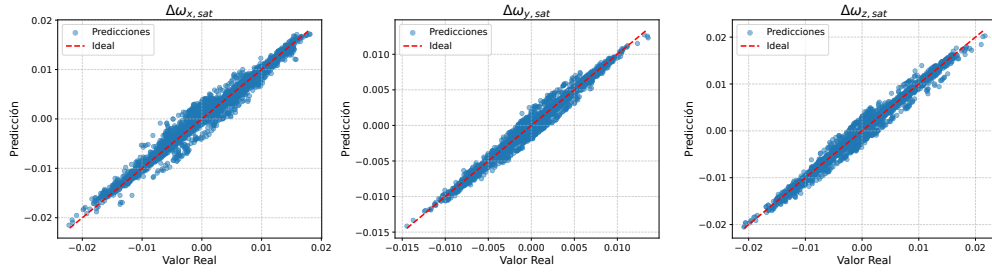


Figura I.3 Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{C}_2$ al ser entrenada con 5 simulaciones

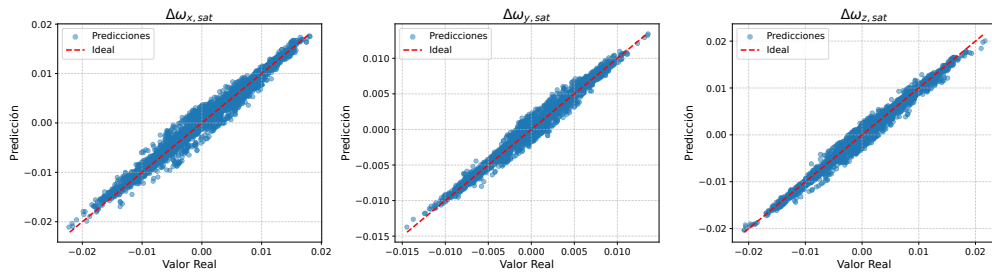


Figura I.4 Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{F}_2$ al ser entrenada con 5 simulaciones

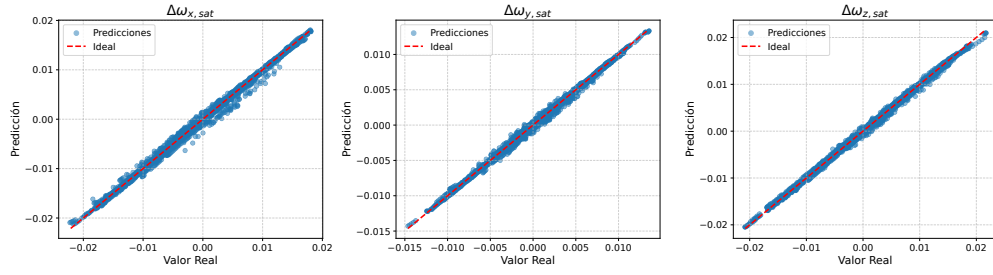


Figura I.5 Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{C}$ al ser entrenada con 50 simulaciones

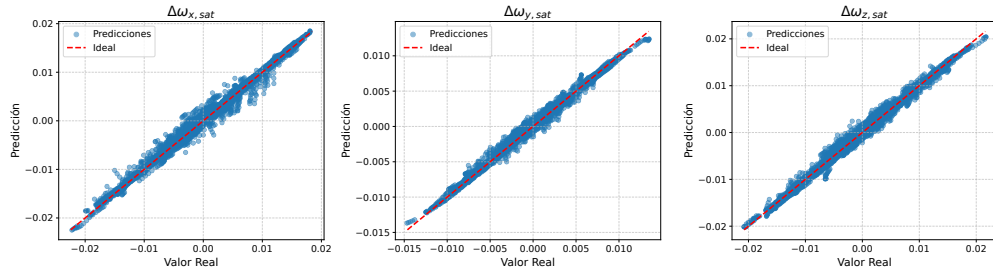


Figura I.6 Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{F}$ al ser entrenada con 50 simulaciones

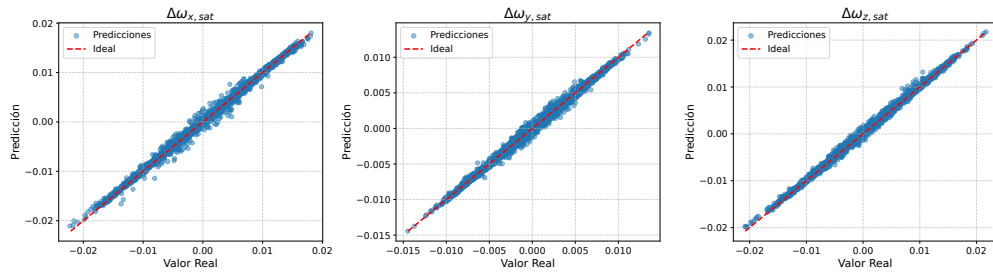


Figura I.7 Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{C}_2$ al ser entrenada con 50 simulaciones

