



UNIVERSIDAD TÉCNICA  
FEDERICO SANTA MARÍA

Departamento de Obras Civiles

**APLICACIÓN DE MODELOS DE APRENDIZAJE PROFUNDO  
INFORMADOS POR LA FÍSICA PARA LA PREDICCIÓN DE  
RESPUESTAS DINÁMICAS EN SISTEMAS ESTRUCTURALES  
NO-LINEALES FRENTE A EXCITACIONES SÍSMICAS DE GRAN  
DENSIDAD TEMPORAL**

Tesis de Grado y Memoria de Título presentada por

**Matías Ignacio Cambara Norambuena**

como requisito parcial para optar al título de

**Ingeniero Civil**

y el grado de

**Magíster en Ciencias de la Ingeniería Civil**

Profesor Guía  
Gastón Fernandois Cornejo

MARZO, 2026



## CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

### 1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

**Tipo de monografía (marcar una opción):**  Memoria o trabajo de título  Tesis de Postgrado

**Título del trabajo:** Aplicación de Modelos de Aprendizaje Profundo Informados por la Física para la Predicción de Respuestas Dinámicas en Sistemas Estructurales No-lineales frente a Excitaciones Sísmicas de Gran Densidad Temporal

**Nombre del candidato(a):** Matías Ignacio Cambara Norambuena

**Carrera / Grado:** Ingeniería Civil y el Grado de Magister en Ciencias de la Ingeniería Civil

**Campus:** San Joaquín **Departamento:** Obras Civiles

### 2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Gastón Andrés Fernandois Cornejo, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

### 3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por (**marcar una opción**):

6 meses  12 meses  2 años  3 años  5 años  10 años

**Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):**

---

---

---

### 4.- FIRMAS

**Profesor(a) guía o director(a) de memoria o tesis:**

Fecha: 1-ABR-2026

Firma:

**Estudiante o Candidato(a):**

Fecha: 13 de Marzo 2026

Firma:

## **Declaración de Autoría**

Declaro por la presente que, salvo donde se haga referencia específica al trabajo de otras personas, el contenido de esta tesis es original y no ha sido presentado, total ni parcialmente, para la obtención de otro grado o título en esta u otra universidad. Esta tesis es de mi autoría y no contiene material que sea resultado de un trabajo realizado en colaboración con otros, salvo lo especificado en el texto y en los agradecimientos.

Asimismo, se hace constar el uso de herramientas de inteligencia artificial exclusivamente con fines de revisión gramatical y mejora de estilo, sin que ello afecte la originalidad de los análisis ni las conclusiones aquí presentadas.

Matías Cambara  
Enero 2026

## **Agradecimientos**

Escribir estas líneas no solo significa cerrar un ciclo, sino también mirar hacia atrás y ver todo lo que costó llegar hasta aquí. Ha sido un camino marcado por el sacrificio, horas interminables frente al computador, noches en vela, altos niveles de estrés y litros de café. Hoy, con la profunda satisfacción de ver este proyecto terminado, sé que todo el esfuerzo ha valido la pena y quiero usar este espacio para agradecer a quienes fueron fundamentales en este viaje.

Agradezco también a los profesores que cruzaron mi camino por compartir su conocimiento y entregarme las herramientas necesarias para llegar a esta instancia.

A mi familia y a mis amigos, gracias por la confianza inquebrantable y el apoyo constante a lo largo de todo este proceso. De manera muy especial, agradezco profundamente a mi mamá por ser siempre un apoyo incondicional. Y a mi novia, que comenzó a acompañarme en estos últimos años: gracias por escucharme, por tu paciencia y por hacer mucho más ligeros los días de mayor presión.

Finalmente, guardo un espacio muy especial en mi corazón y un recuerdo entrañable para mi mascota, quien partió antes de haber terminado este viaje. Su compañía incondicional y sus silencios fueron mi refugio, mi mejor terapia en las madrugadas de estudio y el soporte emocional que necesité para sobrellevar la carga de estos años. A todos, gracias de corazón.

## Abstract

This thesis addresses the challenge of efficiently predicting nonlinear seismic response and permanent structural damage (residual drift), overcoming the high computational cost of traditional time-history analyses and the lack of physical consistency inherent in conventional "black-box" neural networks. The research focuses on Physics-Informed Neural Networks (PINNs) as a hybrid solution.

In the first stage, reference architectures (PhyCNN and PhyLSTM) are evaluated under complex seismic excitations, including long-duration Chilean records. The results reveal that, while these models capture moderate dynamics, they present critical limitations when facing systems with severe hysteresis and stiffness degradation, accumulating numerical errors that prevent a reliable estimation of residual displacements.

To overcome these barriers, this thesis proposes and develops a new architecture, RK4Net, which explicitly integrates a fourth-order Runge-Kutta numerical scheme into the computational graph. It is identified that the direct training of these deep integrators proves unstable due to the high non-convexity of the loss surface (the *complexity paradox*), an obstacle that is resolved through an innovative Warm Start strategy (transfer learning from a simplified EulerNet model). The implementation of this strategy allows RK4Net to effectively eliminate the numerical drift error characteristic of first-order integrators, achieving high-precision predictions of the complete hysteretic response and permanent displacements. Thus, this work establishes a robust foundation for the use of autonomous surrogate models in earthquake engineering, combining the efficiency of deep learning with the rigor of high-order numerical methods.

**Keywords:** Structural Dynamics, Physics-Informed Neural Networks (PINNs), Deep Learning, Hysteretic Systems, RK4Net, Earthquake Engineering

## Resumen

Esta tesis aborda el desafío de predecir la respuesta sísmica no-lineal y el daño estructural permanente (drift residual) de manera eficiente, superando el alto costo computacional de los análisis de tiempo-historia tradicionales y la falta de consistencia física de las redes neuronales convencionales (caja negra). La investigación se centra en las Redes Neuronales Informadas por la Física (PINNs) como solución híbrida.

En una primera etapa, se evalúan arquitecturas de referencia (PhyCNN y PhyLSTM) bajo excitaciones sísmicas complejas, incluyendo registros chilenos de larga duración. Los resultados revelan que, aunque estos modelos capturan dinámicas moderadas, presentan limitaciones críticas ante sistemas con histéresis severa y degradación de rigidez, acumulando errores numéricos que impiden una estimación fiable de los desplazamientos residuales.

Para superar estas barreras, la tesis propone y desarrolla una nueva arquitectura, RK4Net, que integra explícitamente un esquema numérico de Runge-Kutta de cuarto orden en el grafo computacional. Se identifica que el entrenamiento directo de estos integradores profundos resulta inestable debido a la elevada no-convexidad de la superficie de pérdida (*paradoja de la complejidad*), un obstáculo que se resuelve mediante una innovadora estrategia de Warm Start (transferencia de aprendizaje desde un modelo EulerNet simplificado). La implementación de esta estrategia permite que RK4Net elimine eficazmente el error de deriva numérico característico de los integradores de primer orden, logrando predecir con alta precisión la respuesta histerética completa y los desplazamientos permanentes. De este modo, el trabajo establece una base robusta para el uso de modelos surrogados autónomos en ingeniería sísmica, combinando la eficiencia del aprendizaje profundo con el rigor de los métodos numéricos de alto orden.

**Palabras clave:** Dinámica Estructural, Redes Neuronales Basadas en la Física (PINNs), Deep Learning, Sistemas Histeréticos, RK4Net, Ingeniería Sísmica

# Tabla de Contenidos

<b>1. Introducción</b>	<b>1</b>
1.1. Contexto de la Investigación . . . . .	1
1.2. Objetivos . . . . .	4
1.2.1. Objetivo General . . . . .	4
1.2.2. Objetivos Específicos . . . . .	4
1.3. Estructura de la Tesis . . . . .	5
<b>2. Marco Teórico</b>	<b>6</b>
2.1. Notación, convenciones y supuestos . . . . .	6
2.1.1. Notación y símbolos . . . . .	6
2.1.2. Supuestos estructurales . . . . .	7
2.1.3. Dominio temporal y discretización . . . . .	8
2.2. Fundamentos de dinámica estructural no-lineal . . . . .	8
2.2.1. Ecuaciones de movimiento (MDOF) con entrada sísmica . . . . .	8
2.2.2. Formulación en espacio de estados . . . . .	9
2.3. Análisis tiempo–historia no-lineal (NTHA) . . . . .	11
2.3.1. Discretización espacial (MEF) y ensamblaje . . . . .	11
2.3.2. Integración temporal . . . . .	12
2.3.3. Imposición iterativa del equilibrio: Newmark y Newton–Raphson. . . . .	14
2.4. Modelos histeréticos y degradación . . . . .	16
2.4.1. Modelos histeréticos <i>model-driven</i> . . . . .	16
2.4.2. Polinomiales por tramos . . . . .	17
2.4.3. Diferenciales . . . . .	18
2.4.4. Modelos por operadores . . . . .	19
2.4.5. Implicancias en el desplazamiento residual ( <i>drift</i> residual) . . . . .	19
2.5. Sistemas de aislamiento sísmico y péndulo friccional (FPS) . . . . .	20
2.6. Excitaciones sísmicas: caracterización y preparación de registros . . . . .	21
2.6.1. Medidas de intensidad y contenido espectral . . . . .	21
2.6.2. Selección y escalado de registros . . . . .	21
2.6.3. Preprocesamiento: corrección de línea base, filtrado y remuestreo . . . . .	22
2.6.4. Registros artificiales: Kanai–Tajimi modificado y envolventes no estacionarias . . . . .	22
2.6.5. Variabilidad registro–a–registro y consecuencias para la generalización . . . . .	26

2.7.	Aprendizaje profundo para series temporales estructurales . . . . .	26
2.7.1.	Fundamentos de entrenamiento: propagación directa y retropropagación . . . . .	26
2.7.2.	Arquitecturas de redes neuronales . . . . .	30
2.7.3.	Protocolos de partición y normalización de datos . . . . .	38
2.7.4.	Indicadores de desempeño y métricas de evaluación . . . . .	40
2.8.	Redes neuronales informadas por la física (PINNs) . . . . .	45
2.9.	Modelos PINN de referencia en dinámica estructural . . . . .	46
2.9.1.	PhyCNN: red convolucional guiada por la física . . . . .	46
2.9.2.	PhyLSTM: Redes LSTM multi-bloque informadas por la física . . . . .	54
2.9.3.	Esquema de optimización y criterios de entrenamiento . . . . .	61
2.9.4.	Síntesis general y proyección del análisis comparativo . . . . .	61
2.10.	Síntesis del marco teórico y conexión con capítulos posteriores . . . . .	62
<b>3.</b>	<b>Estudio Comparativo de Arquitecturas Recurrentes y Convolucionales Informadas por la Física</b>	
	<b>bajo Demandas Sísmicas Severas</b>	<b>64</b>
3.1.	Introducción . . . . .	64
3.2.	Metodología . . . . .	67
3.2.1.	Formulación física y modelo secuencia–a–secuencia . . . . .	68
3.2.2.	Redes Neuronales Convolucionales 1D y LSTM . . . . .	70
3.2.3.	Indicadores de desempeño y métricas de evaluación . . . . .	72
3.2.4.	Arquitecturas <i>guided-informed</i> . . . . .	75
3.2.5.	Esquema de optimización y criterios de entrenamiento . . . . .	77
3.3.	Casos de Estudio . . . . .	79
3.4.	Escenario I: Oscilador Duffing . . . . .	79
3.4.1.	Desempeño de PhyCNN y Multi-LSTM en el Escenario I . . . . .	81
3.5.	Escenario II: Bouc–Wen BLWN . . . . .	88
3.5.1.	Desempeño de PhyCNN y Multi-LSTM en Escenario II . . . . .	89
3.6.	Escenario III: Bouc–Wen Chile . . . . .	96
3.6.1.	Desempeño de PhyCNN y Multi-LSTM en aislamiento sísmico con registros chilenos . . . . .	99
3.7.	Conclusiones y Trabajo Futuro . . . . .	108
<b>4.</b>	<b>RK4Net: Un Integrador Neuronal Autónomo de Alto Orden para Dinámica Estructural no-lineal</b>	<b>110</b>
4.1.	Introducción . . . . .	110
4.2.	Marco Teórico: Modelado Dinámico y Espacio de Estados . . . . .	112
4.2.1.	Ecuación General de Movimiento . . . . .	112
4.2.2.	Modelos de Fuerzas de Restitución No-lineales . . . . .	113
4.2.3.	Formulación en el Espacio de Estados . . . . .	113
4.3.	Fase I: Arquitectura Base EulerNet y Validación en Sistemas Suaves . . . . .	113
4.3.1.	Formulación del Integrador Neuronal (EulerNet) . . . . .	114
4.3.2.	Arquitectura de la Red Neuronal . . . . .	114

4.3.3.	Estrategia de Entrenamiento y Función de Pérdida . . . . .	115
4.3.4.	Experimento 1: Validación en Sistema Lineal . . . . .	115
4.3.5.	Experimento 2: Generalización a No-linealidad Geométrica (Duffing) . . . . .	124
4.3.6.	Discusión General y Conclusiones de la Fase I . . . . .	131
4.4.	Fase II: Diagnóstico de Limitaciones en Sistemas Rígidos y Caóticos . . . . .	132
4.4.1.	Sistemas de Prueba de Alta Complejidad . . . . .	132
4.4.2.	Caso A: Oscilador de Duffing de Doble Pozo (Bi-estable) . . . . .	132
4.4.3.	Caso B: Sistema con Histéresis Severa (Bouc-Wen) . . . . .	137
4.4.4.	Metodología de Diagnóstico: Análisis de Convergencia del Esquema Numérico . . .	143
4.4.5.	Conclusión Final de la Fase II . . . . .	146
4.5.	Fase III: Arquitectura Avanzada RK4Net y Solución al Drift . . . . .	147
4.5.1.	Formulación Matemática del Modelo RK4Net . . . . .	147
4.5.2.	Entrenamiento Directo (Cold Start) y la Paradoja de Optimización . . . . .	148
4.5.3.	Solución Propuesta: Estrategia de Transferencia (Warm Start) . . . . .	150
4.5.4.	Conclusión General del Capítulo . . . . .	154
4.6.	Conclusiones y Trabajo Futuro . . . . .	155
<b>5.</b>	<b>Conclusiones y Trabajo Futuro</b>	<b>156</b>
5.1.	Síntesis General . . . . .	156
5.2.	Conclusiones Principales . . . . .	156
5.2.1.	Limitaciones de los Enfoques Guided-Informed (PhyCNN/PhyLSTM) . . . . .	156
5.2.2.	Ventajas de la Arquitectura Solver-Informed (RK4Net) . . . . .	157
5.2.3.	Generalización y Robustez . . . . .	158
5.3.	Limitaciones y Recomendaciones Metodológicas . . . . .	158
5.4.	Trabajo Futuro . . . . .	160
5.4.1.	Extensión a Sistemas Exógenos y MDOF . . . . .	160
5.4.2.	Mejoras en la Arquitectura y Entrenamiento . . . . .	160
5.4.3.	Validación Experimental y Aplicaciones en Tiempo Real . . . . .	161
	<b>Referencias</b>	<b>162</b>
	<b>Apéndice A. Recursos Computacionales y Reproducibilidad</b>	<b>169</b>
	<b>Apéndice B. Fundamentos del Entrenamiento de Redes Neuronales</b>	<b>171</b>

# Lista de Figuras

2.1. Diferencia entre métodos model-driven y data driven. (Fuente: Wang et al. (2023a)) . . . . .	17
2.2. Ciclos de Histéresis del modelo Bouc-Wen para diferentes parámetros (Fuente: Charalampakis (2010)) . . . . .	19
2.3. Evolución del desplazamiento relativo bajo excitación sísmica. La componente plástica acumulada se refleja en el <i>drift</i> residual, que persiste aun después de la disipación de la respuesta transitoria. . . . .	20
2.4. Representación gráfica del filtro Kanai-Tajimi . . . . .	23
2.5. Funciones de densidad espectral de potencia de Kanai-Tajimi y Clough-Penzien con $\omega_g = 8\pi$ , $\omega_f = 0.8\pi$ y $\zeta_g = \zeta_f = 0.6$ . Fuente: Zhu et al. (2017) . . . . .	24
2.6. Impacto de la función de modulación de Amin y Ang en la simulación de la aceleración del suelo durante un sismo. . . . .	25
2.7. Esquema de una neurona artificial: combinación lineal de entradas y pesos, más un sesgo $b$ , seguida de una función de activación $\sigma(\cdot)$ . . . . .	27
2.8. Esquema de un perceptrón multicapa (MLP), también conocido como red neuronal totalmente conectada (FCNN). . . . .	27
2.9. Representación capa convolucional con múltiples entradas y aplicación de solo un filtro en la imagen de la izquierda y múltiples filtros en la imagen de la derecha . . . . .	32
2.10. Representación de las operaciones de dilatación, stride y padding en redes convolucionales 1D con un canal de entrada. . . . .	34
2.11. Representación capa convolucional con una entrada y aplicación de solo un filtro en la imagen de la izquierda y multiples filtros en la imagen de la derecha . . . . .	34
2.12. Flujo de datos en la unidad de tiempo $t$ para la Arquitectura de una Red Neuronal Recurrente (RNN). . . . .	35
2.13. Flujo de datos en la unidad de tiempo $t$ de una unidad de LSTM. La puerta de olvido y la celda de memoria evitan problemas de desvanecimiento o explosión de gradiente. . . . .	37
2.14. Esquema básico de una LSTM desenrollada, donde el estado oculto $h_t$ se calcula de manera recurrente utilizando las entradas $x_t$ y el estado oculto anterior $h_{t-1}$ . . . . .	37
2.15. Esquema de una Red LSTM profunda con múltiples capas ocultas, donde las entradas $\mathbf{x}_t$ se procesan a través de varias capas de estados ocultos $\mathbf{h}_t^{(1)}, \dots, \mathbf{h}_t^{(l)}, \mathbf{h}_t^{(l+1)}$ , para modelar secuencias complejas. . . . .	38
2.16. Arquitectura y marco de entrenamiento del modelo PhyCNN-MO. . . . .	50
2.17. Arquitectura de la red PhyCNN-SO. . . . .	53

2.18. Bloque LSTM genérico utilizado en las arquitecturas PhyLSTM. Cada bloque incluye tres capas LSTM (100 unidades) con activación ReLU y una o dos capas densas distribuidas en el tiempo. En los bloques de la PhyLSTM <sup>2</sup> se incorpora una capa densa intermedia de 100 unidades, ausente en la PhyLSTM <sup>3</sup> . . . . .	56
2.19. Arquitectura PhyLSTM <sup>2</sup> , donde cada bloque LSTM representa el bloque presentado en la sección 2.9.2 . . . . .	57
2.20. Arquitectura PhyLSTM <sup>3</sup> , donde cada bloque LSTM representa el bloque presentado en la sección 2.9.2 . . . . .	59
3.1. Esquema general de la metodología. Se consideran tres conjuntos de datos (Duffing, Bouc–Wen BLWN y Bouc–Wen Chile) que alimentan las fases de entrenamiento, inferencia y evaluación de cuatro arquitecturas <i>guided-informed</i> . . . . .	67
3.2. Capa convolucional 1D con múltiples entradas: un único filtro (izquierda) y múltiples filtros (derecha). . . . .	71
3.3. Esquema de LSTM: <b>(a)</b> Celda LSTM elemental: compuertas de entrada ( $i_t$ ), olvido ( $f_t$ ) y salida ( $o_t$ ) que regulan el estado de memoria $\mathbf{c}_t$ . y <b>(b)</b> Arquitectura profunda: apilamiento de capas para modelar dependencias de largo alcance. . . . .	72
3.4. Arquitectura y marco de entrenamiento del modelo <i>PhyCNN–MO</i> . La red predice simultáneamente $\hat{\mathbf{u}}$ , $\hat{\mathbf{v}}$ , $\hat{\mathbf{g}}$ e impone restricciones cinemáticas y dinámicas mediante diferenciación numérica. . . . .	76
3.5. Arquitectura de la red <i>PhyCNN–SO</i> . Predice únicamente el desplazamiento estructural y aplica una restricción física supervisada sobre la aceleración derivada. . . . .	77
3.6. Bloque LSTM genérico utilizado en las arquitecturas <i>PhyLSTM</i> . Contiene tres capas LSTM (100 unidades) con activación ReLU y una capa densa <i>TimeDistributed</i> . . . . .	78
3.7. Arquitectura <i>PhyLSTM<sup>2</sup></i> . Consta de dos bloques LSTM conectados en serie; el segundo bloque ajusta la fuerza interna $\hat{\mathbf{g}}^{(2)}$ e impone la restricción dinámica. . . . .	78
3.8. Arquitectura <i>PhyLSTM<sup>3</sup></i> . Introduce un tercer bloque LSTM para modelar la dinámica de la variable histerética latente $\mathbf{z}$ , reforzando la consistencia temporal y ampliando la capacidad de representar efectos de memoria no-lineales. . . . .	79
3.9. Esquema de la fuerza restauradora vs. desplazamiento para el oscilador de Duffing (Escenario I). Basado en los parámetros técnicos de Zhang et al. (2020a). . . . .	80
3.10. Distribución de L2RE para las tres variables de salida en el oscilador de Duffing (conjunto de prueba), el cuarto grupo corresponde al error bidimensional $L2RE_{2D}$ en el plano fuerza–desplazamiento. El eje vertical se recorta en 1.25 para mejorar la legibilidad. . . . .	84
3.11. Evolución de la pérdida durante entrenamiento y validación (suavizada con ventana móvil de 500 épocas). Eje vertical en escala logarítmica. . . . .	85
3.12. Comparación temporal de las predicciones para los registros de mejor (fila superior) y peor (fila inferior) desempeño. La referencia se muestra con línea discontinua. . . . .	86

3.13. PSD de desplazamiento, velocidad y fuerza interna normalizada para los registros de mejor desempeño (eje vertical logarítmico). . . . .	86
3.14. Espectrograma de desplazamiento para el registro con mejor desempeño (registro 12): (a) real, (b) PhyCNN–SO, (c) <i>error PSD</i> (dB). . . . .	86
3.15. Espectrograma de fuerza interna normalizada para el registro con mejor desempeño (registro 9): (a) real, (b) PhyLSTM–2 (reconstruida), (c) <i>error PSD</i> (dB). . . . .	87
3.16. Curvas fuerza–desplazamiento correspondientes al registro de mejor desempeño (índice 81). La trayectoria refleja una relación no-lineal suave sin histéresis. El error asociado es <b>L2RE–2D</b> . . . . .	88
3.17. Distribución del error relativo cuadrático medio (L2RE) para desplazamiento, velocidad y fuerza interna normalizada en el conjunto de prueba del sistema Bouc–Wen ante ruido blanco filtrado. El eje vertical se recorta en 2.3 para mejorar la legibilidad. . . . .	91
3.18. Curvas de pérdida suavizadas (ventana móvil de 500 épocas) durante el entrenamiento y validación de los modelos para el sistema Bouc–Wen. El eje vertical se representa en escala logarítmica. . . . .	92
3.19. Comparación tiempo–historia de desplazamiento, velocidad y fuerza interna normalizada para los registros de mejor (fila superior) y peor (fila inferior) desempeño en el escenario Bouc–Wen ante ruido blanco. Las líneas continuas representan las respuestas de referencia y las líneas de color las predicciones de cada modelo. . . . .	93
3.20. Espectros de densidad de potencia (PSD) de desplazamiento, velocidad y fuerza interna normalizada para los registros de mejor desempeño en el sistema Bouc–Wen ante ruido blanco. Las líneas discontinuas son las respuestas de referencia y las curvas de color las predicciones de cada modelo. . . . .	94
3.21. Espectrogramas de desplazamiento para el registro con mejor desempeño (registro 2): (a) real, (b) PhyCNN–SO/PhyLSTM–2, (c) <i>error PSD</i> (dB). . . . .	95
3.22. Curvas fuerza–desplazamiento correspondientes al registro de mejor desempeño del sistema Bouc–Wen bajo excitación de ruido blanco filtrado. . . . .	96
3.23. Diagrama de bloques del sistema SDOF con modelo Bouc–Wen completo en <i>Simulink</i> , mostrando la integración del modelo de histéresis y las entradas sísmicas. . . . .	97
3.24. Diagrama de bloques de la implementación inicial del modelo Bouc–Wen en <i>Simulink</i> con entrada sinusoidal. . . . .	97
3.25. Mapa de los eventos sísmicos registrados (círculos) y estaciones sismológicas (cuadrados). Se incluyen 21 eventos observados por 120 estaciones, generando un total de 270 registros correspondientes a ambas direcciones de movimiento. . . . .	98
3.26. Distribución del L2RE en desplazamiento, velocidad y fuerza interna normalizada para el sistema Bouc–Wen representativo de un edificio aislado. . . . .	101
3.27. Evolución de la pérdida (suavizada con ventana de 500 épocas) durante entrenamiento y validación del escenario chileno. . . . .	102
3.28. Predicciones temporales para el sistema Bouc–Wen aislado. La primera fila corresponde a los mejores registros y la segunda a los peores, según L2RE. . . . .	103

3.29. PSD de desplazamiento, velocidad y fuerza interna normalizada para el registro de mejor desempeño del sistema Bouc–Wen aislado bajo registros chilenos. . . . .	104
3.30. Espectrograma de desplazamiento para el registro con mejor desempeño (registro 100): (a) real, (b) PhyCNN–SO, (c) <i>error PSD</i> (dB). . . . .	105
3.31. Espectrograma de fuerza interna normalizada para el registro con mejor desempeño (registro 8): (a) real, (b) PhyLSTM–2 (reconstruida), (c) <i>error PSD</i> (dB). . . . .	105
3.32. Curvas fuerza–desplazamiento correspondientes al registro de mejor desempeño (registro 8). . . . .	106
3.33. Desplazamiento residual absoluto por registro para los 100 registros con mayor desplazamiento residual (ordenado por el residual verdadero) . . . . .	107
4.1. Distribución de las 100 condiciones iniciales aleatorias. Los rangos se han acotado a valores realistas de operación elástica ( $\pm 5$ cm de desplazamiento). . . . .	119
4.2. Evaluación de trayectoria temporal en simulación autónoma. La predicción neuronal (rojo/azul/magenta) se superpone casi perfectamente a la realidad (gris), validando la captura de la frecuencia y el amortiguamiento sin divergencia. . . . .	122
4.3. Retrato de fase ( $u$ vs. $\dot{u}$ ) Fase I, Lineal. La coincidencia geométrica de las espirales confirma que el campo vectorial aprendido preserva la estructura disipativa del sistema. . . . .	123
4.4. Dispersión del error L2RE para las 18 trayectorias de prueba. La consistencia en $u$ y $v$ confirma la fiabilidad del modelo ante diversas condiciones iniciales. . . . .	124
4.5. Comparación de trayectorias en el sistema Duffing. La red sigue con precisión los cambios de frecuencia y fase, capturando la dependencia entre rigidez y amplitud instantánea. . . . .	127
4.6. Retrato de fase ( $u$ vs. $\dot{u}$ ) del sistema Duffing. La predicción autorregresiva reproduce la geometría no lineal de la trayectoria y su convergencia al equilibrio. . . . .	128
4.7. Distribución del error L2RE para el sistema Duffing. A pesar del aumento en la complejidad dinámica, la red mantiene una dispersión controlada y errores medianos inferiores al 5 % en variables de estado . . . . .	129
4.8. Comparación de la relación fuerza-desplazamiento del oscilador de Duffing entre la solución analítica y la reconstrucción basada en la predicción neuronal. . . . .	130
4.9. Correlación Error vs. Amplitud Máxima. El aumento del error proporcional a la deformación identifica a las zonas de alta rigidez no-lineal como la fuente principal de inexactitud del integrador. . . . .	131
4.10. Dispersión del Error Relativo. Nótese la presencia de valores atípicos extremos (cerca del 200 %), indicativos de una falla cualitativa en la predicción de la trayectoria (divergencia de pozo). . . . .	135
4.11. Historia de tiempo en el sistema de doble pozo. Se aprecia la divergencia en fase tras los primeros ciclos de alta energía, donde la red lucha por seguir la tasa de disipación y la captura final en el pozo de potencial. . . . .	136
4.12. Retrato de Fase Duffing Doble Pozo. Aunque la red captura la topología del atractor derecho, se evidencia una dificultad para replicar la tasa exacta de convergencia inicial al equilibrio. . . . .	137

4.13. Diagnóstico de Error vs. Amplitud. Se observa una bifurcación clara: las trayectorias con suficiente energía para acercarse al punto inestable (amplitudes altas) sufren errores catastróficos, confirmando la limitación del integrador ante dinámicas complejas. . . . .	138
4.14. Historia de Tiempo del Sistema Aislado. Obsérvese cómo la velocidad y aceleración son predichas con alta precisión (paneles inferiores), mientras que el desplazamiento (panel superior) sufre una deriva o drift significativo hacia valores negativos, evidenciando la falla de la integración explícita. . . . .	142
4.15. Ciclos de Histéresis. La red replica la forma constitutiva compleja de Bouc-Wen (fuerza vs. desplazamiento relativo), pero el error de integración induce un desplazamiento horizontal de los ciclos, impidiendo el cierre perfecto de los lazos. . . . .	143
4.16. Distribución de Errores por Variable. Se evidencia el contraste dramático entre la alta precisión de las variables derivadas ( $v, a, z$ ) y la alta dispersión de la variable integral ( $u$ ). . . . .	144
4.17. Sensibilidad del Error a la Amplitud. Los errores masivos a bajas amplitudes reflejan la dificultad de predecir desplazamientos residuales pequeños con un integrador que acumula deriva constante. . . . .	144
4.18. Distribución de errores para Duffing con RK4 (Cold Start). Obsérvese la escala masiva del error en velocidad (columna central), con una mediana superior al 20 % y valores atípicos extremos, síntoma de una optimización fallida. . . . .	149
4.19. Respuesta Estructural con RK4Net (Warm Start). La deriva ha sido mitigada eficazmente. El modelo ahora es capaz de predecir desplazamientos permanentes con precisión ingenieril. . .	152
4.20. Distribución de Errores Finales (Bouc-Wen). La mediana del error en todas las variables se encuentra por debajo del 2.9 %, validando la confiabilidad del modelo. . . . .	153
4.21. Distribución de Errores para el sistema de doble pozo (RK4Net). Se observa una precisión superior en el desplazamiento (mediana < 1 %), mientras que la velocidad y aceleración reflejan la dificultad de modelar las derivadas en los saltos de pozo. . . . .	153
4.22. Retrato de Fase con RK4Net (Warm Start). La estabilidad orbital ha mejorado significativamente, capturando con mayor fidelidad la dinámica cerca de la separatriz. . . . .	154

# Lista de Tablas

3.1. Métricas de desempeño promedio (media $\pm$ desviación estándar) sobre el conjunto de prueba para el oscilador de Duffing. La métrica principal es L2RE, complementada por NRMSE y NMAE. . . . .	82
3.2. Métricas de desempeño promedio (media $\pm$ desviación estándar) sobre el conjunto de prueba del sistema Bouc–Wen ante ruido blanco. La métrica principal es L2RE, complementada por NRMSE y NMAE. La fila <i>Fuerza–Desplazamiento</i> reporta L2RE–2D. . . . .	90
3.3. Métricas de desempeño promedio (media $\pm$ desviación estándar) sobre el conjunto de prueba para el sistema Bouc–Wen representativo de edificio aislado bajo registros sísmicos chilenos reales. La métrica principal es L2RE, complementada por NRMSE y NMAE. La fila <i>Fuerza–Desplazamiento</i> reporta L2RE–2D. . . . .	100
4.1. Definición de parámetros físicos para el marco de acero simulado. . . . .	117
4.2. Resumen de Métricas de Desempeño . . . . .	123
4.3. Parámetros del Oscilador de Duffing (Escala Estructural Real). . . . .	126
4.4. Resumen de Métricas de Desempeño (Promedio en Test - Fase II) . . . . .	128
4.5. Parámetros del Sistema de Doble Pozo (2EP). . . . .	133
4.6. Resumen de Métricas de Desempeño (Promedio en Test - Duffing Doble Pozo) . . . . .	134
4.7. Parámetros Físicos y Constitutivos del Sistema de Aislamiento (Bouc-Wen). . . . .	139
4.8. Resumen de Métricas de Desempeño (Promedio en Test - Bouc-Wen) . . . . .	140
4.9. Análisis de Convergencia - Duffing Doble Pozo (Error Relativo L2RE en %) . . . . .	145
4.10. Análisis de Convergencia - Bouc-Wen (Error Relativo L2RE en %) . . . . .	145
4.11. Desempeño Comparativo: Euler vs. RK4 (Cold Start) en Bouc-Wen . . . . .	148
4.12. Resumen de Rendimiento - Duffing Doble Pozo (RK4 Cold Start) . . . . .	148
4.13. Comparación Final de Rendimiento - Sistema Bouc-Wen (L2RE Promedio) . . . . .	151
4.14. Comparación Final de Rendimiento - Duffing Doble Pozo (L2RE Promedio) . . . . .	151
A.1. Especificaciones de la infraestructura de hardware utilizada. . . . .	170

# Nomenclatura

## Símbolos romanos

**A, B, C, D** Matrices del sistema en la representación de espacio de estados

**b** Vector de sesgo (bias) en una red neuronal

**C** Matriz de amortiguamiento viscoso

**f<sub>int</sub>** Vector de fuerzas internas de restitución

**h<sub>t</sub>** Estado oculto en una red recurrente (RNN/LSTM)

**K** Matriz de rigidez

**K<sub>tan</sub>** Matriz de rigidez tangente

**M** Matriz de masa

**p(t)** Vector de fuerzas externas nodales

**r** Residuo dinámico de la ecuación de movimiento

**u,  $\dot{u}$ ,  $\ddot{u}$**  Vectores de desplazamiento, velocidad y aceleración relativos

**W** Matriz de pesos en una red neuronal

**x(t)** Vector de estado del sistema

**y** Vector de salida o mediciones

**z** Vector de variables internas histeréticas

**$\mathcal{L}$**  Función de pérdida (Loss function)

**a<sub>g</sub>(t)** Aceleración del suelo (Ground acceleration)

**c** Coeficiente de amortiguamiento (escalar)

**D** Número de canales de entrada o dimensión de características

**g(t)** Fuerza interna normalizada por la masa

$I_A$	Intensidad de Arias
$K$	Tamaño del kernel en una convolución
$k_0$	Rigidez elástica inicial
$m$	Masa (escalar)
$N$	Número de muestras o pasos de tiempo
$S_a$	Pseudo-aceleración espectral
$T$	Duración total o periodo fundamental
$t$	Variable temporal
$u_y$	Desplazamiento de fluencia
$e_t$	Error instantáneo de predicción
$f_s$	Vector de fuerzas de restitución (notación específica)
$h_l$	Vector de activación de la capa oculta $l$
$k$	Rigidez (general)
$k_3$	Coefficiente de rigidez cúbica (Duffing)
$\mathbf{k}_i$	Pendientes intermedias del método Runge-Kutta
$\mathcal{N}_\theta$	Función de la Red Neuronal parametrizada
$T_n$	Periodo natural de vibración
$v(t)$	Velocidad relativa (escalar)
$\mu$	Media para normalización Z-score
$n, A$	Parámetros de forma adicionales del modelo Bouc-Wen
$\psi$	Vector de parámetros físicos del sistema
$\sigma$	Desviación estándar para normalización Z-score
$\xi$	Razón de amortiguamiento crítico (notación alternativa a $\zeta$ )

Drift Residual Desplazamiento o deformación permanente remanente en la estructura tras un evento sísmico

Metamodelo Modelo sustituto (surrogate model) computacionalmente eficiente que aproxima la respuesta de un modelo de alta fidelidad

Physics-Guided Enfoque que incorpora leyes físicas mediante términos de penalización en la función de pérdida

Solver-Informed Enfoque que integra esquemas numéricos de solución (como Runge-Kutta) directamente en la arquitectura de la red neuronal

### **Símbolos griegos**

$\alpha$  Razón de rigidez post-fluencia a pre-fluencia (Modelo bilineal/Bouc-Wen)

$\beta, \gamma$  Parámetros del método de integración Newmark o del modelo Bouc-Wen

$\Gamma$  Vector de influencia sísmica

$\Phi_t$  Operador discreto de diferenciación temporal

$\theta$  Conjunto de parámetros entrenables de la red neuronal

$\Delta t$  Paso de tiempo de discretización

$\delta$  Error local en retropropagación (delta)

$\eta$  Tasa de aprendizaje (Learning rate)

$\omega_n$  Frecuencia natural vibratoria

$\sigma(\cdot)$  Función de activación no-lineal

$\varepsilon$  Medida del número de desviaciones estándar (epsilon) en CMS

$\zeta$  Razón de amortiguamiento crítico

### **Acrónimos y abreviaciones**

Adam Adaptive Moment Estimation (Optimizador)

AI Artificial Intelligence

ANN Artificial Neural Network (Red Neuronal Artificial)

BLWN Band-Limited White Noise (Ruido blanco limitado en banda)

CESMD Center for Engineering Strong Motion Data

DL Deep Learning

FEM Finite Element Method (Método de Elementos Finitos)

L2RE L2 Relative Error (Error Relativo Cuadrático)

L-BFGS Limited-memory Broyden–Fletcher–Goldfarb–Shanno (Optimizador)

LSTM Long Short-Term Memory (Red de Memoria a Corto y Largo Plazo)

MDOF Multiple Degrees of Freedom (Sistema de Múltiples Grados de Libertad)

MGDL Múltiples Grados de Libertad (Sigla en español)

ML Machine Learning

NMAE Normalized Mean Absolute Error

NRMSE Normalized Root Mean Square Error

NTHA Nonlinear Time-History Analysis (Análisis Tiempo-Historia No-lineal)

ODE Ordinary Differential Equation (Sigla en inglés para EDO)

PEER NGA Pacific Earthquake Engineering Research, Next Generation Attenuation Database

PhyCNN Physics-guided Convolutional Neural Network

PhyLSTM Physics-guided Long Short-Term Memory Network

PINN Physics-Informed Neural Network (Red Neuronal Informada por la Física)

PSD Power Spectral Density (Densidad Espectral de Potencia)

ResNet Residual Neural Network (Red Neuronal Residual)

RMSE Root Mean Square Error (Raíz del Error Cuadrático Medio)

RNA Red Neuronal Artificial (ANN en inglés)

RTHS Real-Time Hybrid Simulation (Simulación Híbrida en Tiempo Real)

SDOF Single Degree of Freedom (Sistema de un Grado de Libertad)

SHM Structural Health Monitoring

STFT Short-Time Fourier Transform (Transformada de Fourier de Tiempo Corto)

CNN Convolutional Neural Network (Red Neuronal Convolutiva)

EDO Ecuación Diferencial Ordinaria (ODE)

FPS Friction Pendulum System (Sistema de Péndulo Friccional)

GDL Grados de Libertad (DOF)

MLP Multilayer Perceptron (Perceptrón Multicapa)

MSE Mean Squared Error (Error Cuadrático Medio)

PGA Peak Ground Acceleration

ReLU Rectified Linear Unit

RK4 Runge-Kutta de 4to orden

RNN Recurrent Neural Network (Red Neuronal Recurrente)

SI Sistema Internacional de Unidades

# Capítulo 1

## Introducción

### 1.1. Contexto de la Investigación

Debido a que las edificaciones diseñadas bajo supuestos elásticos pueden sufrir deformaciones inelásticas severas en sismos extremos, resulta imprescindible un método capaz de simular su respuesta más allá del régimen elástico (Deierlein et al., 2010). El análisis tiempo-historia no-lineal (*Nonlinear Time-History Analysis*, NTHA) ofrece esta capacidad al calcular, paso a paso, la evolución de desplazamientos, aceleraciones y fuerzas internas durante todo el registro sísmico, incorporando explícitamente la degradación de rigidez y resistencia, fenómenos de plastificación y acumulación de daño. Esto permite evaluaciones estructurales bajo condiciones extremas con un alto grado de realismo, superando con creces las simplificaciones de los enfoques estáticos o lineales (Pinho, 2007; Gora et al., 2025).

En la práctica, la abundancia de datos generados por el NTHA ayuda a localizar zonas críticas y diseñar refuerzos preventivos (Núñez et al., 2021). No obstante, la simulación detallada mediante elementos finitos (FEM) exige mallas muy refinadas, modelos de materiales complejos (incluyendo plastificación, degradación y fractura) e integración de registros dinámicos de alta resolución temporal (Phuvoravan y Sotelino, 2005). Las estrategias de baja fidelidad mitigan el costo, pero pueden omitir fenómenos críticos debido a las incertidumbres del sistema (Liel et al., 2009). Aunque las técnicas de paralelización y optimización han reducido parcialmente los tiempos de cómputo, el elevado costo de múltiples NTHA, especialmente en estudios de confiabilidad estructural, sigue siendo un desafío, incluso con acceso a supercomputadoras (Pellissetti, 2009).

Este escenario se enmarca en la filosofía del diseño sísmico basado en desempeño, que fija objetivos estructurales explícitos para distintos niveles de amenaza y exige validar su cumplimiento mediante simulaciones no-lineales (Salgado y Guner, 2018). En la práctica, ello implica ejecutar decenas o cientos de NTHA con registros representativos de cada nivel de intensidad, de modo de evaluar estadísticamente la probabilidad de superar umbrales críticos. Aunque la mejora del hardware y los softwares de modelación han impulsado su adopción, las limitaciones computacionales siguen siendo un obstáculo para cumplir con eficiencia los requisitos normativos internacionales y las guías de evaluación de pérdidas (Vielma-Quintero et al., 2024).

Frente a esta tensión entre precisión y costo, han surgido enfoques alternativos. La simulación híbrida en tiempo real (Real-Time Hybrid Simulation, RTHS) combina el modelado numérico de parte de la estructura con ensayos físicos de componentes críticos, interactuando en tiempo real (Dong et al., 2022). Sin embargo, los requerimientos de tiempo real limitan la complejidad del submodelo numérico y las capacidades de los

actuadores, lo que puede afectar la estabilidad de las pruebas y restringir la demanda sísmica aplicable (Bas y Moustafa, 2020).

Una solución emergente consiste en entrenar metamodelos a partir de datos de ensayos físicos o simulaciones de alta fidelidad, para reproducir la respuesta no-lineal sin necesidad de FEM detallados en cada corrida (Bas y Moustafa, 2020; Dong et al., 2022). Las redes neuronales artificiales (RNAs) han demostrado su eficacia en diversos ámbitos de la ingeniería estructural desde hace años, aplicándose en tareas como la identificación de daños, la predicción de propiedades de materiales y la modelación del comportamiento estructural bajo cargas estáticas y dinámicas (Salehi y Burgueño, 2018; Vanluchene y Sun, 1990; Thai, 2022; Málaga-Chuquitaype, 2022; Sun et al., 2021). Estos enfoques aprovechan la capacidad de las RNAs para aproximar funciones complejas y detectar patrones en datos, ofreciendo modelos de sustitución (surrogate models) que evitan la necesidad de resolver directamente ecuaciones constitutivas o de equilibrio. Por ejemplo, Wu y Jahanshahi (2019) desarrollaron una red neuronal convolucional profunda capaz de estimar respuestas dinámicas de estructuras a partir de entradas de excitación, logrando capturar el comportamiento esencial sin integrar las ecuaciones de movimiento de forma explícita. De manera análoga, Zhang et al. (2019) propusieron un modelo basado en memorias recurrentes de largo y corto plazo (LSTM) para predecir la respuesta sísmica no-lineal de estructuras, aprovechando la habilidad de estas redes recurrentes para aprender la historia temporal de los eventos y representando efectivamente las no linealidades moderadas en el comportamiento histerético a partir de datos (Zhang et al., 2019). Modelos híbridos más recientes incorporan mecanismos de atención para mejorar la precisión de estas predicciones historia-temporal (Chen et al., 2023). En conjunto, estos trabajos iniciales basados en aprendizaje profundo confirman el potencial de las RNAs para emular la respuesta sísmica estructural de forma rápida; sin embargo, operan típicamente como cajas negras dependientes de datos, sin incorporar de manera explícita las leyes físicas subyacentes del sistema, lo que puede limitar su capacidad de extrapolar comportamientos fuera del rango de entrenamiento o bajo condiciones no observadas.