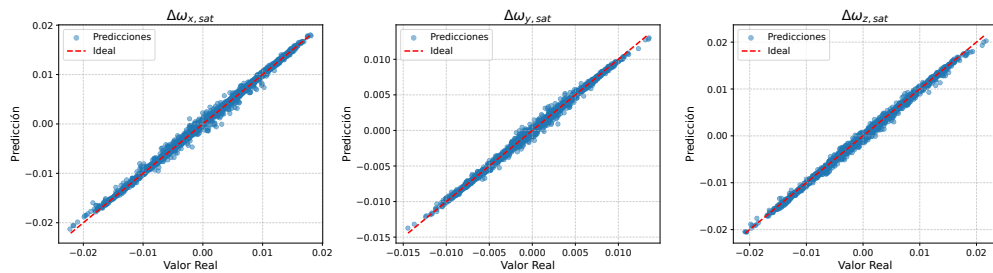


Figura I.8 Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{F}_2$ al ser entrenada con 50 simulaciones

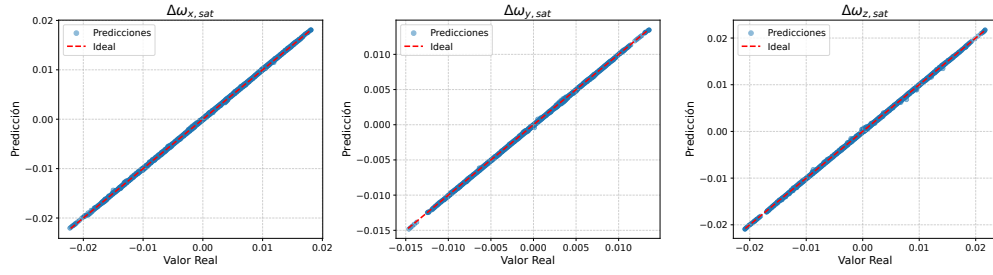


Figura I.9 Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{C}$ al ser entrenada con 500 simulaciones

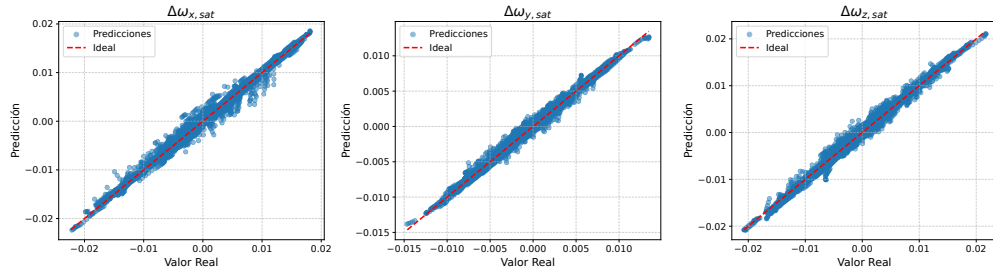


Figura I.10 Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{F}$ al ser entrenada con 500 simulaciones

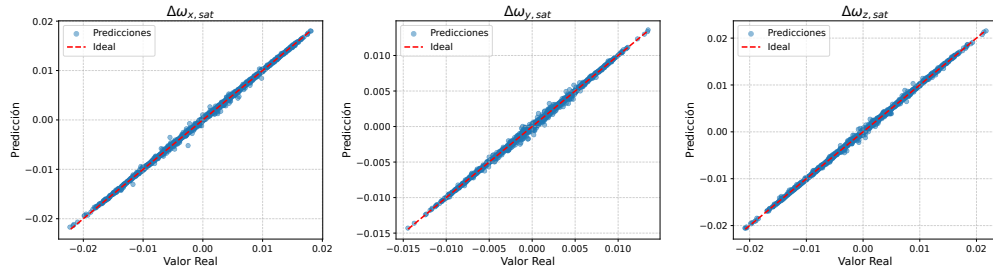


Figura I.11 Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{C}_2$ al ser entrenada con 500 simulaciones

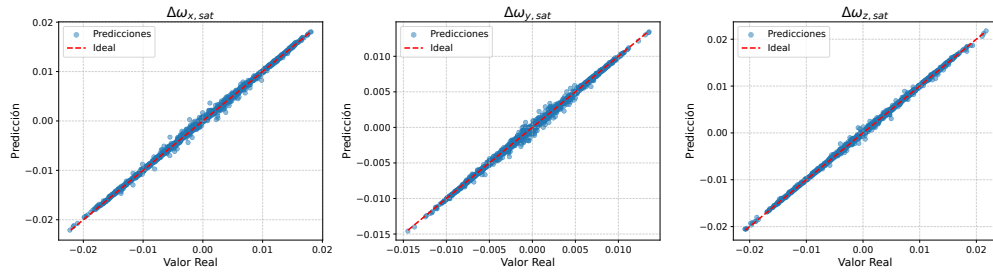


Figura I.12 Gráficos de correlación entre las variables predichas y reales de cada componente del vector $\Delta\omega_{sat}$ de la ANN $\mathbb{F}_2$ al ser entrenada con 500 simulaciones