Ante esta limitación, han cobrado auge las llamadas redes neuronales informadas por la física (*Physics-Informed Neural Networks*, PINNs) como una evolución de los modelos de aprendizaje automático para problemas de ingeniería. Las PINNs integran conocimiento físico explícito dentro del proceso de entrenamiento de la red neuronal, incorporando en la función de pérdida los principios gobernantes del sistema (por ejemplo, las ecuaciones diferenciales de movimiento, condiciones de equilibrio o relaciones constitutivas) (Raissi et al., 2019; Karniadakis et al., 2021). Al penalizar las desviaciones de las leyes físicas durante el ajuste de la red, este enfoque produce soluciones que satisfacen inherentemente las ecuaciones del modelo, combinando la flexibilidad de las RNAs con la consistencia física de los métodos tradicionales (Farea et al., 2024). En el campo de la dinámica estructural, las PINNs se han introducido recientemente como una alternativa atractiva para resolver problemas directo e inversos, demostrando que pueden predecir la respuesta tiempo-historia de sistemas simples cumpliendo las ecuaciones de movimiento con buena precisión (Zhai et al., 2023; Su et al., 2024). Por ejemplo, los enfoques solver-informed han mostrado que es posible integrar la física directamente en la arquitectura y no solo en la pérdida: Guo et al. (2023) acoplan un integrador exponencial para la parte lineal con un flow-map residual aprendido para la no-linealidad, logrando trayectorias estables y buena generalización con “ventanas” autorregresivas paralelizables; de forma afín, Sadeghi Eshkevari et al. (2021) diseñan un bloque recurrente inspirado en esquemas implícitos de dinámica que mantiene el error acotado en

horizontes largos con menos parámetros; y en la familia Runge–Kutta, variantes recientes con compuertas GRU y refuerzo por gradientes mejoran la estabilidad temporal y la interpretabilidad del avance de estados incluso con instrumentación limitada en puentes reales (Zhai et al., 2024; Xing et al., 2024). En paralelo, las PINNs que incorporan la física en la función objetivo han probado que la red puede concentrarse en aprender solo la porción desconocida del modelo, reduciendo la dependencia a grandes volúmenes de datos: desde formulaciones physics-guided que penalizan el residuo de la ecuación de movimiento (Yu et al., 2020) y las variantes PhyCNN/PhyLSTM con variables latentes histeréticas (Zhang et al., 2020a; Zhang et al., 2020b), hasta PI-LSTM con mejoras sistemáticas frente a LSTM puras en SDOF y en un edificio experimental (Liu et al., 2023). En conjunto, estas aplicaciones evidencian que al informar la red con la física, ya sea embebiéndola en el solucionador o imponiéndola en la pérdida, se recorta la demanda de datos, se mejora la estabilidad a largo plazo y se preserva la fidelidad dentro y fuera del rango de entrenamiento.

No obstante, cabe señalar que la mayoría de los estudios previos con PINNs en ingeniería sísmica estructural se han enfocado principalmente en casos con no linealidades moderadas. En otras palabras, la mayoría de trabajos ha validado estas redes en estructuras cuyos daños inelásticos son limitados, sin alcanzar grandes deformaciones residuales ni extensas plastificaciones en sus elementos. Por ejemplo, buena parte de las validaciones se ha circunscrito a SDOF o MDOF de baja dimensión con histeresis idealizadas y estables, o a marcos que no alcanzan demandas cercanas al colapso: las formulaciones con pérdidas físicas tipo PhyLSTM y PI-LSTM se prueban sobre Bouc–Wen y edificios de ensayo con daño moderado (Zhang et al., 2020b; Liu et al., 2023); los integradores híbridos y solver-informed (p. ej., DynNet, PDHI) separan la parte lineal de la no-lineal aprendida y reportan trayectorias precisas, pero en configuraciones controladas y sin degradación acumulativa (Sadeghi Eshkevari et al., 2021; Guo et al., 2023); y las variantes Runge–Kutta embebidas (p. ej., RK4-PINN y su versión con refuerzo por gradiente) se validan en Duffing/SDOF o en puentes modelados numéricamente, con histeresis estable y sin pinching ni pérdida cíclica de rigidez (Zhai et al., 2023; Xing et al., 2024). Revisiones generales sobre PINNs enfatizan además retos pendientes, como la integración de leyes físicas complejas y el costo computacional del entrenamiento (Farea et al., 2024), lo que ayuda a explicar la escasez de estudios con plasticidad acumulativa intensa o escenarios cercanos al colapso. Precisamente, un indicador de desempeño crucial como el desplazamiento residual (o drift residual), que refleja deformaciones permanentes tras un sismo severo y está directamente vinculado a los costos de reparación y posibilidad de demolición de la estructura, no ha sido abordado en profundidad mediante modelos PINN hasta la fecha.

Algunos investigadores han comenzado a explorar arquitecturas de aprendizaje más avanzadas, como las redes tipo Transformer, para afrontar la predicción de respuestas estructurales bajo demandas altamente no-lineales y con drift residual. Estas redes, originarias del procesamiento de lenguaje natural, pueden modelar dependencias de largo alcance en secuencias temporales complejas y han sido aplicadas recientemente a la síntesis de historias de respuesta sísmica con gran fidelidad, incluso considerando degradaciones significativas (Chen et al., 2023; Zhang et al., 2024). No obstante, el entrenamiento de modelos basados en Transformers es sustancialmente costoso. A diferencia de redes convencionales o recurrentes, la autoatención estándar presenta complejidad temporal y de memoria cuadrática con la longitud de la secuencia, lo que se convierte en un cuello de botella para series largas; en la práctica, esto implica tiempos de entrenamiento elevados y la necesidad de hardware especializado para procesar lotes y contextos extensos (Zhang et al., 2023; Zhuang

et al., 2023). Además, aunque estos modelos capturan dependencias globales, no explotan de forma nativa las estructuras locales típicas de las series temporales y, por su complejidad y número de parámetros, requieren grandes volúmenes de datos para un entrenamiento eficaz (Thundiyil et al., 2023). En consecuencia, persiste el desafío de diseñar modelos eficientes que simulen con precisión la respuesta dinámica de estructuras altamente no-lineales sin incurrir en costos computacionales prohibitivos.

Motivados por este vacío, la presente investigación se centra en desarrollar y validar un modelo basado en PINN capaz de predecir la respuesta dinámica historia-temporal de estructuras sometidas a excitaciones sísmicas variables, capturando comportamientos con alta no-linealidad y poniendo especial énfasis en la estimación del drift residual. El modelo propuesto busca reproducir con fidelidad los grandes desplazamientos inelásticos y la plastificación acumulada que pueden presentarse bajo sismos severos, manteniendo a la vez un costo computacional reducido frente a simulaciones no-lineales tradicionales o enfoques de aprendizaje profundo puramente data-driven. Con ello, este trabajo pretende llenar el vacío identificado en la literatura y aportar una herramienta innovadora para la evaluación sísmica de estructuras que experimentan daños extremos. En este marco, se han definido objetivos específicos que orientan el desarrollo, implementación y validación del modelo, los cuales se presentan en la siguiente sección.

## 1.2. Objetivos

A partir del análisis del estado del arte y de las limitaciones detectadas en los enfoques actuales, en particular, la escasez de estudios que traten explícitamente el *drift* residual en estructuras histeréticas con degradación significativa de rigidez, se establecen a continuación los objetivos que guiarán el desarrollo de la presente investigación, organizados en un objetivo general y un conjunto de objetivos específicos.

### 1.2.1. Objetivo General

Desarrollar y evaluar un modelo de redes neuronales informadas por la física (PINN) capaz de predecir, con un alto grado de precisión, la respuesta dinámica historia-temporal de estructuras histeréticas con degradación de rigidez ante excitaciones sísmicas, con énfasis en la estimación del drift residual.

### 1.2.2. Objetivos Específicos

1. Conformar y preparar un conjunto de datos sísmicos con registros de alta densidad temporal, tanto artificiales como reales, y recopilar la respuesta estructural correspondiente (desplazamiento, velocidad, aceleración y fuerzas internas), para disponer de una base robusta para el entrenamiento, validación y test de los modelos.
2. Evaluar modelos PINN de referencia representativos de formulaciones physics-guided para identificar configuraciones con mejor desempeño y sus limitaciones ante secuencias largas y no-linealidad intensa.
3. Proponer un modelo PINN con el fin de mejorar la precisión y robustez en comparación con modelos existentes, asegurando su generalización a través de datos no vistos durante el entrenamiento.

### 1.3. Estructura de la Tesis

El presente documento se organiza en cinco capítulos que articulan el desarrollo de la investigación, desde la fundamentación teórica hasta la validación experimental de la arquitectura propuesta. Adicionalmente, el trabajo incluye una sección de nomenclatura al inicio y apéndices técnicos al final para facilitar la consulta de términos, símbolos y códigos fuente, garantizando así la reproducibilidad del estudio.

El **Capítulo 1** establece el contexto y la motivación del estudio, delimitando el problema de la predicción de respuesta sísmica en estructuras con comportamiento no lineal severo. En esta sección se resumen las brechas detectadas en el estado del arte, se definen los objetivos generales y específicos que guían la investigación, y se presentan los alcances y limitaciones del trabajo.

Posteriormente, el **Capítulo 2** sistematiza los fundamentos conceptuales necesarios para el desarrollo de la tesis. Se abordan la dinámica estructural no lineal, los modelos histeréticos con degradación de rigidez (Bouc-Wen) y el costo computacional asociado al análisis tiempo–historia (NTHA). Asimismo, se introduce el paradigma de las Redes Neuronales Informadas por la Física (PINNs), estableciendo una distinción taxonómica crítica entre los enfoques con física en la función de pérdida (*guided-informed*) y los enfoques integrados en la arquitectura (*solver-informed*), definiendo además las métricas de evaluación utilizadas transversalmente en el estudio.

A continuación, el **Capítulo 3** evalúa críticamente el desempeño de las arquitecturas de referencia en la literatura *guided-informed*, específicamente PhyCNN y PhyLSTM, bajo escenarios de alta demanda sísmica. Se describe la metodología experimental, que incluye la generación de conjuntos de datos con sismos artificiales y registros chilenos reales, y se diagnostican las limitaciones de estabilidad de estos modelos, poniendo especial énfasis en su incapacidad para predecir correctamente el *drift* residual y la acumulación de errores en secuencias de larga duración.

La contribución central de la tesis se presenta en el **Capítulo 4**, donde se desarrolla la arquitectura **RK4Net**. Se justifica el cambio de paradigma hacia un enfoque *solver-informed* que integra un esquema numérico de Runge-Kutta de cuarto orden directamente en el grafo computacional. Se detalla la estrategia de entrenamiento curricular (*Warm Start*) diseñada para superar la paradoja de la complejidad y estabilizar la optimización, demostrando finalmente cómo esta propuesta mitiga el problema del *drift* numérico identificado en el capítulo precedente.

Finalmente, el **Capítulo 5** sintetiza los hallazgos principales y discute las implicancias de adoptar integradores neuronales profundos para la evaluación sísmica rápida. El documento cierra delineando las perspectivas de investigación futura, incluyendo la extensión a sistemas de múltiples grados de libertad (MDOF) y la incorporación de incertidumbre en los parámetros físicos.

# Capítulo 2

## Marco Teórico

### 2.1. Notación, convenciones y supuestos

En esta sección se fijan las convenciones de notación, las hipótesis estructurales básicas y el tratamiento del dominio temporal que se emplearán a lo largo de la tesis. El propósito es asegurar consistencia formal entre el desarrollo teórico, la implementación de modelos PINN y la validación numérica.

#### 2.1.1. Notación y símbolos

**Escalares, vectores y matrices:** Se emplea tipografía en *romana* para escalares ( $a, b, \dots$ ); negrita *minúscula* para vectores ( $\mathbf{u}, \mathbf{v}, \dots$ ) y negrita *mayúscula* para matrices ( $\mathbf{M}, \mathbf{K}, \mathbf{C}$ ). El  $i$ -ésimo componente de  $\mathbf{u}$  se denota por  $u_i$  y el elemento  $(i, j)$  de  $\mathbf{A}$  por  $A_{ij}$ .  $\mathbf{0}$  y  $\mathbf{I}$  denotan el vector y la matriz identidad de tamaño compatible, respectivamente. La traspuesta se indica con  $(\cdot)^T$  y la inversa con  $(\cdot)^{-1}$  cuando existe.

**Operadores y normas:** El producto interno euclídeo entre vectores  $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$  es  $\langle \mathbf{u}, \mathbf{v} \rangle := \mathbf{u}^T \mathbf{v}$ . La norma euclídea es  $\|\mathbf{u}\|_2 := \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle}$ . Para cualquier vector  $\mathbf{u} \in \mathbb{R}^n$  y un  $p \geq 1$  se define la **norma-p** como:  $\|\mathbf{u}\|_p := \sqrt[p]{|u_1|^p + |u_2|^p + \dots + |u_n|^p}$ . Así para el caso  $p = 1$  se obtiene  $\|\mathbf{u}\|_1 := |u_1| + |u_2| + \dots + |u_n|$ , y para el caso  $p = 2$  se obtiene la norma euclídea explicada más arriba. El producto de Hadamard es  $\mathbf{A} \circ \mathbf{B}$  donde  $(\mathbf{A} \circ \mathbf{B})_{ij} := A_{ij} \cdot B_{ij}$ .

**Derivadas y operadores diferenciales:** El punto sobre una variable denota derivación temporal:  $\dot{\mathbf{u}} := d\mathbf{u}/dt$ ,  $\ddot{\mathbf{u}} := d^2\mathbf{u}/dt^2$ . El operador gradiente respecto de  $\mathbf{u}$  se indica por  $\nabla_{\mathbf{u}}(\cdot)$ ; para funciones vectoriales,  $\mathbf{J} = \partial \mathbf{f} / \partial \mathbf{u}$  denota la matriz jacobiana. Cuando sea necesario distinguir derivadas parciales, se utilizará  $\partial(\cdot) / \partial(\cdot)$ .

**Conjuntos e índices:**  $\mathcal{T} = [0, T]$  indica el intervalo temporal del análisis.  $\mathcal{E}$  denota el conjunto de eventos sísmicos (registros); los subíndices *tr*, *val* y *te* designan particiones de entrenamiento, validación y prueba sin solapamiento por evento, cada registro completo pertenece exclusivamente a una de las particiones. Adicionalmente, se considera el subconjunto  $\mathcal{E}_{\text{col}}$  de registros para colocación, compuesto por registros que pueden ser utilizados como entradas por los modelos para ajustar términos físicos o evaluar su robustez, pero cuya respuesta de referencia no se encuentra disponible durante el entrenamiento. Estos registros suelen contener eventos adicionales a los de entrenamiento y pueden provenir de los conjuntos de prueba y/o validación, sin que su inclusión afecte la independencia de los subconjuntos utilizados para la evaluación del desempeño.

**Unidades y normalización:** Se adopta el **SI**: longitud en metros (m), tiempo en segundos (s), masa en kilogramos (kg), fuerza en newtons (N), aceleración en  $\text{m/s}^2$ . Las aceleraciones sísmicas podrán reportarse normalizadas por  $g_0 = 9.80665 \text{ m/s}^2$ , es decir,  $\tilde{a}_g(t) = a_g(t)/g_0$ . Para el preprocesamiento de entradas/salidas en el entrenamiento, se empleará normalización consciente de la física cuando proceda (p. ej., desplazamientos escalados por el desplazamiento de fluencia  $u_y$ ), o estandarización tipo  $z$ -score por variable cuando se requiera homogeneizar magnitudes en la función de pérdida. En casos donde se desee restringir las magnitudes a un rango específico, se podrá emplear normalización min–max por variable, es decir,  $\tilde{\mathbf{x}} = \frac{\mathbf{x} - x_{\min}}{x_{\max} - x_{\min}}$ , para reescalar a un intervalo  $[0, 1]$  u otro rango definido según la naturaleza del problema. La no dimensionalización se indicará con tilde:  $\tilde{\mathbf{u}} = \mathbf{u}/u_\star$ ,  $\tilde{\mathbf{a}}_g = \mathbf{a}_g/A_{g\star}$ , etc., siendo  $u_\star$  y  $A_{g\star}$  escalas de referencia coherentes con el problema. Las elecciones concretas de escalas, cuando afecten a la ponderación de términos físicos en la pérdida, se documentarán explícitamente en los capítulos de implementación.

### 2.1.2. Supuestos estructurales

**Modelo de referencia (MDOF):** Se considera una estructura discretizada con  $n$  grados de libertad (GDL), con desplazamientos relativos  $\mathbf{u}(t) \in \mathbb{R}^n$  respecto de la base, velocidad  $\dot{\mathbf{u}}(t)$  y aceleración  $\ddot{\mathbf{u}}(t)$ . El vector de variables internas histeréticas (p. ej., distorsiones plásticas, parámetros de Bouc–Wen u otras) se denota por  $\mathbf{z}(t) \in \mathbb{R}^m$ . La ecuación de movimiento se escribe en forma general:

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{f}_{\text{int}}(\mathbf{u}(t), \dot{\mathbf{u}}(t), \mathbf{z}(t); \boldsymbol{\theta}) = \mathbf{p}(t) - \mathbf{M}\boldsymbol{\Gamma} a_g(t), \quad (2.1)$$

donde  $\mathbf{M} \in \mathbb{R}^{n \times n}$  es la matriz de masas,  $\mathbf{C} \in \mathbb{R}^{n \times n}$  la matriz de amortiguamiento (posiblemente dependiente del estado),  $\mathbf{f}_{\text{int}}$  las fuerzas internas (lineales y no-lineales) parametrizadas por  $\boldsymbol{\theta}$ ,  $\mathbf{p}(t)$  fuerzas nodales equivalentes y  $a_g(t)$  la aceleración de base (escalar) aplicada mediante el vector de influencia  $\boldsymbol{\Gamma} \in \mathbb{R}^n$  (Chopra, 2017).

**Linealidad geométrica y no-linealidad material:** Salvo indicación en contrario, se asume **linealidad geométrica** (pequeñas deformaciones y rotaciones), permitiendo **no-linealidad material** a través de  $\mathbf{f}_{\text{int}}$  y de la evolución de  $\mathbf{z}$ . La dinámica de variables internas se modela como

$$\dot{\mathbf{z}}(t) = \mathbf{g}(\mathbf{u}(t), \dot{\mathbf{u}}(t), \mathbf{z}(t); \boldsymbol{\theta}), \quad (2.2)$$

lo que captura memoria histerética, umbrales de fluencia y potencial *degradación* de rigidez y/o resistencia bajo carga cíclica (Baber y Wen, 1981).

**Amortiguamiento:** Se adopta amortiguamiento viscoso efectivo; en régimen lineal puede representarse como amortiguamiento proporcional (p. ej., Rayleigh  $\mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K}$ ). En régimen no-lineal se permite dependencia del estado (p. ej.,  $\mathbf{C}(\mathbf{u}, \dot{\mathbf{u}}, \mathbf{z})$ ) o amortiguamiento histerético implícito en  $\mathbf{f}_{\text{int}}$  (Charney, 2008). Las elecciones específicas para cada caso de estudio se declaran en los capítulos de implementación.

**Entrada sísmica y fuerzas equivalentes:** La excitación base se modela por  $a_g(t)$ , de modo que el término inercial equivalente es  $-\mathbf{M}\Gamma a_g(t)$ . Alternativamente, cuando convenga, se puede trabajar con fuerzas externas equivalentes  $\mathbf{p}_{\text{eq}}(t) := \mathbf{p}(t) - \mathbf{M}\Gamma a_g(t)$ , manteniendo (2.1) con el lado derecho agrupado; sin embargo, en la mayoría de los casos se considera  $\mathbf{p}(t) = 0$ , salvo que se indique lo contrario.

**Condiciones iniciales y contorno:** Salvo que se indique lo contrario, se consideran condiciones iniciales de reposo:  $\mathbf{u}(0) = \mathbf{0}$ ,  $\dot{\mathbf{u}}(0) = \mathbf{0}$ , y  $\mathbf{z}(0) = \mathbf{z}_0$  (usualmente nula).

### 2.1.3. Dominio temporal y discretización

**Tiempo continuo y muestreo:** Las magnitudes dinámicas se definen en tiempo continuo  $t \in \mathcal{T} = [0, T]$  y se observan/discretizan en instantes uniformemente espaciados  $t_k = k \Delta t$  con  $k = 0, 1, \dots, N$ ,  $\Delta t > 0$  y  $T = N \Delta t$ .

**Notación discreta:** Para cualquier señal  $x(t)$ , su versión muestreada es  $x_k := x(t_k)$ . De forma análoga,  $\mathbf{u}_k := \mathbf{u}(t_k)$ ,  $\dot{\mathbf{u}}_k := \dot{\mathbf{u}}(t_k)$ ,  $\ddot{\mathbf{u}}_k := \ddot{\mathbf{u}}(t_k)$ ,  $\mathbf{z}_k := \mathbf{z}(t_k)$  y  $a_{g,k} := a_g(t_k)$ . Cuando resulte más conveniente, se empleará la notación en bloque  $\mathbf{U}_{0:N} := \{\mathbf{u}_0, \dots, \mathbf{u}_N\}$  para designar trayectorias completas. En esta notación, el subíndice  $k$  indica el instante de muestreo  $t_k$  y  $\mathbf{u}_k$  representa el vector completo de desplazamientos en dicho instante (de todos los gdl). Si se requiere hacer referencia a un grado de libertad específico  $i$  en el instante  $t_k$ , se empleará  $u_{i,k}$ , que es un componente escalar.

**Alineación de señales:** Las señales de entrada/salida (p. ej.,  $a_g$ ,  $\mathbf{u}$ ,  $\dot{\mathbf{u}}$ ,  $\ddot{\mathbf{u}}$ , fuerzas internas) se almacenarán temporalmente alineadas en cada evento. Cualquier procesamiento a los datos originales, como corrección de línea base, filtrado o remuestreo, se declarará en el capítulo de preparación de datos para asegurar replicabilidad.

## 2.2. Fundamentos de dinámica estructural no-lineal

Esta sección fija la formulación matricial de referencia para sistemas con múltiples grados de libertad (MDOF) sometidos a excitación sísmica e introduce su representación en espacio de estados.

### 2.2.1. Ecuaciones de movimiento (MDOF) con entrada sísmica

Sea una estructura discretizada con  $n$  grados de libertad (GDL), con desplazamientos relativos respecto de la base  $\mathbf{u}(t) \in \mathbb{R}^n$ , velocidad  $\dot{\mathbf{u}}(t)$  y aceleración  $\ddot{\mathbf{u}}(t)$ . La ecuación de movimiento (EDM) en coordenadas relativas adopta la forma general

$$\mathbf{M} \ddot{\mathbf{u}}(t) + \mathbf{C}(\mathbf{u}, \dot{\mathbf{u}}, \mathbf{z}) \dot{\mathbf{u}}(t) + \mathbf{f}_{\text{int}}(\mathbf{u}(t), \dot{\mathbf{u}}(t), \mathbf{z}(t); \boldsymbol{\theta}) = \mathbf{p}(t) - \mathbf{M}\Gamma a_g(t), \quad (2.3)$$

donde  $\mathbf{M} \in \mathbb{R}^{n \times n}$  es la matriz de masas (simétrica definida positiva),  $\mathbf{C}$  representa el amortiguamiento efectivo (posiblemente dependiente del estado),  $\mathbf{f}_{\text{int}}$  son las fuerzas internas (lineales/no-lineales) parametrizadas por  $\boldsymbol{\theta}$ ,

$\mathbf{p}(t)$  son fuerzas nodales equivalentes (cargas aplicadas),  $a_g(t)$  es la aceleración de base (escalar) y  $\mathbf{\Gamma} \in \mathbb{R}^n$  es el vector de influencia que mapea la excitación basal a coordenadas generalizadas.(Chopra, 2017)<sup>1</sup>

En régimen puramente lineal,  $\mathbf{f}_{\text{int}}(\mathbf{u}, \dot{\mathbf{u}}, \mathbf{z}) \equiv \mathbf{K}\mathbf{u}$  y la matriz de amortiguamiento suele modelarse como proporcional (p. ej., Rayleigh  $\mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K}$ ).(Chopra, 2017)

En régimen no-lineal,  $\mathbf{f}_{\text{int}}$  incluye la contribución elástica y la histerética, y su dependencia de variables internas  $\mathbf{z}(t) \in \mathbb{R}^m$  captura memoria y degradación (p. ej., modelos diferenciales tipo Bouc–Wen (Bouc, 1971; Wen, 1976), modelos por tramos basados en reglas de fluencia como IMK (Ibarra et al., 2005) y modelos basados en operadores Preisach (Mayergoyz, 1991)). La cinemática de  $\mathbf{z}$  se describe mediante

$$\dot{\mathbf{z}}(t) = \mathbf{g}(\mathbf{u}(t), \dot{\mathbf{u}}(t), \mathbf{z}(t); \boldsymbol{\theta}), \quad (2.4)$$

con  $\mathbf{g}(\cdot)$  definida por el modelo constitutivo adoptado (p. ej., reglas de fluencia por tramos tipo IMK (Ibarra et al., 2005) o variantes de Bouc–Wen con degradación (Baber y Wen, 1981), así como formulaciones basadas en operadores (Mayergoyz, 1991)).

Para notación compacta, definimos el *residuo dinámico* de (2.3):

$$\mathbf{r}(\ddot{\mathbf{u}}, \dot{\mathbf{u}}, \mathbf{u}, \mathbf{z}; t) := \mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{f}_{\text{int}}(\mathbf{u}, \dot{\mathbf{u}}, \mathbf{z}) - \mathbf{p}(t) + \mathbf{M}\mathbf{\Gamma}a_g(t), \quad \mathbf{r} \equiv \mathbf{0}. \quad (2.5)$$

Esta forma es la que se empleará, más adelante, para definir términos de pérdida física en los modelos informados por la física (PINNs), ya sea tal cual o en versiones algebraicamente equivalentes (p. ej., normalizadas por  $\mathbf{M}$ ).

### 2.2.2. Formulación en espacio de estados

Una representación de espacios de estados es un modelo matemático de un sistema físico descrito mediante un conjunto de entradas, salidas y variables de estado relacionadas por ecuaciones diferenciales. Con una representación en espacio de estados se puede reescribir la EDM de segundo orden (2.3) como un sistema de primer orden, lo que facilita el análisis, la identificación y el diseño de estimadores y controladores.(Craig y Kurdila, 2006)

**Representación general.** Una forma general de representación en espacio de estados para un sistema con  $p$  entradas,  $q$  salidas y  $n$  variables de estado es

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t), \quad \mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{r}(t), \quad (2.6)$$

donde:

$\mathbf{x}(t) \in \mathbb{R}^n$  Vector de estado (variables mínimas que confieren propiedad de Markov<sup>2</sup>).

$\mathbf{r}(t) \in \mathbb{R}^p$  Vector de entradas o excitaciones externas

<sup>1</sup>En ausencia de cargas nodales se usa  $\mathbf{p}(t) \equiv \mathbf{0}$  y el lado derecho de (2.3) queda reducido a  $-\mathbf{M}\mathbf{\Gamma}a_g(t)$ .

<sup>2</sup>La propiedad de Markov establece que la evolución futura de un proceso estocástico depende únicamente de su estado actual y no de su historia pasada.

$\mathbf{y}(t) \in \mathbb{R}^q$  Vector de salidas o medidas (sensores/variables de interés)

$\mathbf{A}(t) \in \mathbb{R}^{n \times n}$  Matriz de dinámica interna

$\mathbf{B}(t) \in \mathbb{R}^{n \times p}$  Matriz de acoplamiento de entradas

$\mathbf{C}(t) \in \mathbb{R}^{q \times n}$  Matriz de mapeo de estado a salidas

$\mathbf{D}(t) \in \mathbb{R}^{q \times p}$  Matriz de transmisión directa (efecto instantáneo de  $\mathbf{r}$  en  $\mathbf{y}$ ).

En *sistemas lineales invariantes en el tiempo* (LTI) las matrices son constantes; en sistemas *LTV* son explícitamente dependientes de  $t$  (o de parámetros programables)(Craig y Kurdila, 2006).

**Caso lineal (MDOF en coordenadas relativas).** Para la EDM (2.3) en régimen lineal, con  $\mathbf{f}_{\text{int}} = \mathbf{K}\mathbf{u}$  y amortiguamiento constante, se adoptan:

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{u}(t) \\ \dot{\mathbf{u}}(t) \end{bmatrix} \in \mathbb{R}^{2n}, \quad \mathbf{r}(t) = \begin{bmatrix} \mathbf{p}(t) \\ a_g(t) \end{bmatrix} \in \mathbb{R}^{n+1}.$$

Con esta elección, (2.6) toma la forma

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{r}, \quad \mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{r}, \quad (2.7)$$

donde

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \underbrace{\mathbf{M}^{-1}}_{\text{asociada a } \mathbf{p}(t)} & \underbrace{-\mathbf{\Gamma}}_{\text{asociada a } a_g(t)} \end{bmatrix}$$

Las salidas se configuran según las variables de interés:

- *Desplazamientos relativos:*  $\mathbf{y}_u = \mathbf{u} \Rightarrow \mathbf{C} = [\mathbf{1} \ \mathbf{0}]$ ,  $\mathbf{D} = \mathbf{0}$ .
- *Aceleraciones absolutas:*  $\ddot{\mathbf{q}} = \ddot{\mathbf{u}} + \mathbf{\Gamma}a_g = -\mathbf{M}^{-1}\mathbf{K}\mathbf{u} - \mathbf{M}^{-1}\mathbf{C}\dot{\mathbf{u}} + \mathbf{M}^{-1}\mathbf{p}$ , de modo que  $\mathbf{C} = [-\mathbf{M}^{-1}\mathbf{K} \ -\mathbf{M}^{-1}\mathbf{C}]$ ,  $\mathbf{D} = [\mathbf{M}^{-1} \ \mathbf{0}]$ .

**Extensión no-lineal con variables internas.** Cuando la respuesta incluye efectos histeréticos/degradativos, se introducen variables internas  $\mathbf{z}(t) \in \mathbb{R}^m$  (p. ej., Bouc–Wen, reglas por tramos, operadores). Ahora se define el estado aumentado

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{u}(t) \\ \dot{\mathbf{u}}(t) \\ \mathbf{z}(t) \end{bmatrix} \in \mathbb{R}^{2n+m}$$

conduce a la forma no-lineal

$$\dot{\mathbf{x}} = \mathbf{\Psi}(\mathbf{x}; \boldsymbol{\theta}) + \mathbf{B}_r \mathbf{r}, \quad \mathbf{y} = \mathbf{h}(\mathbf{x}, \mathbf{r}), \quad (2.8)$$

con

$$\Psi(\mathbf{x}) = \begin{bmatrix} \dot{\mathbf{u}} \\ \mathbf{M}^{-1}(-\mathbf{C}(\mathbf{u}, \dot{\mathbf{u}}, \mathbf{z})\dot{\mathbf{u}} - \mathbf{f}_{\text{int}}(\mathbf{u}, \dot{\mathbf{u}}, \mathbf{z}; \boldsymbol{\theta})) \\ \mathbf{g}(\mathbf{u}, \dot{\mathbf{u}}, \mathbf{z}; \boldsymbol{\theta}) \end{bmatrix}, \quad \mathbf{B}_r = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{M}^{-1} & -\boldsymbol{\Gamma} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

Aquí  $\mathbf{g}$  describe la cinemática de las variables internas (memoria/degradación) y  $\mathbf{h}$  el mapeo a salidas.

**Realización equivalente con subsistema histerético.** En la práctica, la variable histerética puede *generarse* en un subsistema dinámico que integra  $\dot{\mathbf{z}} = \mathbf{g}(\mathbf{u}, \dot{\mathbf{u}}, \mathbf{z}; \boldsymbol{\theta})$  y cuya salida  $\mathbf{z}(t)$  se realimenta como una de las entradas de (2.7) (junto con  $a_g$  y/o  $\mathbf{p}$ ). Si  $\mathbf{z}(t)$  no es exógena sino el resultado de esa ODE, la interconexión resultante es matemáticamente **equivalente** al modelo con estado aumentado (2.8): el estado total del sistema cerrado es  $\mathbf{x}(t) = [\mathbf{u}^\top \dot{\mathbf{u}}^\top \mathbf{z}^\top]^\top$ , se preserva la causalidad y se mantiene la consistencia energética. Esta equivalencia justifica describir  $\mathbf{z}$  ya sea como *estado interno* o como señal producida por un *subsistema dinámico* realimentado, según convenga a la implementación.

## 2.3. Análisis tiempo–historia no-lineal (NTHA)

El *análisis tiempo–historia no-lineal* (NTHA) consiste en resolver la EDM (2.3) a lo largo de un registro sísmico, permitiendo materiales y dispositivos con respuesta dependiente del estado (histerética y/o degradativa). La ejecución práctica del NTHA descansa en tres pilares: (i) la discretización espacial por elementos finitos y el *ensamblaje* global; (ii) la *integración temporal* de las ecuaciones de evolución; y (iii) la *imposición iterativa del equilibrio* en cada paso de tiempo mediante métodos de Newton con una rigidez (tangente) coherente con el modelo constitutivo y el esquema temporal. A continuación se presentan estos componentes de forma operativa, en coherencia con la notación establecida en §2.1–2.2.

### 2.3.1. Discretización espacial (MEF) y ensamblaje

**Modelo discreto.** La estructura continua se representa por un conjunto de elementos finitos conectados en nodos. Cada nodo aporta grados de libertad (GDL) de desplazamiento/rotación según el tipo de elemento (barras, vigas, placas, cascarones, resortes no-lineales, aisladores, etc.). La *cinemática de elemento* define deformaciones a partir de desplazamientos nodales y, vía el modelo constitutivo adoptado, produce fuerzas internas elementales.

**Contribuciones de elemento.** En cada elemento  $e$  se evalúan: (i) la *fuerza interna elemental*  $\mathbf{f}_{\text{int}}^e$  en sus GDL locales (función de las deformaciones y de variables internas  $\mathbf{z}^e$ ); (ii) la *matriz de masa*  $\mathbf{M}^e$  (consistente o lumped, según el caso); (iii) la *contribución de amortiguamiento*  $\mathbf{C}^e$  (p. ej., proporcional o dependiente del estado). Para elementos no-lineales  $\mathbf{f}_{\text{int}}^e$  y la *rigidez tangente elemental*  $\mathbf{K}_{\text{tan}}^e$  dependen del punto de operación (deformaciones/variables internas) alcanzado en el paso de tiempo.

**Ensamblaje global.** Con la matriz de conectividad se mapean magnitudes locales a coordenadas globales. El *ensamblaje* consiste en acumular por superposición las contribuciones elementales en las matrices y vectores globales:

$$\mathbf{M} = \sum_e \mathbf{A}_e^T \mathbf{M}^e \mathbf{A}_e, \quad \mathbf{C} = \sum_e \mathbf{A}_e^T \mathbf{C}^e \mathbf{A}_e, \quad \mathbf{f}_{\text{int}}(\cdot) = \sum_e \mathbf{A}_e^T \mathbf{f}_{\text{int}}^e(\cdot).$$

Aquí  $\mathbf{A}_e$  recoge los GDL globales activos del elemento  $e$ . Las restricciones cinemáticas (apoyos, diafragmas rígidos, vínculos multi–punto) se imponen mediante reducción de GDL, multiplicadores o penalización, dejando una EDM en los GDL *libres*. La excitación basal se incorpora como  $-\mathbf{M}\Gamma a_g(t)$  o, alternativamente, como cargas nodales equivalentes (cf. §2.1–2.2).

La elección *masa consistente* mejora exactitud modal; la *masa lumped* puede ser ventajosa en problemas muy grandes o con integradores explícitos. En presencia de no-linealidad marcada, conviene que los elementos entreguen no sólo  $\mathbf{f}_{\text{int}}^e$  sino también su *tangente consistente*  $\mathbf{K}_{\text{tan}}^e$  para acelerar la convergencia del equilibrio global (véase §2.3.3).

### 2.3.2. Integración temporal

Dado un particionado temporal  $t_k = k \Delta t$ , el objetivo es avanzar el estado desde  $t_k$  hasta  $t_{k+1}$  haciendo compatibles: (i) la EDM (2.3), (ii) la evolución de variables internas (2.4), y (iii) la consistencia con las condiciones iniciales. En el régimen no-lineal, la integración temporal suele ser *implícita* para garantizar estabilidad y robustez frente a rigideces efectivas altas; el equilibrio en  $t_{k+1}$  se satisface iterativamente.

**Esquemas explícitos vs. implícitos** Los esquemas **explícitos** actualizan el estado en  $t_{k+1}$  únicamente con información conocida en  $t_k$  (y en  $t_{k-1}$ ), por lo que no requieren resolver sistemas no-lineales en cada paso (p.,ej., diferencias centrales, Runge–Kutta explícitos). En cambio, en los **implícitos** el estado en  $t_{k+1}$  aparece dentro de las ecuaciones, por lo que en cada paso se plantea un problema (lineal o no-lineal) que típicamente se resuelve con Newton–Raphson.

En términos de estabilidad y costo, los métodos **explícitos** son *condicionalmente estables*; en consecuencia,  $\Delta t$  debe respetar un límite controlado por la frecuencia natural máxima (para diferencias centrales lineales no amortiguadas,  $\Delta t \lesssim 2/\omega_{\text{máx}}$ ; con amortiguamiento o no-linealidad el límite suele ser más estricto). Tienen un costo por paso muy bajo y resultan convenientes cuando de por sí se requiere un  $\Delta t$  pequeño. Además, los métodos explícitos clásicos (p.,ej., diferencias centrales) no introducen amortiguamiento numérico en alta frecuencia, lo que puede amplificar el ruido si  $\Delta t$  no es suficientemente pequeño.

Por su parte, los métodos **implícitos** suelen ser *A-estables* (o de estabilidad amplia), permitiendo pasos mayores sin perder robustez en sistemas rígidos e histeréticos. Aunque cada paso resulta más costoso (pues implica resolver sistemas de ecuaciones y, en no-lineales, iterar hasta el equilibrio), los métodos implícitos permiten usar pasos de tiempo mayores sin perder estabilidad. Por ello, para alcanzar la misma precisión global, suelen necesitar menos pasos totales en estructuras rígidas o con comportamiento no-lineal complejo. Muchos implícitos (p.,ej., Newmark de aceleración promedio) conservan bien la energía en el rango de interés sin añadir amortiguamiento artificial; extensiones como HHT o generalized- $\alpha$  permiten *amortiguamiento numérico controlado* de altas frecuencias sin contaminar la respuesta.

En NTHA con histéresis/degradación y tamaños de modelo típicos de edificaciones, los esquemas **implícitos** son el estándar por su robustez (con Newton y tangente consistente). Los **explícitos** resultan atractivos cuando el propio modelo impone un  $\Delta t$  reducido y se dispone de masa diagonalizada.

**Diferencias centrales y variantes (explícitos de segundo orden).** El esquema de *diferencias centrales* (DC) actualiza el desplazamiento sin resolver sistemas por paso. Para el sistema

$$M \ddot{\mathbf{u}} + C \dot{\mathbf{u}} + K \mathbf{u} = \mathbf{p}(t),$$

se computa la aceleración en  $t_k$  desde el equilibrio

$$M \mathbf{a}_k = \mathbf{p}_k - C \mathbf{v}_k - K \mathbf{u}_k,$$

y se avanza con

$$\mathbf{u}_{k+1} = 2 \mathbf{u}_k - \mathbf{u}_{k-1} + \Delta t^2 \mathbf{a}_k, \quad \mathbf{v}_k \approx \frac{\mathbf{u}_k - \mathbf{u}_{k-1}}{\Delta t}.$$

El arranque se realiza con una aproximación de  $\mathbf{u}_{-1}$  (p. ej., vía expansión de Taylor) u otra forma de mejorar la aproximación en los extremos es que en  $t_0$  y/o al final de la integración, se usan *métodos unilaterales* (adelante/atrás) para aproximar derivadas cuando no hay datos a ambos lados:

$$\dot{\mathbf{u}}_0 \approx \frac{-3\mathbf{u}_0 + 4\mathbf{u}_1 - \mathbf{u}_2}{2\Delta t} \quad (\text{adelante, 2}^\circ \text{ orden}), \quad \dot{\mathbf{u}}_0 \approx \frac{\mathbf{u}_1 - \mathbf{u}_0}{\Delta t} \quad (\text{adelante, 1}^\circ \text{ orden}).$$

Análogos atrás se emplean cerca de  $t_N$ . Estas variantes son útiles *sólo* en los primeros/últimos pasos: las fórmulas de 2º orden mantienen la precisión global; las de 1º orden introducen más dispersión/difusión numérica.

Para este metodo de integración el costo por paso es muy bajo siempre que  $M$  sea diagonal (masa *lumped*), pues  $\mathbf{a}_k$  se obtiene con una sola “inversa” componente a componente. Con masa consistente, el método pierde su carácter estrictamente explícito. Tal como se mencionó antes DC es *condicionalmente estable*; para el caso lineal no amortiguado, el límite clásico es

$$\Delta t \lesssim \frac{2}{\omega_{\text{máx}}},$$

donde  $\omega_{\text{máx}}$  es la mayor frecuencia natural del modelo. Amortiguamiento, no-linealidades fuertes o rigideces locales muy altas (p. ej., contactos) imponen límites más estrictos. Por otro lado DC es de segundo orden en tiempo y no introduce amortiguamiento numérico en altas frecuencias. Esto preserva la amplitud de modos altos pero también puede amplificar ruido si  $\Delta t$  no es suficientemente pequeño.

DC y sus variantes son especialmente atractivos cuando el propio problema exige  $\Delta t$  pequeño (impacto, propagación de ondas, elementos muy esbeltos con malla fina) y se dispone de masa diagonal. En modelos de edificación típicos, donde la banda de interés es baja y hay histéresis/degradación, el costo total suele ser mayor que con implícitos por la severa restricción en  $\Delta t$ .

**Runge–Kutta y el esquema clásico de cuarto orden (RK4).** *Runge–Kutta* designa una familia de métodos de un paso para ecuaciones diferenciales ordinarias (EDO) de la forma

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t)),$$

que aproximan la integral de evolución mediante combinaciones ponderadas de pendientes evaluadas dentro del intervalo  $[t_k, t_{k+1}]$ . Sus coeficientes se eligen para reproducir el desarrollo de Taylor hasta un cierto orden  $p$ , lo que garantiza exactitud global de orden  $p$  bajo hipótesis de suavidad. En notación de Butcher, cada método queda descrito por un *tableau* que fija los tiempos internos (*etapas*) y sus pesos.

El esquema explícito de cuarto orden (orden  $p = 4$ ) se define las actualizaciones:

$$\begin{aligned} \mathbf{k}_1 &= \mathbf{f}(t_k, \mathbf{x}_k), \\ \mathbf{k}_2 &= \mathbf{f}\left(t_k + \frac{\Delta t}{2}, \mathbf{x}_k + \frac{\Delta t}{2} \mathbf{k}_1\right), \\ \mathbf{k}_3 &= \mathbf{f}\left(t_k + \frac{\Delta t}{2}, \mathbf{x}_k + \frac{\Delta t}{2} \mathbf{k}_2\right), \\ \mathbf{k}_4 &= \mathbf{f}(t_k + \Delta t, \mathbf{x}_k + \Delta t \mathbf{k}_3), \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \frac{\Delta t}{6} (\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4). \end{aligned}$$

Es un método explícito, de costo moderado y buen equilibrio precisión/costo cuando la dinámica permite tamaños de paso no demasiado pequeños. Su región de estabilidad es acotada (no es  $A$ -estable), por lo que en problemas rígidos la selección de  $\Delta t$  queda limitada por las frecuencias altas del sistema.

La sección §2.2.2 propone reescribir la ecuación de movimiento de segundo orden como un sistema de primer orden

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{r}),$$

ya sea lineal invariante en el tiempo,  $\mathbf{f}(t, \mathbf{x}, \mathbf{r}) = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{r}(t)$ , o no-lineal con variables internas. En este marco, RK4 se aplica directamente sustituyendo  $\mathbf{f}$  en las etapas  $\mathbf{k}_1, \dots, \mathbf{k}_4$ . Si hay entradas  $\mathbf{r}(t)$  (cargas/registro sísmico), se evalúan de manera coherente en  $t_k, t_k + \Delta t/2$  y  $t_k + \Delta t$  (interpolación o forma analítica). Esta conexión proporciona un integrador de referencia claro para el sistema de primer orden y un puente natural hacia métodos orientados a dinámica estructural.

Aunque RK4 ofrece alto orden y una formulación conceptualmente simple sobre el espacio de estados, en análisis tiempo–historia de sistemas estructurales rígidos y con histéresis resulta más conveniente adoptar a continuación los **métodos implícitos de la familia de Newmark** (y sus extensiones HHT y *generalized- $\alpha$* ), que proporcionan estabilidad amplia y control de disipación en altas frecuencias con costos por paso competitivos en modelos típicos de edificación.

### 2.3.3. Imposición iterativa del equilibrio: Newmark y Newton–Raphson.

Los métodos de Newmark constituyen una familia de esquemas de integración temporal para sistemas mecánicos de segundo orden que se construyen parametrizando las relaciones cinemáticas entre desplazamientos, velocidades y aceleraciones en dos instantes consecutivos. Dado un paso de integración  $\Delta t$  y los parámetros

$(\beta, \gamma)$ , las leyes cinemáticas que vinculan el estado en  $t_k$  y  $t_{k+1} = t_k + \Delta t$  se escriben como

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \Delta t \dot{\mathbf{u}}_k + \Delta t^2 \left[ \left( \frac{1}{2} - \beta \right) \ddot{\mathbf{u}}_k + \beta \ddot{\mathbf{u}}_{k+1} \right], \quad (2.9)$$

$$\dot{\mathbf{u}}_{k+1} = \dot{\mathbf{u}}_k + \Delta t \left[ (1 - \gamma) \ddot{\mathbf{u}}_k + \gamma \ddot{\mathbf{u}}_{k+1} \right]. \quad (2.10)$$

La elección de  $(\beta, \gamma)$  determina las propiedades de estabilidad y disipación: cuando  $\beta = 0$  (típicamente con  $\gamma = \frac{1}{2}$ ) se recupera un esquema explícito equivalente a diferencias centrales; para  $\beta > 0$  el método es implícito. En la práctica estructural destacan dos elecciones: aceleración promedio ( $\beta = \frac{1}{4}, \gamma = \frac{1}{2}$ ), incondicionalmente estable en el caso lineal y sin amortiguamiento numérico de alta frecuencia, y aceleración lineal ( $\beta = \frac{1}{6}, \gamma = \frac{1}{2}$ ), que mejora la aproximación de baja frecuencia pero presenta estabilidad condicional y exige un control más estricto del tamaño de paso (Newmark, 1959; Chopra, 2017).

El núcleo del procedimiento consiste en combinar (2.9)–(2.10) con el equilibrio dinámico impuesto para el paso  $k + 1$ . Para un sistema MDOF con posibles no-linealidades internas se adopta la siguiente formulación de equilibrio en  $t_{k+1}$ :

$$\mathbf{r}(\mathbf{u}_{k+1}, \dot{\mathbf{u}}_{k+1}, \ddot{\mathbf{u}}_{k+1}; \mathbf{z}_{k+1}) = \underbrace{\mathbf{M} \ddot{\mathbf{u}}_{k+1} + \mathbf{C}_{k+1} \dot{\mathbf{u}}_{k+1} + \mathbf{f}_{\text{int}}(\mathbf{u}_{k+1}, \dot{\mathbf{u}}_{k+1}, \mathbf{z}_{k+1})}_{\hat{\mathbf{f}}_{\text{int}}} - \mathbf{p}_{k+1} = \mathbf{0}, \quad (2.11)$$

donde  $\mathbf{M}$  es la matriz de masa;  $\mathbf{C}_{k+1}$  representa el amortiguamiento efectivo (posiblemente dependiente del estado);  $\mathbf{f}_{\text{int}}$  reúne las fuerzas internas, lineales o histeréticas (mediante variables internas  $\mathbf{z}$ );  $\hat{\mathbf{f}}_{\text{int}}$  agrupa todas las fuerzas del sistema (inerciales, amortiguadoras y restauradoras); y  $\mathbf{p}_{k+1}$  comprende las excitaciones externas, incluyendo fuerzas nodales y/o la contribución inercial equivalente por aceleración sísmica de base.

Para resolver (2.11) resulta conveniente eliminar las incógnitas cinemáticas redundantes y trabajar con una sola variable primaria. Reordenando (2.9)–(2.10) y sustituyendo en (2.11), el problema en  $t_{k+1}$  queda formulado en la única incógnita  $\mathbf{u}_{k+1}$ , sobre la base de predicciones puramente cinemáticas determinadas por el estado conocido en  $t_k$ .

En régimen no-lineal, la solución se plantea mediante Newton–Raphson, utilizando la rigidez tangente consistente, entendida como la derivada direccional de la fuerza interna respecto del vector de desplazamientos, con la actualización de las variables internas. Denotando por  $\mathbf{K}_{\text{tan}, k+1} = \partial \hat{\mathbf{f}}_{\text{int}} / \partial \mathbf{u}_{k+1}$  dicha rigidez y linealizando el residuo, el sistema incremental a resolver en cada iteración  $i$  adopta la forma

$$\mathbf{K}_{\text{tan}, k+1}^{\text{eff}} \Delta \mathbf{u}^{(i)} = -\mathbf{r}(\mathbf{u}_{k+1}^{(i)}), \quad \mathbf{u}_{k+1}^{(i+1)} = \mathbf{u}_{k+1}^{(i)} + \Delta \mathbf{u}^{(i)}. \quad (2.12)$$

La utilización de una matriz tangente coherente con el algoritmo constitutivo favorece la convergencia cuasi–cuadrática de Newton y un balance energético adecuado. Es posible emplear una rigidez secante o una aproximación numérica por diferencias finitas, pero ello suele degradar la tasa de convergencia, incrementando el número de iteraciones y la sensibilidad a las tolerancias. Tras la convergencia, las magnitudes cinemáticas del paso  $k + 1$  se recuperan directamente a partir de (2.9)–(2.10).

**Estructura por niveles (local–global).** En cada iteración global  $i$ :

1. Se *predicen* variables cinemáticas en  $t_{k+1}$  (Newmark) con los valores de  $\mathbf{u}_{k+1}^{(i)}$ .
2. A nivel de elemento, se integran las leyes constitutivas (actualizando  $\mathbf{z}^e$ ) y se obtienen  $\mathbf{f}_{\text{int}}^e$  y  $\mathbf{K}_{\text{tan}}^e$  *consistentes*.
3. Se *ensamblan*  $\mathbf{r}_{k+1}^{\text{eff}}$  y  $\mathbf{K}_{k+1}^{\text{eff}}$ .
4. Se resuelve (2.12) y se actualiza  $\mathbf{u}_{k+1}^{(i+1)}$  (con opción de *line search* cuando es conveniente).
5. Se verifica convergencia; en caso contrario se repite.

### Esquema resumido (Newmark implícito).

1. Para  $k = 0, \dots, N-1$ : construir fuerzas externas/equivalentes en  $t_{k+1}$  y predictores cinemáticos (Newmark).
2. Iterar  $i = 0, 1, \dots$ :
  - a) Actualizar estados locales de cada elemento  $\Rightarrow \mathbf{f}_{\text{int}}^e, \mathbf{K}_{\text{tan}}^e$  consistentes.
  - b) Ensamblar  $\mathbf{r}_{k+1}^{\text{eff}}, \mathbf{K}_{k+1}^{\text{eff}}$ .
  - c) Resolver  $\mathbf{K}_{k+1}^{\text{eff}} \Delta \mathbf{u} = -\mathbf{r}_{k+1}^{\text{eff}}$ .
  - d) Corregir  $\mathbf{u}$  (y cinemática asociada) y chequear tolerancias.
3. Al converger: aceptar  $\{\mathbf{u}_{k+1}, \dot{\mathbf{u}}_{k+1}, \ddot{\mathbf{u}}_{k+1}, \mathbf{z}_{k+1}\}$  y avanzar  $k$ .

**Variantes con amortiguamiento numérico selectivo.** Las extensiones de Hilber–Hughes–Taylor (HHT- $\alpha$ ) y generalized- $\alpha$  modifican los puntos de evaluación de fuerzas y aceleraciones para introducir disipación numérica controlada en la banda alta preservando la exactitud de la banda baja; esto resulta útil para suprimir oscilaciones espurias asociadas a discretización o idealizaciones locales sin contaminar la respuesta de interés (Hilber et al., 1977; Chung y Hulbert, 1993).

## 2.4. Modelos hysteréticos y degradación

La no-linealidad inelástica en sistemas estructurales se modela mediante modelos matemáticos con memoria (*hysteréticos*) capaces de reproducir plastificación, ablandamiento y degradación bajo carga cíclica. En este capítulo revisamos los modelos hysteréticos **paramétricos** o *model-driven* y profundizamos en dos representantes centrales para análisis tiempo–historia: el **modelo bilineal** (familia polinomial por tramos) y el **modelo de Bouc–Wen** (familia diferencial). Finalmente, discutimos los **mecanismos de degradación** y sus **implicancias en el desplazamiento residual**.

### 2.4.1. Modelos hysteréticos *model-driven*

En métodos *model-driven*, se elige primero una forma paramétrica del modelo hysterético y luego se identifican sus parámetros a partir de datos (ensayos experimentales o registros numéricos). El resultado es un modelo compacto, interpretable y físicamente trazable, capaz de describir con precisión lazo y memoria siempre que la forma elegida recoja los rasgos relevantes (asimetría, *pinching*, degradación, dependencia de velocidad, etc.). Por ello, en *model-driven* la selección del modelo es crítica: si el rasgo clave no está contemplado, ningún ajuste paramétrico lo “creará” a posteriori. (Wang et al., 2023a)

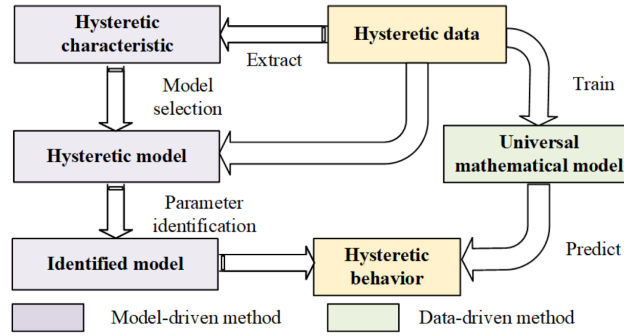


Figura 2.1: Diferencia entre métodos model-driven y data driven. (Fuente: Wang et al. (2023a))

En contraste, los métodos *data-driven* (Sección 2.6) aprenden directamente la relación entrada–salida desde los datos, sin imponer una forma de lazo específica. Esto ofrece flexibilidad, pero exige más datos y cuidados de generalización/interpretabilidad. La Fig. 2.1 resume bien ambos enfoques y su complementariedad: selección + identificación en *model-driven* frente a aprendizaje directo en *data-driven* (Wang et al., 2023a).

Dentro de los modelos *model-driven* tenemos tres familias representativas: (i) polinomiales por tramos, (ii) diferenciales y (iii) por operadores. A continuación se destacan sus ideas base, cuándo se usan y qué rasgos capturan:

### 2.4.2. Polinomiales por tramos

Los modelos *por tramos* representan la relación fuerza–deformación mediante **tramos** con pendientes conocidas (lineales o curvas) y **reglas de recarga/descarga** que definen qué tramo está activo cuando se invierte la carga. Su fortaleza es la *trazabilidad*: cada parámetro controla un rasgo claro de la *backbone* (rigidez inicial  $k_0$ , resistencia  $F_y$ , rigidez posfluencia  $k_p = \alpha k_0$ ) o de las ramas de recarga (p. ej., *pinching*). Esto los hace muy eficientes en elementos concentrados (rótulas) y macroelementos para marcos.(Chopra, 2017). Ejemplo más simple y representativo de estos tipos de modelos es el modelo bilineal.

**Modelo bilineal** Sea  $u$  el desplazamiento relativo y  $f$  la fuerza restauradora. Se define como desplazamiento de fluencia  $u_y := F_y/k_0$  y la rigidez posfluencia  $k_p = \alpha k_0$  con  $\alpha \in [0, 1]$ . Una formulación compacta con endurecimiento cinemático (backstress  $\xi$ ) es:

$$f = k_0 (u - u^P) + k_p u^P, \quad \dot{u}^P = \begin{cases} 0, & |f - \xi| < F_y, \\ \frac{k_0}{k_0 + k_p} \dot{u} \operatorname{sgn}(f - \xi), & |f - \xi| = F_y, \end{cases} \quad \dot{\xi} = \frac{k_0 k_p}{k_0 + k_p} \dot{u} \operatorname{sgn}(f - \xi),$$

con  $u^P(0) = \xi(0) = 0$ . En implementación de elementos finitos se usan *reglas de recarga* equivalentes y un algoritmo de *return-mapping* que garantiza consistencia y una **rigidez tangente** bien definida para Newton. En inversión de carga, la trayectoria vuelve *paralela* a la rama elástica, generando lazos lineales sin *pinching*.

**Extensiones del modelo** El **IMK modificado** (Ibarra et al., 2005) introduce *deterioro cíclico* gobernado por ductilidad y energía; **Pinching4** es un modelo que forma parte de los modelos incorporados en el entorno del software OpenSees (PEER 2006), el cual es controlado con 16 parámetros incorporando *pinching* explícito y reglas de degradación separadas de rigidez/resistencia.

### 2.4.3. Diferenciales

Los modelos **diferenciales** introducen **variables internas** que cumplen EDOs y se acoplan a la ecuación de movimiento. En ellos, la fuerza se descompone en una parte *elástica* y otra *histórica* gobernada por la variable interna, cuya evolución recoge *memoria y forma* del lazo.

**Bouc–Wen clásico.** El modelo de Bouc-Wen, inicialmente desarrollado por Bouc (1967) y Bouc (1971) y posteriormente generalizado por Wen (1976), es ampliamente utilizado para representar el comportamiento no-lineal hysterético de estructuras sometidas a cargas dinámicas. Su aplicación trasciende disciplinas como la electrónica, la mecánica y la ingeniería estructural. Según este modelo la fuerza interna de restauración esta dada por  $\mathbf{f}_{\text{int}} = \alpha k_0 \mathbf{u} + (1 - \alpha) k_0 \mathbf{z}$ , donde  $\alpha k_0 \mathbf{u}$  representa la *componente elástica* (describe la respuesta instantánea) y la segunda expresión  $(1 - \alpha) k_0 \mathbf{z}$  es el *componente de histerésis* (el cual depende de la historia de esfuerzos y deformaciones pasadas),  $k_0$  es la rigidez lineal pre-fluencia del sistema,  $\alpha \in [0, 1]$  es la relación entre la rigidez post-fluencia y la rigidez lineal inicial,  $\mathbf{z}$  es la variable histórica adimensional no observable que obedece la ecuación diferencial no-lineal 2.14 con condición inicial cero ( $z(0) = 0$ ), la cual está normalizada por el desplazamiento de fluencia del sistema  $u_y$ , por lo que la fuerza hysterética se define como  $(1 - \alpha) k_0 u_y \mathbf{z}$  y la ecuación dinámica de equilibrio para un sistema SDOF queda descrita por,

$$m\ddot{u}(t) + c\dot{u}(t) + \alpha k_0 u(t) + (1 - \alpha) k_0 u_y z(t) = p(t) - m a_g(t), \quad (2.13)$$

$$\dot{z}(t) = u_y^{-1} \left( A \dot{u}(t) - \beta |\dot{u}(t)| |z(t)|^{n-1} z(t) - \gamma \dot{u}(t) |z(t)|^n \right) \quad (2.14)$$

donde  $A, \beta, \gamma$  son parámetros adimensionales que afectan el tamaño y la forma del ciclo histórico. Es importante notar que la notación para estos parámetros puede variar entre diferentes estudios, y a menudo los roles de  $\beta$  y  $\gamma$  se intercambian. El parámetro exponencial  $n \geq 1$  influye en la transición del comportamiento elástico al post-elástico del sistema: valores pequeños de  $n$  resultan en una transición suave, mientras que valores grandes hacen que esta transición sea abrupta, similar a un modelo bilineal. Esta estructura permite reproducir lazos suaves, con transición y memoria realista (Wen, 1976). En la Fig. 2.2 se observa cómo cambia el ciclo de histerésis dependiendo de los parámetros seleccionados y la comparación de este modelo con el modelo bilineal.

**Degradación (Baber–Wen–Noori).** La degradación cíclica se introduce haciendo  $A, \beta, \gamma$  y/o  $(1 - \alpha) k_0$  dependientes de una medida de historia (p. ej., energía disipada  $\mathcal{E}$ ):  $A(\mathcal{E}), \beta(\mathcal{E}), \gamma(\mathcal{E})$ . Con ello se reducen rigidez y resistencia efectivas por ciclo, y pueden emerger asimetrías bajo historias no balanceadas. (Baber y Wen, 1981; Baber y Noori, 1985)

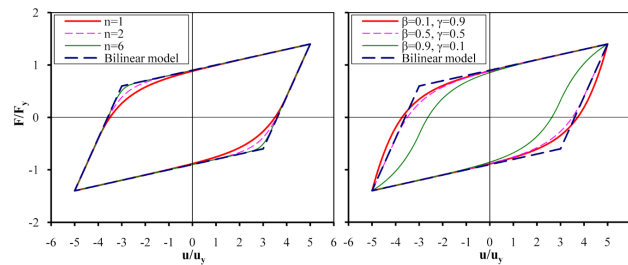


Figura 2.2: Ciclos de Histéresis del modelo Bouc-Wen para diferentes parámetros (Fuente: Charalampakis (2010))

En MDOF,  $z$  puede asociarse a grados de libertad, elementos o fibras, manteniendo una formulación en espacio de estados natural (Sec. 2.2).

#### 2.4.4. Modelos por operadores

Los modelos por operadores representan la histéresis como la superposición de respuestas elementales con memoria, lo que permite reproducir fenómenos complejos como sub-lazos internos y dependencia de la historia. Entre los enfoques más representativos se encuentran el modelo de Preisach, el Prandtl–Ishlinskii (PI) y sus extensiones (como el Krasnosel’skii–Pokrovskii o el Maxwell–Slip) (Wang et al., 2023a).

Estos modelos son útiles para sistemas con memoria compleja e histéresis independiente de la velocidad. Sin embargo, su aplicación práctica enfrenta el reto de la identificación de parámetros, normalmente más exigente que en modelos más simples (p. ej., bilineal, IMK o Bouc–Wen).

#### 2.4.5. Implicancias en el desplazamiento residual (*drift residual*)

El **drift residual** posterior a un evento sísmico sintetiza la porción plástica que permanece en la estructura incluso después de disiparse la vibración transitoria. Este parámetro resulta fundamental en la evaluación del desempeño sísmico, ya que condiciona las decisiones sobre *reparación* o *demolición*, al estar directamente relacionado con la habitabilidad y la reparabilidad de la edificación (McCormick et al., 2008).

Su magnitud depende de diversos factores interrelacionados:

1. la **geometría y el área** de los ciclos de histéresis, asociada a la energía disipada por ciclo,
2. las **asimetrías** en los mecanismos de endurecimiento o ablandamiento entre ramas de carga y descarga,
3. la **degradación cíclica** de rigidez y resistencia que acumula la estructura con el número de ciclos, y
4. la **historia temporal de la excitación** y el **tipo de sismo**; por ejemplo, eventos de subducción de múltiples ciclos suelen inducir menores deformaciones residuales en comparación con sismos de carácter impulsivo (como aquellos con pulsos cercanos a la falla).

La Figura 2.3 ilustra estos efectos mostrando la evolución del **desplazamiento relativo** durante un registro sísmico. En la señal original (línea verde) se observa una respuesta altamente fluctuante, reflejo de la naturaleza transitoria y oscilatoria del movimiento. Al aplicar una media móvil (línea roja), se revela con mayor claridad

la tendencia acumulativa de la respuesta, destacando que, incluso tras la atenuación de las vibraciones, persiste un desplazamiento remanente. Este valor residual corresponde precisamente al *drift* plástico permanente, que se mantiene estable una vez que cesa la entrada de energía. Las bandas de dispersión  $\pm\sigma$  (líneas negras) refuerzan además la variabilidad local de la señal.

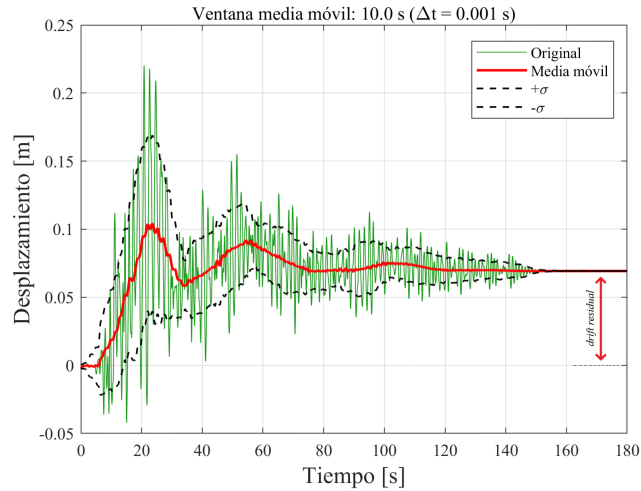


Figura 2.3: Evolución del desplazamiento relativo bajo excitación sísmica. La componente plástica acumulada se refleja en el *drift* residual, que persiste aun después de la disipación de la respuesta transitoria.

De este modo, la figura refuerza visualmente cómo la acumulación de deformaciones plásticas, dependiente de la energía disipada, la asimetría de la respuesta y la historia temporal, se traduce en un **desplazamiento residual medible**, parámetro crítico para la evaluación pos-sismo.

## 2.5. Sistemas de aislamiento sísmico y péndulo friccional (FPS)

Los **sistemas de aislamiento sísmico** son una estrategia efectiva para reducir la demanda sísmica en las estructuras al desacoplar el movimiento de la superestructura respecto del terreno. Su principio es simple: interponer un sistema flexible y disipativo que alargue el período estructural y, al mismo tiempo, reduzca las aceleraciones transmitidas. Esto permite que las edificaciones mantengan su integridad estructural incluso frente a movimientos sísmicos severos. Estos sistemas son inherentemente **no-lineales**, ya que su rigidez y disipación varían con el nivel de desplazamiento y la historia de carga.

Entre las distintas tecnologías disponibles, los **aisladores de péndulo friccional (FPS)** destacan por su capacidad de acomodar **desplazamientos muy grandes** gracias a su superficie cóncava y su mecanismo de deslizamiento con fricción controlada. La curvatura de la superficie proporciona una fuerza recentrante natural, mientras que la fricción introduce disipación estable durante el movimiento. Esta combinación permite un alargamiento significativo del período estructural y una reducción de las aceleraciones transmitidas, aunque exige un control adecuado del espacio disponible para estos grandes desplazamientos.

La naturaleza no-lineal del comportamiento de los sistemas aislados hace que sea conveniente representarlos mediante un **sistema de un grado de libertad (SDOF)**. Esta aproximación es suficiente en muchos casos,

ya que el movimiento principal ocurre en el plano de aislamiento, y la respuesta de la superestructura puede capturarse con modelos equivalentes o asumir que esta tendrá un comportamiento como bloque rígido. Esta no-linealidad también puede representarse con diferentes modelos constitutivos, desde leyes de fricción simples hasta formulaciones como el modelo **Bouc–Wen**, que permite capturar la transición entre rigidez, deslizamiento y disipación bajo diferentes niveles de demanda.

Aunque los sistemas de aislamiento tienden a recentrar la estructura después del evento sísmico, pueden presentarse **desplazamientos residuales (drift)** en ciertas condiciones. Entre las principales causas se incluyen movimientos sísmicos con pulsos asimétricos, variaciones transitorias en las propiedades de fricción, efectos térmicos o de velocidad durante el sismo, y fenómenos asociados al arranque y detención del deslizamiento. Estos desplazamientos residuales, deben ser considerados en el diseño y verificación para garantizar el funcionamiento adecuado del sistema después de un evento sísmico.

## 2.6. Excitaciones sísmicas: caracterización y preparación de registros

El desempeño y la fidelidad de un análisis tiempo–historia no-lineal dependen de una caracterización adecuada de la excitación sísmica y de un preprocesamiento cuidadoso de los registros. Esta sección resume las medidas de intensidad más utilizadas, criterios de selección y escalado de conjuntos de acelerogramas, buenas prácticas de preprocesamiento, la síntesis de registros artificiales mediante modelos estocásticos (Kanai–Tajimi modificado con envolventes no estacionarias) y las implicancias de la variabilidad registro–a–registro para la generalización de modelos y metamodelos.

### 2.6.1. Medidas de intensidad y contenido espectral

Para cuantificar la severidad de un movimiento y anticipar su efecto sobre una estructura, se emplean medidas de intensidad (IM) complementarias. Entre las *IM escalares* más comunes están el *PGA (Peak Ground Acceleration)* y el *PGV (Peak Ground Velocity)*, y, en el dominio respuesta, la aceleración o desplazamiento espectral de pseudo-respuesta  $S_a(T, \zeta)$  y  $S_d(T, \zeta)$  respectivamente, del oscilador SDOF con periodo  $T$  y amortiguamiento  $\zeta$ , que se correlaciona fuertemente con demandas de aceleración/deformación a nivel de modos dominantes (Chopra, 2017). Para caracterizar *duración* y contenido energético se utilizan la *intensidad de Arias*  $I_A = \frac{\pi}{2g} \int_0^{t_f} a^2(t) dt$  y la *duración significativa* basada en percentiles de  $I_A$  (por ejemplo,  $D_{5-95}$ ) (Arias, 1970). En frecuencia, el *contenido espectral* del movimiento se analiza mediante el *Fourier Amplitude Spectrum (FAS)* y, cuando se busca caracterizar cómo se distribuye la energía a lo largo de las frecuencias, mediante la *densidad espectral de potencia (PSD)*. Estas herramientas, junto con medidas derivadas como espectros condicionados a un período de interés, son críticas al seleccionar registros que sean compatibles con un espectro objetivo o que reproduzcan las características dinámicas de una estructura específica.

### 2.6.2. Selección y escalado de registros

La selección de un conjunto de acelerogramas debe representar la amenaza sísmica relevante (mecanismo de falla, magnitud, distancia y condiciones de sitio) y el tipo de movimiento, pues estos factores influyen

en duración, pulsos y contenido espectral. En la práctica normativa, se forman conjuntos de registros reales y se *escalán* de modo que la media de  $S_a(T, \zeta)$  en un intervalo de periodos alrededor del fundamental de la estructura no sea menor que un espectro objetivo (p. ej., espectro de diseño) dentro de tolerancias especificadas. Alternativamente, enfoques *hazard-consistentes* emplean el *Conditional Mean Spectrum* (CMS), que condiciona la selección de registros al valor de  $S_a(T^*)$  en un período de interés y a su desviación estándar asociada, representada por  $\varepsilon$ . Este parámetro  $\varepsilon$  indica qué tan por encima o por debajo del valor medio esperado se encuentra la intensidad espectral en ese período, lo que permite seleccionar registros más representativos y estadísticamente compatibles con el nivel de amenaza y la variabilidad del sitio. Para alineación estricta espectral, el *spectral matching* puede aplicarse, aunque debe usarse con cautela para no introducir artificialidades temporales; el escalado simple por factor mantiene la fase y la estructura temporal originales, siendo preferido cuando es suficiente cumplir metas espectrales promedio.

### 2.6.3. Preprocesamiento: corrección de línea base, filtrado y remuestreo

Los acelerogramas deben ser corregidos para remover *drifts* y artefactos instrumentales que contaminan velocidad y desplazamiento integrados. Un flujo de trabajo típico incluye: (i) corrección de *línea base* (p. ej., remoción de offset y tendencias de orden bajo por tramos), (ii) filtrado paso alto/paso bajo (Butterworth u otros) con frecuencias de corte coherentes con el ancho de banda útil del registro y del sistema estructural, (iii) sincronización temporal de componentes y (iv) remuestreo a un  $\Delta t$  uniforme y adecuado para la integración numérica. Estas decisiones afectan directamente las IM y la integridad de  $I_A$ ; por ello, es imprescindible documentar parámetros de filtrado y compensaciones adoptadas.

### 2.6.4. Registros artificiales: Kanai–Tajimi modificado y envolventes no estacionarias

La generación de registros artificiales controlables es útil para estudios paramétricos y para equilibrar bases de datos cuando se requieren muchas realizaciones. Un enfoque clásico modela la aceleración de base como un proceso lineal filtrado, este es el filtro de Kanai-Tajimi (KT), desarrollado inicialmente por Kanai (1957) y posteriormente refinado por Tajimi (1960), es un modelo simplificado que representa la respuesta del suelo durante excitaciones sísmicas. Este modelo considera el suelo como un sistema de un grado de libertad (*Single Degree of Freedom*, SDOF), donde la señal sísmica originaria del lecho rocoso, es modelada como ruido blanco y posteriormente modulada por las características del sitio, como la frecuencia natural del suelo  $\omega_g$  y su nivel de amortiguamiento  $\zeta_g$ , adaptándose así a los efectos de sitio específicos, esta representación se ilustra en la Figura 2.4.

La dinámica del filtro se describe mediante la siguiente ecuación diferencial:

$$\ddot{x}_g(t) + 2\zeta_g\omega_g\dot{x}_g(t) + \omega_g^2x_g(t) = -\ddot{W}(t) \quad (2.15)$$

Donde  $\ddot{x}_g(t)$ ,  $\dot{x}_g(t)$ , y  $x_g(t)$  representan la aceleración, velocidad y desplazamiento relativos del sistema SDOF que sirve de filtro, respectivamente; y  $\ddot{W}(t)$  representa la aceleración absoluta en la roca, la cual se asume que se puede modelar como un ruido blanco con una densidad espectral de potencia constante  $S_0$ . La

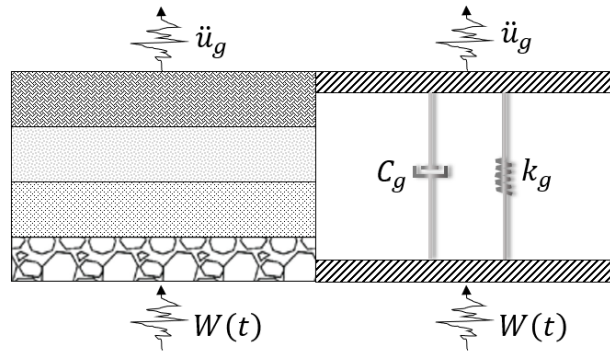


Figura 2.4: Representación gráfica del filtro Kanai-Tajimi

aceleración absoluta del suelo para este modelo ( $\ddot{u}_{KT}$ ) se expresa como:

$$\ddot{u}_{KT}(t) = \ddot{W}(t) + \ddot{x}_g(t) = -2\zeta_g\omega_g\dot{x}_g(t) - \omega_g^2x_g(t) \quad (2.16)$$

Realizando el análisis en el dominio de la frecuencia, se obtiene la densidad espectral de potencia de la aceleración del suelo:

$$S^{KT}(\omega) = S_0 \frac{\omega_g^4 + 4\zeta_g^2\omega_g^2\omega^2}{(\omega_g^2 - \omega^2)^2 + 4\zeta_g^2\omega_g^2\omega^2} \quad (2.17)$$

La varianza del proceso, que es igual al área bajo la curva de  $S^{KT}(\omega)$ , se calcula como:

$$\sigma_{\ddot{u}_{KT}}^2(\omega) = \frac{\pi}{4} \frac{\omega_g (1 + 4\zeta_g^2)}{\zeta_g} S_0 \quad (2.18)$$

Sin embargo a pesar de su amplio uso en la simulación de excitaciones sísmicas, el filtro tiende a sobreestimar la densidad espectral en bajas frecuencias, lo que llevó a Clough y Penzien (1993) a introducir una modificación. Esta ajusta el espectro original para que tenga potencia cero en  $\omega \approx 0$ , resultando en una mayor precisión para reflejar las condiciones reales del movimiento del suelo. La dinámica del filtro adicional introducido por Clough-Penzien esta regida por la siguiente expresión:

$$\ddot{x}_f(t) + 2\zeta_f\omega_f\dot{x}_f(t) + \omega_f^2x_f(t) = \ddot{u}_{KT}(t) \quad (2.19)$$

Donde  $\ddot{x}_f(t)$ ,  $\dot{x}_f(t)$ , y  $x_f(t)$  representan respectivamente la aceleración, velocidad y desplazamiento relativos del sistema SDOF que sirve de filtro extra al de KT,  $\omega_f$  y  $\zeta_f$  son los parámetros del modelo de Clough-Penzien (CP), los cuales se ajustan para compensar la sobreestimación en bajas frecuencias, y  $\ddot{u}_{KT}(t)$  es la aceleración sísmica horizontal del modelo de KT definida en la ecuación 2.16. Por lo tanto, el registro de aceleraciones del suelo para el modelo de CP, se define como la respuesta de aceleraciones del filtro KT definido por la siguiente ecuación:

$$\ddot{u}_{CP}(t) = \ddot{x}_f(t) = -2\zeta_f\omega_f\dot{x}_f(t) - \omega_f^2x_f(t) - 2\zeta_g\omega_g\dot{x}_g(t) - \omega_g^2x_g(t) \quad (2.20)$$

Por lo que realizando el análisis en el dominio de la frecuencia, la densidad espectral de potencia de Clough-Penzien (CP), se formula como:

$$S_{CP}(\omega) = S_0 \left( \frac{\omega^4}{(\omega_f^2 - \omega^2)^2 + 4\zeta_f^2 \omega_f^2 \omega^2} \right) \left( \frac{\omega_g^4 + 4\zeta_g^2 \omega_g^2 \omega^2}{(\omega_g^2 - \omega^2)^2 + 4\zeta_g^2 \omega_g^2 \omega^2} \right) \quad (2.21)$$

La varianza del proceso se define como:

$$\sigma_{\ddot{u}_{CP}}^2 = \frac{\pi A(\omega)}{2 B(\omega)} S_0 \quad (2.22)$$

donde:

$$A(\omega) = \omega_g^4 (\zeta_g \omega_f + \zeta_f \omega_g) + 4\zeta_g^2 \omega_g^2 (\zeta_g \omega_f^3 + \zeta_f \omega_g^3 + 4\zeta_g \zeta_f \omega_g \omega_f (\zeta_g \omega_f + \zeta_f \omega_g)) \quad (2.23)$$

$$B(\omega) = 2\zeta_g \zeta_f \left[ (\omega_g^2 - \omega_f^2)^2 + 4\omega_g^2 \omega_f^2 (\zeta_g^2 + \zeta_f^2) + 4\zeta_g \zeta_f \omega_g \omega_f (\omega_g^2 + \omega_f^2) \right] \quad (2.24)$$

La densidad espectral de potencia  $S_0$  se puede estimar considerando que el valor medio máximo de la aceleración del suelo (PGA por sus siglas en inglés) es del orden de tres veces la desviación estándar de la aceleración sísmica  $PGA = 4\sigma_{\ddot{u}_{CP}}$ . Reemplazando esta relación en la ecuación 2.22 se obtiene  $S_0$  mediante la siguiente formulación:

$$S_0 = \left( \frac{PGA}{4} \right)^2 \frac{B(\omega)}{\pi A(\omega)} \quad (2.25)$$

Ambas formulaciones se ilustran en la Figura 2.5, mostrando las diferencias en la respuesta espectral debido a las modificaciones propuestas por Clough y Penzien (1993).

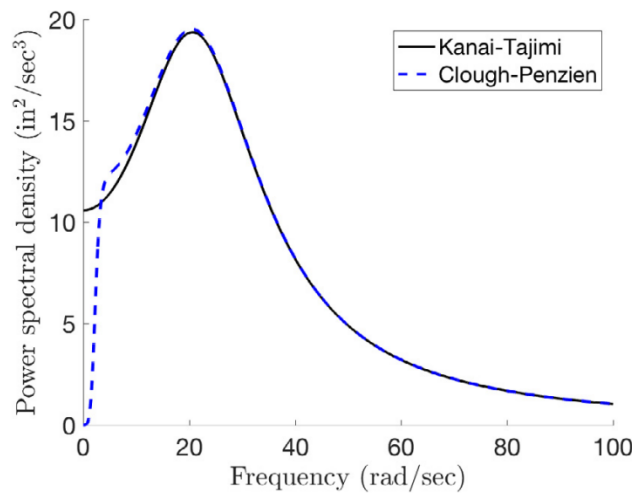


Figura 2.5: Funciones de densidad espectral de potencia de Kanai-Tajimi y Clough-Penzien con  $\omega_g = 8\pi$ ,  $\omega_f = 0.8\pi$  y  $\zeta_g = \zeta_f = 0.6$ . Fuente: Zhu et al. (2017)

Este ruido blanco filtrado que contiene las frecuencias del suelo de interés no posee las características no estacionarias de un registro sísmico real, es por esto que se emplean funciones de modulación, uno de estos modelos es el modelo propuesto por Amin y Ang (1968) el cual destaca por su capacidad de simular las fases del movimiento del terreno durante un sismo. Esta función se caracteriza por tener tres ramas que representan la fase ascendente, la fase de movimiento fuerte y la fase de desvanecimiento de la aceleración.

La función de modulación se define de la siguiente manera:

$$\Psi(t) = \begin{cases} \left(\frac{t}{t_1}\right)^2 & \text{para } 0 \leq t < t_1 \\ 1 & \text{para } t_1 \leq t < t_2 \\ e^{-c(t-t_2)} & \text{para } t \geq t_2 \end{cases} \quad (2.26)$$

En esta ecuación,  $c$  representa la tasa de atenuación de la aceleración,  $t_1$  es el tiempo hasta el inicio de la fase de movimiento fuerte, y  $t_2 - t_1$  la duración de esta fase. Este modelo captura de manera eficaz la no estacionariedad del movimiento del suelo, un aspecto crucial para los análisis detallados de la respuesta estructural ante sismos.

Para ilustrar cómo esta función afecta la simulación de movimientos sísmicos, el gráfico en la figura 2.6 muestra la aplicación de  $\Psi(t)$  en el tiempo, mostrando las tres fases de modulación.

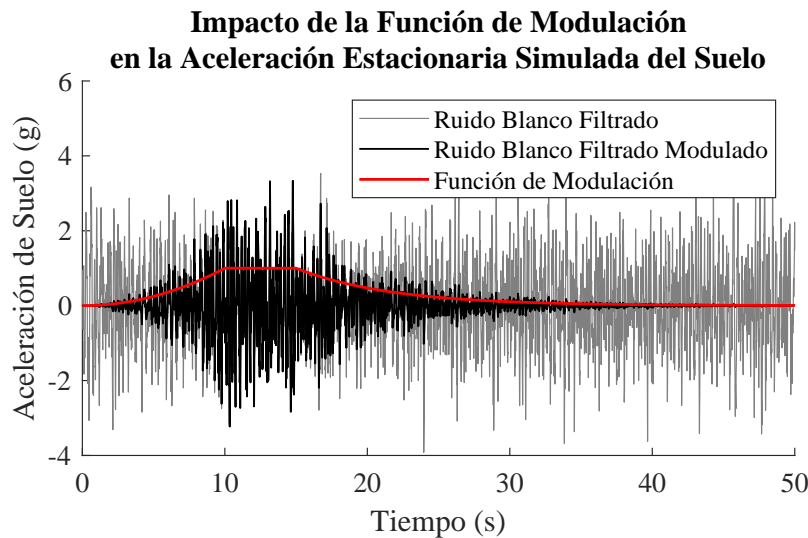


Figura 2.6: Impacto de la función de modulación de Amin y Ang en la simulación de la aceleración del suelo durante un sismo.

Este enfoque detallado de modulación permite realizar simulaciones más precisas de los efectos sísmicos, representando la no estacionariedad de estos, lo cual es esencial para mejorar el diseño y análisis estructural en zonas sísmicas.

### 2.6.5. Variabilidad registro–a–registro y consecuencias para la generalización

Incluso tras una selección y escalado cuidadosos, la *variabilidad registro–a–registro* de demandas sigue siendo elevada, particularmente bajo no-linealidad intensa. Esta dispersión (producto de diferencias en fase, duración y pulsos) se refleja en conjuntos de curvas de respuesta notablemente dispersos, donde para un mismo nivel de intensidad sísmica las demandas pueden variar ampliamente entre registros. Ello genera una incertidumbre epistémica significativa al extrapolar los resultados. Para metamodelos y PINNs, ello implica que los conjuntos de entrenamiento/validación deben *estratificarse por evento* (no mezclar cortes de un mismo registro en particiones distintas), abarcar diversidad de espectros y duraciones, y evaluar capacidad de generalización en registros fuera de la distribución de entrenamiento. Protocolos de *stress testing* por intensidad, duración y contenido espectral ayudan a diagnosticar extrapolabilidad y robustez, asegurando que el diseño sea robusto ante eventos extremos que, aunque poco probables, en algunos casos ayudan a evaluar el desempeño del modelo.

En resumen, medidas de intensidad bien escogidas, selección y escalado de acelerogramas consistentes con la amenaza, un preprocesamiento transparente y la posibilidad de generar registros artificiales controlados constituyen la base para análisis y aprendizaje confiables. El reconocimiento explícito de la variabilidad registro–a–registro es esencial para interpretar dispersiones y para diseñar metamodelos que generalicen más allá de los casos vistos.

## 2.7. Aprendizaje profundo para series temporales estructurales

El aprendizaje profundo ofrece arquitecturas y procedimientos de entrenamiento capaces de aproximar operadores dinámicos no-lineales a partir de datos, lo que lo convierte en un marco natural para modelar series estructurales (aceleraciones, velocidades, desplazamientos, fuerzas internas). En esta sección se resumen los fundamentos del entrenamiento, las arquitecturas más utilizadas, los esquemas de representación de entradas/salidas temporales y aspectos prácticos de normalización y estabilización del entrenamiento en presencia de dependencias largas.

### 2.7.1. Fundamentos de entrenamiento: propagación directa y retropropagación

El entrenamiento de una red neuronal parte de una idea sencilla: dado un conjunto de datos  $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ , se busca que el modelo parametrizado por  $\theta$  sea capaz de aproximar la relación subyacente entre entradas  $\mathbf{x}_i \in \mathbb{R}^d$  y salidas  $\mathbf{y}_i \in \mathbb{R}^m$ . Matemáticamente, la red implementa una función  $f_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^m$ , de modo que para cada ejemplo se obtiene una predicción

$$\hat{\mathbf{y}}_i = f_\theta(\mathbf{x}_i), \quad (2.27)$$

proceso conocido como *propagación directa* o *forward pass*. En este paso, la información fluye a través de la red, aplicando sucesivamente transformaciones lineales y no-lineales hasta producir la salida estimada.

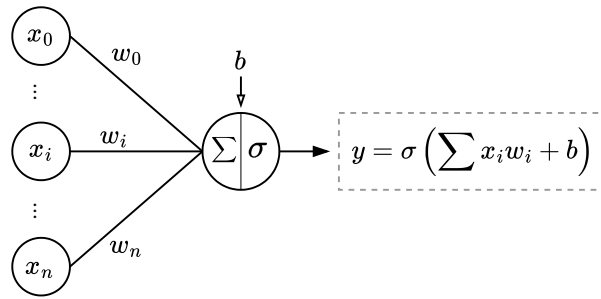


Figura 2.7: Esquema de una neurona artificial: combinación lineal de entradas y pesos, más un sesgo  $b$ , seguida de una función de activación  $\sigma(\cdot)$ .

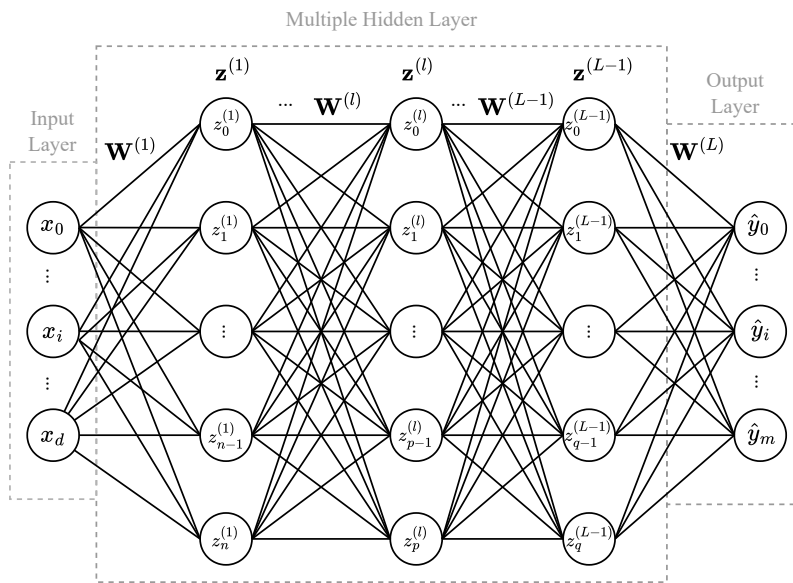


Figura 2.8: Esquema de un perceptrón multicapa (MLP), también conocido como red neuronal totalmente conectada (FCNN).

La unidad básica de las redes neuronales en general es la **neurona artificial**, mostrada en la Fig. 2.7, que recibe un conjunto de entradas  $x_i$ , las pondera mediante pesos  $w_i$ , añade un sesgo  $b$  y aplica una función de activación  $\sigma(\cdot)$  para producir una salida.

A partir de la interconexión de múltiples neuronas en capas se construye la arquitectura más simple y, a la vez, fundamental: el **perceptrón multicapa** (MLP, por sus siglas en inglés), también denominado *red neuronal totalmente conectada* o *fully-connected neural network (FCNN)*. En la Fig. 2.8 se muestra su estructura típica: una capa de entrada, una o varias capas ocultas y una capa de salida, donde cada neurona de una capa se conecta con todas las de la siguiente.

Suponiendo un vector de entrada  $\mathbf{x} \in \mathbb{R}^d$  y una red de  $L$  capas, cada capa  $l$  realiza una transformación afín seguida de una activación no-lineal:

$$\mathbf{z}^{(l)} = \sigma(\mathbf{W}^{(l)}\mathbf{z}^{(l-1)} + \mathbf{b}^{(l)}), \quad \mathbf{z}^{(0)} = \mathbf{x}, \quad \mathbf{z}^{(L)} = \hat{\mathbf{y}}.$$

Aquí,  $\mathbf{W}^{(l)}$  es la matriz de pesos que conecta la capa  $l - 1$  con la capa  $l$ , mientras que  $\mathbf{b}^{(l)}$  es el **vector de bias**, que permite a cada neurona desplazar su superficie de activación independientemente de las entradas. Sin los términos de bias, todas las transformaciones pasarían necesariamente por el origen, restringiendo fuertemente la flexibilidad del modelo.

El operador  $\sigma(\cdot)$  introduce la no-linealidad indispensable para que la red pueda aproximar funciones complejas. Si todas las capas fueran lineales, el modelo completo sería equivalente a una sola transformación lineal, independientemente de la profundidad. Entre las funciones de activación más comunes destacan:

- La **sigmoide** y la **tangente hiperbólica**, usadas en las primeras redes, aunque limitadas por problemas de saturación de gradientes.
- La **ReLU** (*Rectified Linear Unit*), definida como  $\max(0, u)$ , ampliamente utilizada por su simplicidad y estabilidad numérica.

La elección de la activación tiene un impacto directo en la eficiencia del entrenamiento, la estabilidad numérica y la capacidad de generalización del modelo.

Hasta aquí hemos descrito la estructura básica de una red neuronal y el modo en que las entradas se transforman en salidas a través de capas sucesivas. El paso siguiente es introducir la noción de **entrenamiento**. Entrenar una red neuronal consiste en ajustar sus *parámetros internos* de modo que las predicciones  $\hat{\mathbf{y}}_i$  se aproximen lo mejor posible a los valores verdaderos  $\mathbf{y}_i$ . Este ajuste se realiza de manera automática a través de algoritmos de optimización que modifican los parámetros en función del error cometido.

Es fundamental distinguir entre **parámetros** e **hiperparámetros**. Los parámetros son las cantidades que la red aprende durante el entrenamiento: las matrices de pesos  $\mathbf{W}^{(l)}$  y los vectores de bias  $\mathbf{b}^{(l)}$ , que determinan cómo se combinan y transforman las entradas en cada capa. De manera compacta, se denotan como

$$\boldsymbol{\theta} = \{\mathbf{W}^{(1)}, \dots, \mathbf{W}^{(L)}, \mathbf{b}^{(1)}, \dots, \mathbf{b}^{(L)}\},$$

y constituyen el conjunto de parámetros que la red ajusta automáticamente durante el proceso de entrenamiento.

En contraste, los **hiperparámetros** son establecidos externamente por el investigador y definen el marco en el cual ocurre el aprendizaje. Entre los más relevantes se encuentran el número de capas y neuronas, la función de activación, la tasa de aprendizaje  $\eta$ , el tamaño de lote (*batch size*) y el número de épocas de entrenamiento.

Mientras que los parámetros son el resultado del entrenamiento (la red los “aprende”), los hiperparámetros condicionan la forma en que dicho entrenamiento ocurre y, en consecuencia, tienen un efecto decisivo sobre la capacidad del modelo para generalizar.

Para que el entrenamiento pueda llevarse a cabo es necesario definir un criterio que indique qué tan buenas o malas son las predicciones de la red en relación con los datos reales. En otras palabras, se requiere una medida cuantitativa del error que sirva de señal para guiar la actualización de los parámetros  $\boldsymbol{\theta}$ . Este rol lo cumplen las *funciones de pérdida*, que asignan un valor escalar a la discrepancia entre  $\hat{\mathbf{y}}_i$  y  $\mathbf{y}_i$ . En problemas de regresión, una de las más empleadas es el error cuadrático medio (MSE),

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^N \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_2^2, \quad (2.28)$$

mientras que en tareas de clasificación se utiliza frecuentemente la entropía cruzada,

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^m y_{ij} \log \hat{y}_{ij}. \quad (2.29)$$

De esta manera, el entrenamiento se formula como un problema de optimización: encontrar los parámetros  $\theta$  que minimicen la función de pérdida promedio en el conjunto de entrenamiento.

El paso siguiente consiste en determinar cómo modificar los parámetros para reducir dicha pérdida. La dirección más natural para ajustar  $\theta$  es aquella que indique el gradiente de  $\mathcal{L}$  respecto a cada parámetro. En este punto aparece la retropropagación, un algoritmo eficiente que permite calcular dichos gradientes. La idea central es aplicar la regla de la cadena de forma sistemática desde la capa de salida hacia las capas iniciales, de modo que el error se asigne correctamente a cada parámetro. Para evitar ambigüedades, introducimos la *preactivación* de la capa  $l$ ,

$$\mathbf{u}^{(l)} = \mathbf{W}^{(l)} \mathbf{z}^{(l-1)} + \mathbf{b}^{(l)}, \quad \mathbf{z}^{(l)} = \sigma(\mathbf{u}^{(l)}),$$

y recordamos que el objetivo del cálculo de gradientes es minimizar la pérdida  $\mathcal{L}$  ajustando  $\theta$ . La retropropagación define los *errores locales* (o deltas)

$$\delta^{(L)} = \nabla_{\mathbf{z}^{(L)}} \mathcal{L} \odot \sigma'(\mathbf{u}^{(L)}), \quad \delta^{(l)} = (\mathbf{W}^{(l+1)})^\top \delta^{(l+1)} \odot \sigma'(\mathbf{u}^{(l)}), \quad l = L-1, \dots, 1,$$

que encapsulan la contribución de cada neurona de la capa  $l$  al error total. Con estas cantidades, los gradientes respecto de los parámetros de la *misma* capa  $l$  adoptan una forma simple y eficiente:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(l)}} = \delta^{(l)} (\mathbf{z}^{(l-1)})^\top, \quad \frac{\partial \mathcal{L}}{\partial \mathbf{b}^{(l)}} = \delta^{(l)}.$$

En términos escalares, esto equivale a

$$\frac{\partial \mathcal{L}}{\partial w_{jk}^{(l)}} = \delta_j^{(l)} z_k^{(l-1)},$$

lo que muestra de manera explícita cómo el error asignado a la neurona  $j$  de la capa  $l$  se combina con la activación de entrada  $k$  proveniente de la capa anterior. En conjunto, estas expresiones constituyen la esencia de la retropropagación: un procedimiento recursivo que transmite las derivadas desde la salida hasta los parámetros iniciales, evitando recomputaciones y haciendo posible el entrenamiento eficiente de redes profundas.

Una vez calculados los gradientes de todos los parámetros, el siguiente paso consiste en emplearlos para reducir la función de pérdida. Esto se logra mediante algoritmos de optimización iterativos. El esquema más simple es el descenso de gradiente estocástico (SGD), que actualiza los parámetros de la red como

$$\theta^{(k+1)} = \theta^{(k)} - \eta \nabla_{\theta} \mathcal{L}, \quad (2.30)$$

donde  $\eta > 0$  es la tasa de aprendizaje que regula la magnitud de cada ajuste.

En la práctica, el cálculo del gradiente rara vez se realiza sobre el conjunto completo de datos, sino sobre subconjuntos denominados **mini-lotes** (*mini-batches*). El tamaño de lote (*batch size*) constituye un hiperparámetro fundamental, pues controla el compromiso entre precisión y costo computacional: lotes grandes producen estimaciones más estables del gradiente pero demandan mayor memoria, mientras que lotes pequeños introducen más ruido en la estimación, lo que puede ayudar a escapar de mínimos locales pero a costa de una convergencia más fluctuante.

Sobre esta base se han desarrollado métodos más avanzados, entre los cuales destaca Adam, que incorpora promedios móviles de primer y segundo orden de los gradientes para adaptar dinámicamente la escala de las actualizaciones. Este tipo de optimizadores suele proporcionar convergencia más rápida y estable en comparación con el SGD puro.

En síntesis, el entrenamiento de redes neuronales se articula como un ciclo que comienza con la propagación directa para generar predicciones, continúa con la retropropagación para calcular el error que corresponde asignar a cada parámetro, y concluye con la optimización que ajusta dichos parámetros. La repetición de este proceso sobre múltiples ejemplos permite que la red vaya ajustando progresivamente su función aproximadora hasta capturar la relación buscada en los datos.

### 2.7.2. Arquitecturas de redes neuronales

El mecanismo de propagación directa y retropropagación constituye el núcleo del aprendizaje en redes neuronales. Sin embargo, la forma en que se organizan las conexiones entre neuronas da lugar a distintas **arquitecturas**, cada una diseñada para aprovechar mejor ciertas propiedades de los datos y resolver problemas específicos de manera más eficiente.

En la sección anterior se introdujo el perceptrón multicapa (MLP), la arquitectura más simple y a la vez fundamental, que sirve como punto de partida conceptual para el aprendizaje profundo. Los MLP son capaces, al menos en teoría, de aproximar funciones continuas en dominios compactos, lo que sugiere que pueden representar una gran variedad de relaciones no-lineales presentes en los datos. No obstante, su estructura totalmente densa hace que la dimensionalidad de la entrada se propague hacia todas las capas posteriores, lo que implica un número de parámetros y operaciones que crece rápidamente con el tamaño del problema. Por esta razón, aunque los MLP siguen siendo útiles en tareas de regresión o clasificación sobre datos tabulares, en aplicaciones con datos estructurados, como imágenes, secuencias temporales o mallas, suelen complementarse o reemplazarse por arquitecturas más eficientes.

A partir de esta base se han desarrollado modelos más especializados: las redes convolucionales (CNN), adecuadas para explotar patrones espaciales; las recurrentes (RNN), diseñadas para capturar dependencias temporales; y las arquitecturas basadas en mecanismos de atención, como los Transformers, que hoy dominan las tareas de secuencias largas y contextos complejos.

En lo que sigue se describen estas arquitecturas, resaltando sus principios de funcionamiento, sus ventajas y las limitaciones que presentan en el contexto de la modelación estructural.

## Redes convolucionales (CNN)

Una capa lineal o totalmente conectada puede recibir como entrada un tensor de cualquier forma, siempre que este se reorganice en un vector con la cantidad adecuada de coeficientes. Sin embargo, este tipo de capas no resulta práctico cuando se trabaja con tensores de gran tamaño. La razón es que tanto el número de parámetros como la cantidad de operaciones crece de manera proporcional al producto entre las dimensiones de entrada y de salida. Por ejemplo, si se desea procesar una imagen RGB de tamaño  $256 \times 256$  píxeles y producir una salida de la misma dimensión, se requerirían del orden de  $4 \times 10^{10}$  parámetros y multiplicaciones, lo que es computacionalmente inviable (Fleuret, 2024).

Además de estas limitaciones, las señales de alta dimensión (series temporales, campos espaciales e imágenes) exhiben estructura estadística no explotada por una capa densa: correlaciones locales, dependencias secuenciales e invariancias aproximadas a traslaciones (entre otras). Las *capas convolucionales* incorporan un sesgo inductivo adecuado para estas regularidades al compartir pesos localmente y desplazar filtros a lo largo de la señal, reduciendo drásticamente el número de parámetros y el costo computacional sin perder capacidad expresiva.

### Convolución 1D

La convolución unidimensional es la operación prototípica cuando los datos varían a lo largo de un único eje (p. ej., tiempo o una dimensión espacial). Su formulación es general: también puede verse como el procesamiento de una línea de píxeles en una imagen o de cualquier señal definida sobre un eje.

Una convolución 1D se caracteriza por tres hiperparámetros principales:

- el tamaño del kernel  $K$ , que determina la extensión local (ventana) sobre la que se combinan las muestras;
- el número de canales de entrada  $D$ ;
- el número de canales de salida  $D'$ .

Sea  $\mathbf{X} \in \mathbb{R}^{D \times T}$  el tensor de entrada, donde  $T$  es la longitud de la secuencia (tiempo o posición a lo largo de un eje) y  $D$  el número de canales o componentes por muestra. En señales univariadas  $D = 1$ ; para señales multivariadas  $D > 1$  puede representar (i) el número de señales observadas simultáneamente (p. ej., sensores o variables medidas en paralelo; en imágenes, filas o columnas procesadas en una dirección fija) o (ii) distintas características asociadas a cada instante. Así, cada vector columna  $\mathbf{X}_{[:, t_k]} \in \mathbb{R}^D$  contiene las  $D$  observaciones en el instante  $t_k$ .

Intuitivamente, la convolución recorre la secuencia aplicando filtros de tamaño  $D \times K$  sobre bloques locales de longitud  $K$ , y a partir de  $D'$  filtros genera  $D'$  mapas de salida que capturan características locales. Pasando a una descripción formal, definimos un conjunto de filtros compartidos a lo largo del eje y una transformación lineal con sesgo aplicada sobre cada bloque:

$$\varphi(\cdot; W) : \mathbb{R}^{D \times K} \rightarrow \mathbb{R}^{D'}, \quad \mathbf{u}_{t_k} = \varphi(\mathbf{X}_{[:, t_k:t_k+K-1]}; W) + \mathbf{b},$$

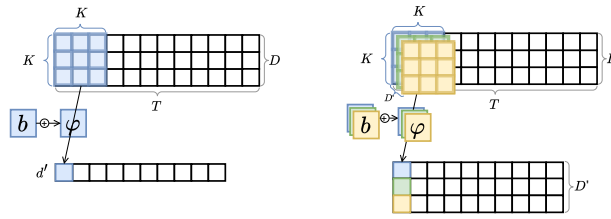


Figura 2.9: Representación capa convolucional con múltiples entradas y aplicación de solo un filtro en la imagen de la izquierda y múltiples filtros en la imagen de la derecha

para  $t_k = 1, \dots, T - K + 1$ , donde  $W \in \mathbb{R}^{D' \times D \times K}$  y  $\mathbf{b} \in \mathbb{R}^{D'}$ . La salida completa se organiza como

$$\mathbf{Z} \in \mathbb{R}^{D' \times (T-K+1)}, \quad \mathbf{Z}_{[:, t_k]} = \mathbf{u}_{t_k}.$$

Esta notación compacta oculta la forma explícita de la operación, que en la práctica se implementa como suma de productos entre entradas y coeficientes del filtro, para esto definiremos que la variable  $u_{t_k}^{(d')}$  representa la *pre-activación* para el filtro  $d'$  en el instante temporal  $t_k$  (para  $d' = 1, \dots, D'$ ). Se calcula aplicando un filtro sobre una ventana local de la secuencia de entrada  $\mathbf{X}$  de tamaño  $K$ , sumando los productos de los valores de entrada con los pesos correspondientes y añadiendo el sesgo  $b^{(d')}$ :

$$u_{t_k}^{(d')} = \sum_{d=1}^{D'} \sum_{\tau=0}^{K-1} W_{d,\tau}^{(d')} X_{d, t_k+\tau} + b^{(d')}.$$

Una vez calculada esta pre-activación, se pasa a través de una función de activación no-lineal  $\sigma$ , como ReLU u otra, que da lugar a  $z_{t_k}^{(d')}$ , el *estado interno activado*. Este proceso es crucial porque introduce no-linealidades en el modelo, permitiéndole aprender representaciones más complejas de la señal de entrada:

$$z_{t_k}^{(d')} = \sigma(u_{t_k}^{(d')}).$$

El tensor de salida,  $\mathbf{Z}$ , contiene las activaciones de todos los filtros para todas las posiciones temporales. La salida para el filtro  $d'$  en la posición temporal  $t_k$  está dada por  $z_{t_k}^{(d')}$ , y así, toda la salida se organiza como un tensor:

$$\mathbf{Z} \in \mathbb{R}^{D' \times (T-K+1)}, \quad \mathbf{Z}_{[:, t_k]} = z_{t_k}^{(d')}.$$

La salida  $\mathbf{Z}$  puede ser la salida final de la capa convolucional, o bien, puede pasar por capas posteriores (como *pooling* o *capas totalmente conectadas*) que generen la salida final  $\mathbf{Y}$ , dependiendo del objetivo de la red. En resumen,  $\mathbf{Z}$  contiene las activaciones locales de los filtros, mientras que  $\mathbf{Y}$  representa la salida final procesada.

*Convención:* la expresión anterior corresponde a la *correlación discreta* (sin inversión del kernel), que es la convención habitual en bibliotecas de *deep learning*. La convolución estricta se obtiene revirtiendo el kernel, lo que no altera el análisis teórico ni la implementación práctica.

### Hiperparámetros adicionales (1D)

Para entender el efecto de los hiperparámetros, consideremos primero una señal univariada  $\mathbf{x} = (x_1, \dots, x_T)$ , un filtro  $\mathbf{w} = (w_0, \dots, w_{K-1})$  y un sesgo  $b$ :

$$u_{t_k} = \sum_{\tau=0}^{K-1} w_{\tau} x_{t_k+\tau} + b, \quad z_{t_k} = \sigma(u_{t_k}).$$

La operación es **causal** si se restringe a valores pasados ( $t_k + \tau \geq 1$ ), de modo que la salida en el instante  $t_k$  depende únicamente de muestras anteriores. Además, puede incorporarse un *factor de dilatación*  $d_{\text{dil}} \in \mathbb{N}$ , que amplía el intervalo cubierto por el filtro sin incrementar el número de parámetros:

$$u_{t_k} = \sum_{\tau=0}^{K-1} w_{\tau} x_{t_k+d_{\text{dil}}\tau} + b.$$

Un valor de  $d_{\text{dil}} > 1$  introduce espacios entre los coeficientes del filtro, aumentando el *campo receptivo* y permitiendo capturar dependencias de mayor alcance.

En la definición anterior se asume un **stride** (paso de convolución) igual a 1, lo que implica que el filtro se aplica en todas las posiciones consecutivas de la secuencia. En general, un  $s \geq 1$  desplaza el filtro cada  $s$  posiciones:

$$u_{t_k} = \sum_{\tau=0}^{K-1} w_{\tau} x_{s t_k+\tau} + b,$$

reduciendo la resolución de la salida (submuestreo) y el costo computacional.

Al no existir suficientes muestras en los extremos para cubrir el filtro completo, la salida tiende a acortarse. Para controlarlo se usa **padding**, que consiste en agregar  $p$  valores ficticios (habitualmente ceros) al inicio y/o al final de la secuencia:

$$\tilde{\mathbf{x}} = (\underbrace{0, \dots, 0}_{p \text{ ceros}}, x_1, \dots, x_T, \underbrace{0, \dots, 0}_{p \text{ ceros}}),$$

lo que permite preservar la misma longitud de entrada en la salida cuando es deseable. Según  $p$ , se distinguen configuraciones frecuentes:

- **Válida (valid):**  $p = 0$ ; no se aplica padding, la salida es más corta que la entrada.
- **Same:**  $p$  suficiente para igualar longitudes de entrada y salida.
- **Full:** se aplica el filtro en todas las posiciones posibles, incluso donde solo parte del kernel se superpone con la señal.

Con dilatación  $d_{\text{dil}}$ , stride  $s$  y padding  $p$ , la longitud de salida es

$$T_{\text{out}} = \left\lfloor \frac{T + 2p - d_{\text{dil}}(K-1) - 1}{s} \right\rfloor + 1.$$

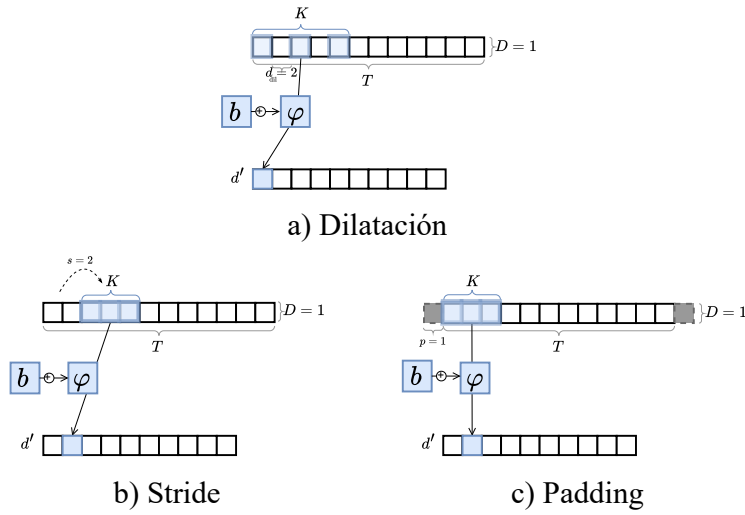


Figura 2.10: Representación de las operaciones de dilatación, stride y padding en redes convolucionales 1D con un canal de entrada.

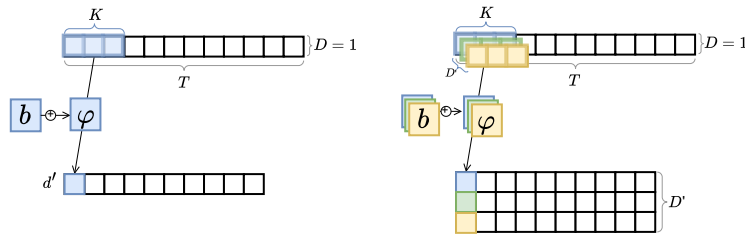


Figura 2.11: Representación capa convolucional con una entrada y aplicación de solo un filtro en la imagen de la izquierda y multiples filtros en la imagen de la derecha

Con múltiples filtros  $\{\mathbf{w}^{(d')}\}_{d'=1}^{D'}$  se obtiene una salida con  $D'$  canales:

$$u_{t_k}^{(d')} = \sum_{\tau=0}^{K-1} w_{\tau}^{(d')} x_{t_k+\tau} + b^{(d')}, \quad z_{t_k}^{(d')} = \sigma(u_{t_k}^{(d')}).$$

Cada filtro genera una representación distinta de la señal, por lo que la salida puede interpretarse como  $D'$  mapas o canales que describen diferentes características locales. La formulación anterior corresponde a una única señal de entrada ( $D = 1$ ).

Sin embargo, en muchos problemas prácticos se trabaja con señales multivariadas o con múltiples canales de entrada, como es el caso de datos provenientes de sensores o variables simultáneas. En estos casos, si la entrada es multivariada ( $D > 1$ ) se modela como un tensor  $\mathbf{X} \in \mathbb{R}^{D \times T}$ , donde  $D$  es el número de señales de entrada y  $T$  su longitud. Cada filtro se extiende ahora a todas las entradas, dando lugar a un conjunto de pesos  $W \in \mathbb{R}^{D' \times D \times K}$  y sesgos  $\mathbf{b} \in \mathbb{R}^{D'}$ :

$$u_{t_k}^{(d')} = \sum_{d=1}^D \sum_{\tau=0}^{K-1} W_{d',d,\tau} X_{d,t_k+\tau} + b^{(d')}, \quad z_{t_k}^{(d')} = \sigma(u_{t_k}^{(d')}),$$

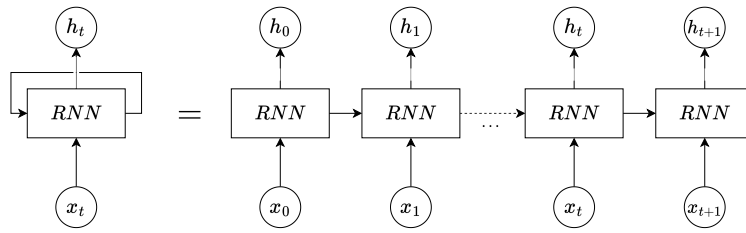


Figura 2.12: Flujo de datos en la unidad de tiempo  $t$  para la Arquitectura de una Red Neuronal Recurrente (RNN).

El *campo receptivo* crece con  $K$ ,  $d_{dil}$  y la profundidad de la red; el *padding* controla el tamaño de salida; el *stride* ajusta la resolución y el costo. Esta combinación hace que las convoluciones 1D sean especialmente eficaces para patrones locales y de mediano alcance en secuencias largas.

### Extensión al caso 2D

La convolución 2D es una extensión natural de la 1D cuando los datos varían en dos ejes (alto  $H$  y ancho  $W$ ). Con entrada  $\mathbf{X} \in \mathbb{R}^{D \times H \times W}$  y filtros  $W \in \mathbb{R}^{D' \times D \times K \times L}$ , la salida es  $\mathbf{Y} \in \mathbb{R}^{D' \times H' \times W'}$  con

$$Y_{d',i,j} = \sum_{d=1}^D \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} W_{d',d,k,l} X_{d,i+k,j+l} + b_{d'}$$

donde  $H'$  y  $W'$  se determinan análogamente al caso 1D (considerando los parámetros de *padding*, *stride* y *dilatación* en ambas direcciones). Aunque su uso es ubicuo en visión por computador, su estructura matemática es análoga a la del caso 1D y todas las consideraciones anteriores se trasladan componente a componente.

### Redes LSTM (Long Short-Term Memory)

Las Redes Neuronales Recurrentes (RNN) fueron diseñadas específicamente para procesar datos secuenciales, como series temporales o texto. A diferencia de las redes tradicionales, las RNN mantienen una memoria interna que les permite capturar dependencias temporales a través de conexiones cíclicas, donde, aparte de la entrada  $x_t$ , la cual es alimentada a la red paso a paso, la salida de una neurona en un paso dado, conocido como estado oculto  $h_t$ , puede retroalimentarse como entrada en el siguiente paso. Esta arquitectura se ilustra en la Fig. 2.12, donde se muestra la forma compartida (a la izquierda) y la forma desembuelta (a la derecha) de una RNN. En la forma compartida, se observa que la misma estructura de red se aplica en cada paso temporal, mientras que en la forma desembuelta se visualiza cómo los inputs  $x_t$  se procesan a través de las capas recurrentes y se retroalimentan al paso siguiente a través de las salidas  $h_t$ . Esto hace que las RNN sean ideales para modelar datos que dependen de eventos pasados, como el análisis de secuencias temporales o en nuestro caso para la predicción de comportamientos dinámicos.

Las redes recurrentes simples (RNN) actualizan un estado oculto  $\mathbf{h}_t$  en función de la entrada y del estado previo:

$$\mathbf{h}_t = \sigma(\mathbf{W}_h \mathbf{h}_{t-1} + \mathbf{W}_x \mathbf{x}_t + \mathbf{b}).$$

Si bien este esquema captura dependencias temporales, las RNN tradicionales sufren de un problema conocido como *desvanecimiento y explosión del gradiente*. Principalmente el fenómeno de desvanecimiento ocurre cuando los gradientes de las funciones de pérdida que se retropropagan a través de las capas en la RNN se vuelven muy pequeños a medida que retroceden en el tiempo. Como resultado, las actualizaciones de los pesos se vuelven insignificantes para las capas más anteriores de la red, lo que dificulta el aprendizaje de dependencias a largo plazo. Este problema es especialmente problemático en tareas donde las relaciones temporales pueden extenderse a lo largo de secuencias largas, como en el procesamiento de texto o la predicción de series temporales.

Para superar las limitaciones del desvanecimiento del gradiente, se desarrollaron las *Long Short-Term Memory (LSTM)*, una variante avanzada de las RNN. Las LSTM están diseñadas con una estructura interna más compleja que incluye *puertas* (input o entrada, forget u olvido y output o salida) que controlan el flujo de información a lo largo del tiempo y permiten retener o desechar información de manera más eficaz. Gracias a este mecanismo, las LSTM son capaces de aprender y mantener dependencias a largo plazo, siendo extremadamente útiles en tareas donde la memoria temporal es crucial, como el procesamiento de lenguaje natural, la predicción de secuencias o la modelación de dinámicas estructurales en sistemas complejos.

Las LSTM introducen un *estado de memoria*  $\mathbf{x}_t$  y un sistema de compuertas que regulan el flujo de información, mitigando estos problemas.

En cada paso temporal  $t$ , con entrada  $\mathbf{x}_t \in \mathbb{R}^f$  y estado oculto previo  $\mathbf{h}_{t-1} \in \mathbb{R}^u$ :

$$\begin{aligned} \mathbf{i}_t &= \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i) && \text{(compuerta de entrada),} \\ \mathbf{f}_t &= \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f) && \text{(compuerta de olvido),} \\ \mathbf{o}_t &= \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o) && \text{(compuerta de salida),} \\ \tilde{\mathbf{c}}_t &= \tanh(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-1} + \mathbf{b}_c) && \text{(estado candidato).} \end{aligned}$$

Con estas cantidades se actualiza el estado de memoria y el estado oculto:

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t, \quad \mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t).$$

donde la salida  $\mathbf{h}_t \in \mathbb{R}^u$  puede interpretarse como un vector de  $u$  representaciones o estados ocultos que codifican la información de la secuencia hasta el instante  $t$ .

La idea es que  $\mathbf{f}_t$  controla cuánto de la memoria pasada se conserva,  $\mathbf{i}_t$  regula cuánto de la nueva información se incorpora, y  $\mathbf{o}_t$  decide cuánto del contenido de la memoria se expone como salida en  $\mathbf{h}_t$ . En la Fig. 2.13 se puede observar cómo fluyen los datos a través de las diferentes compuertas mencionadas anteriormente.

Una LSTM individual transforma la secuencia de entrada  $\{\mathbf{x}_t\}_{t=1}^T$  en una secuencia de estados ocultos  $\{\mathbf{h}_t\}_{t=1}^T$ , que pueden ser interpretados como representaciones de  $u$  dimensiones en cada paso temporal. Esto se puede observar en la Fig. 2.14. Para incrementar la capacidad de modelado, es común *apilar* varias capas LSTM en forma jerárquica. En una LSTM apilada con  $L$  capas, la salida oculta de la capa  $l$  en el instante  $t$  se convierte en la entrada de la capa  $l + 1$  en el mismo instante temporal:

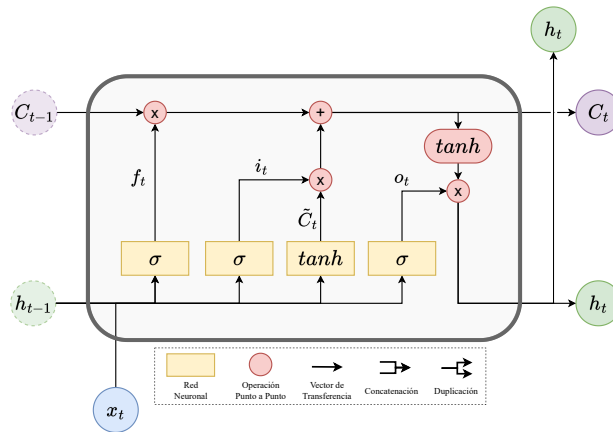


Figura 2.13: Flujo de datos en la unidad de tiempo  $t$  de una unidad de LSTM. La puerta de olvido y la celda de memoria evitan problemas de desvanecimiento o explosión de gradiente.

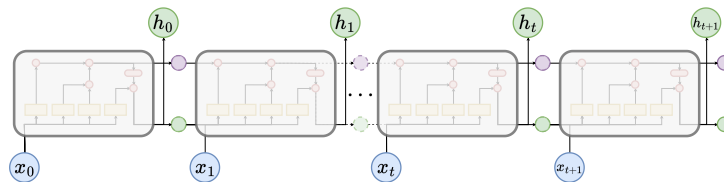


Figura 2.14: Esquema básico de una LSTM desenrollada, donde el estado oculto  $h_t$  se calcula de manera recurrente utilizando las entradas  $x_t$  y el estado oculto anterior  $h_{t-1}$ .

$$\mathbf{h}_t^{(l)}, \mathbf{c}_t^{(l)} = \text{LSTM}^{(l)}(\mathbf{h}_{t-1}^{(l)}, \mathbf{c}_{t-1}^{(l)}, \mathbf{x}_t^{(l)}), \quad \mathbf{x}_t^{(l)} = \begin{cases} \mathbf{x}_t, & l = 1, \\ \mathbf{h}_t^{(l-1)}, & l > 1. \end{cases}$$

De este modo, la primera capa recibe la señal original  $\mathbf{x}_t$ , mientras que las capas superiores reciben como entrada representaciones más abstractas generadas por las capas inferiores. El estado total de la red en el tiempo  $t$  está dado por el conjunto  $\{(\mathbf{h}_t^{(l)}, \mathbf{c}_t^{(l)})\}_{l=1}^L$ , en la Fig. 2.15 se presenta este esquema de múltiples capas LSTM.

La apilación incrementa la profundidad en la dimensión de características, de modo que las capas inferiores capturan patrones de bajo nivel (por ejemplo, cambios rápidos en la señal), mientras que las capas superiores integran información en horizontes temporales más amplios. En consecuencia, las LSTM apiladas tienden a mejorar la capacidad de generalización del modelo, aunque a costa de un mayor número de parámetros y un incremento en el costo computacional.

En conjunto, la arquitectura LSTM resulta especialmente eficaz para tareas de predicción secuencial y modelado de series temporales, pues permite retener información relevante durante largos intervalos y manejar dependencias de distinto alcance de manera flexible.

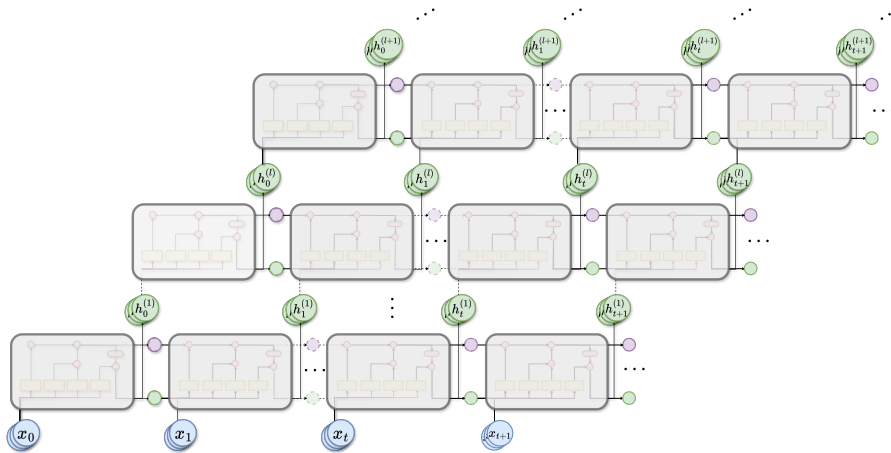


Figura 2.15: Esquema de una Red LSTM profunda con múltiples capas ocultas, donde las entradas  $\mathbf{x}_t$  se procesan a través de varias capas de estados ocultos  $\mathbf{h}_t^{(1)}, \dots, \mathbf{h}_t^{(l)}, \mathbf{h}_t^{(l+1)}$ , para modelar secuencias complejas.

**Transformers y mecanismos de atención**

Más recientemente, los Transformers (Vaswani et al., s.f.) reemplazan la recurrencia por *mecanismos de autoatención*, donde cada elemento de la secuencia pondera su relación con todos los demás mediante productos escalares entre proyecciones denominadas  $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ . La salida de la autoatención para un bloque es

$$\text{Att}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right) \mathbf{V}.$$

De este modo, se capturan dependencias de largo alcance sin necesidad de desenrollar la dinámica temporal. Aunque la complejidad del cálculo es cuadrática en la longitud de la secuencia, diversas variantes eficientes aproximan la atención completa con mecanismos locales o de baja dimensionalidad proyectada.

**Discusión**

Cada una de estas arquitecturas representa un compromiso entre expresividad, costo computacional y adecuación a la estructura de los datos. Los MLP son simples pero costosos cuando las entradas son largas; las CNN explotan la localidad y son altamente escalables; las RNN son naturales para secuencias aunque difíciles de entrenar en horizontes largos; y los Transformers dominan actualmente en tareas secuenciales complejas gracias a su capacidad para modelar dependencias globales, sin embargo, a costa de un mayor consumo de memoria y tiempo de cómputo.

**2.7.3. Protocolos de partición y normalización de datos**

El desempeño de un modelo de aprendizaje profundo no depende únicamente de su arquitectura, sino también de la manera en que se preparan y utilizan los datos. Para evaluar de forma objetiva la capacidad de

generalización de la red, es fundamental dividir el conjunto de datos disponible en particiones con propósitos diferenciados: entrenamiento, validación y prueba.

### Conjunto de entrenamiento

El conjunto de entrenamiento (*training set*) se emplea directamente para ajustar los parámetros  $\theta$  de la red mediante un algoritmo de optimización. Cada ejemplo  $(\mathbf{x}_i, \mathbf{y}_i)$  contribuye al cálculo de la pérdida  $\mathcal{L}$  y, por medio de la retropropagación, a la actualización de los pesos. Un tamaño suficiente y una diversidad adecuada en este conjunto son esenciales para que el modelo aprenda patrones representativos del dominio de interés.

### Conjunto de validación

El conjunto de validación (*validation set*) se utiliza durante el proceso de entrenamiento para estimar el rendimiento del modelo en datos no vistos. Sirve para ajustar *hiperparámetros* (tasa de aprendizaje, número de capas, regularización, etc.), así como para implementar criterios de *early stopping*, es decir, detener el entrenamiento cuando la pérdida de validación deja de mejorar y comienza a evidenciar sobreajuste. De este modo, la validación proporciona una medida indirecta de la capacidad de generalización.

### Conjunto de prueba

El conjunto de prueba (*test set*) se reserva estrictamente para la evaluación final del modelo, una vez completado el entrenamiento y la selección de hiperparámetros. Su propósito es ofrecer una estimación imparcial del desempeño en datos completamente nuevos, evitando que la evaluación se vea contaminada por decisiones tomadas durante la fase de entrenamiento.

### Validación cruzada

Cuando la cantidad de datos es limitada, la partición fija en entrenamiento/validación/prueba puede producir estimaciones inestables. En tales casos se recurre a la *validación cruzada* (*cross-validation*), que consiste en dividir el conjunto de datos (exceptuando el bloque reservado para prueba) en  $k$  particiones (*folds*) de tamaño similar. El entrenamiento se repite  $k$  veces, utilizando en cada iteración  $k - 1$  particiones para entrenar y la restante para validar. El rendimiento final se estima como el promedio sobre las  $k$  iteraciones:

$$\widehat{R} = \frac{1}{k} \sum_{j=1}^k R^{(j)},$$

donde  $R^{(j)}$  es la métrica de desempeño obtenida en la  $j$ -ésima validación.

Es importante subrayar que el conjunto de prueba (*test set*) no participa en la validación cruzada: permanece congelado hasta la etapa final del experimento. Una vez seleccionados los hiperparámetros y la configuración del modelo mediante validación cruzada, se reentrena con la totalidad de los datos de entrenamiento–validación y sólo entonces se evalúa en el conjunto de prueba, obteniendo así una estimación imparcial de la capacidad de generalización.

La validación cruzada reduce la varianza de la estimación y permite aprovechar mejor los datos disponibles, aunque a costa de un mayor tiempo de cómputo debido a la necesidad de entrenar el modelo múltiples veces. En entornos de aprendizaje profundo, donde los costos de entrenamiento son elevados, se utilizan a menudo versiones simplificadas, como la validación cruzada con  $k = 5$  o  $k = 10$  folds, o bien estrategias anidadas (*nested cross-validation*) para ajustar hiperparámetros de manera más rigurosa.

### Normalización y preprocesamiento

Para facilitar el aprendizaje y mejorar la estabilidad numérica, es habitual aplicar procesos de normalización o estandarización a los datos de entrada. Entre las técnicas más comunes se encuentran:

- **Estandarización:** transformar cada variable para que tenga media cero y desviación estándar unitaria.
- **Normalización min–max:** reescalar las variables a un rango acotado, por ejemplo  $[0, 1]$ .
- **Normalización por canal o característica:** en datos multivariados o con múltiples sensores, se normaliza cada canal por separado para compensar diferencias de escala.

Estos procedimientos buscan que todas las características contribuyan de manera equilibrada al proceso de optimización, evitando que parámetros asociados a variables de gran escala dominen las actualizaciones de gradiente.

### Discusión

En conjunto, la correcta partición de los datos y la aplicación de técnicas de normalización constituyen pasos fundamentales en el flujo de trabajo del aprendizaje profundo. Una mala práctica, como evaluar sobre el mismo conjunto usado para entrenar, conduce a estimaciones optimistas y poco realistas del rendimiento. Por el contrario, la separación clara entre entrenamiento, validación y prueba, junto con la validación cruzada y un preprocesamiento adecuado, garantiza que los resultados obtenidos reflejen mejor la capacidad real del modelo para generalizar a situaciones nuevas.

#### 2.7.4. Indicadores de desempeño y métricas de evaluación

La evaluación cuantitativa del desempeño del modelo requiere definir métricas que cuantifiquen la discrepancia entre las predicciones  $\hat{y}_t$  y las señales de referencia  $y_t$  a lo largo del tiempo. En el caso de series temporales, dichas métricas deben reflejar tanto la precisión global del ajuste como aspectos específicos de la dinámica, tales como la amplitud, el error acumulado o la posible deriva residual. Todas las métricas aquí presentadas se calculan sobre el conjunto de prueba, considerando la secuencia completa de salida.

#### Métricas básicas

Las métricas de error en el dominio temporal permiten evaluar la fidelidad del modelo al reproducir las secuencias de respuesta observadas. Sea  $e_t = \hat{y}_t - y_t$  el error puntual en el instante  $t$ , definido como la

diferencia entre la predicción y el valor de referencia. A partir de esta serie de errores se derivan los indicadores más comunes de desempeño.

La **raíz del error cuadrático medio (RMSE)** se define como

$$\text{RMSE} = \sqrt{\frac{1}{T} \sum_{t=1}^T e_t^2}, \quad \sigma_{\text{RMSE}} = \sqrt{\frac{1}{T-1} \sum_{t=1}^T (e_t^2 - \overline{e^2})^2}.$$

Esta métrica representa la magnitud promedio del error cuadrático y penaliza con mayor intensidad las desviaciones de gran amplitud, siendo por tanto sensible a picos locales o errores aislados de alta magnitud. La desviación asociada,  $\sigma_{\text{RMSE}}$ , cuantifica la variabilidad temporal de los errores cuadráticos, proporcionando una medida de la consistencia del ajuste a lo largo de la serie.

De manera complementaria, el **error absoluto medio (MAE)** evalúa la magnitud promedio del error sin ponderar los valores extremos, según

$$\text{MAE} = \frac{1}{T} \sum_{t=1}^T |e_t|, \quad \sigma_{\text{MAE}} = \sqrt{\frac{1}{T-1} \sum_{t=1}^T (|e_t| - \overline{|e|})^2}.$$

A diferencia del RMSE, el MAE es menos sensible a errores puntuales de gran magnitud y entrega una estimación más robusta del error medio global. Su desviación  $\sigma_{\text{MAE}}$  refleja la dispersión temporal del ajuste.

Cuando las señales presentan distintas escalas o unidades, resulta conveniente emplear las versiones normalizadas de estas métricas, expresadas en términos del rango dinámico de la señal de referencia:

$$\text{NRMSE} = \frac{\text{RMSE}}{y_{\text{máx}} - y_{\text{mín}}}, \quad \text{NMAE} = \frac{\text{MAE}}{y_{\text{máx}} - y_{\text{mín}}},$$

con desviaciones análogas

$$\sigma_{\text{NRMSE}} = \frac{\sigma_{\text{RMSE}}}{y_{\text{máx}} - y_{\text{mín}}}, \quad \sigma_{\text{NMAE}} = \frac{\sigma_{\text{MAE}}}{y_{\text{máx}} - y_{\text{mín}}}.$$

Estas versiones permiten comparar el desempeño del modelo entre señales de diferente amplitud, expresando el error en forma adimensional y relativa al rango de variación de la respuesta.

Otra métrica de uso extendido es el **error cuadrático relativo (L2RE)**, definido como

$$\text{L2RE} = \frac{\|\hat{\mathbf{y}} - \mathbf{y}\|_2}{\|\mathbf{y}\|_2} = \frac{\sqrt{\sum_{t=1}^T e_t^2}}{\sqrt{\sum_{t=1}^T y_t^2}}.$$

Esta relación compara la energía del error con la energía total de la señal real, y su valor adimensional facilita la comparación entre secuencias de distinta escala. Un valor  $\text{L2RE} \ll 1$  indica una predicción de alta fidelidad relativa. De forma equivalente, el **error energético normalizado (NEE)** se obtiene como el cuadrado del

anterior,

$$\text{NEE} = (\text{L2RE})^2 = \frac{\sum_{t=1}^T e_t^2}{\sum_{t=1}^T y_t^2},$$

lo que representa la fracción de energía de la señal real no capturada por el modelo.

El **error normalizado punto a punto** permite analizar la distribución estadística del error relativo de manera adimensional:

$$\varepsilon_t = \frac{e_t}{\max_t |y_t|}, \quad \bar{\varepsilon} = \frac{1}{T} \sum_{t=1}^T \varepsilon_t, \quad \sigma_\varepsilon = \sqrt{\frac{1}{T-1} \sum_{t=1}^T (\varepsilon_t - \bar{\varepsilon})^2}.$$

Este indicador es especialmente útil para evaluar la consistencia local de las predicciones.

En un sentido global, el **coeficiente de determinación**  $R^2$  mide la proporción de la varianza de la señal real explicada por el modelo:

$$R^2 = 1 - \frac{\sum_{t=1}^T (\hat{y}_t - y_t)^2}{\sum_{t=1}^T (y_t - \bar{y})^2}.$$

Valores cercanos a uno indican una buena correspondencia entre la respuesta observada y la predicha. Finalmente, el **coeficiente de correlación de Pearson** cuantifica la linealidad entre ambas señales:

$$r = \frac{\sum_{t=1}^T (y_t - \bar{y})(\hat{y}_t - \bar{\hat{y}})}{\sqrt{\sum_{t=1}^T (y_t - \bar{y})^2 \sum_{t=1}^T (\hat{y}_t - \bar{\hat{y}})^2}},$$

oscilando entre  $-1$  (correlación inversa perfecta) y  $1$  (correlación directa perfecta).

### Agregación sobre múltiples muestras

Cuando el modelo se evalúa sobre un conjunto de  $N$  secuencias o muestras independientes, cada métrica  $M_i$  (por ejemplo,  $\text{RMSE}_i$  o  $\text{MAE}_i$ ) se calcula individualmente para cada muestra  $i$ . El desempeño global se caracteriza mediante el promedio y la desviación muestral:

$$\bar{M} = \frac{1}{N} \sum_{i=1}^N M_i, \quad \sigma_M = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (M_i - \bar{M})^2}.$$

El promedio  $\bar{M}$  representa el desempeño medio del modelo en el conjunto de prueba, mientras que  $\sigma_M$  cuantifica la variabilidad del error entre diferentes realizaciones, aportando una medida de la estabilidad del modelo frente a distintas condiciones de entrada.

### Métricas para curvas de histéresis bidimensionales

En sistemas estructurales no-lineales, la respuesta histerética se representa mediante curvas fuerza–desplazamiento, donde cada punto temporal  $\mathbf{y}_t = [d_t, f_t]^\top$  describe de manera simultánea el desplazamiento  $d_t$  y

la fuerza  $f_t$ . La predicción del modelo genera una secuencia análoga  $\widehat{\mathbf{y}}_t = [\widehat{d}_t, \widehat{f}_t]^\top$ , por lo que la evaluación de desempeño debe considerar el error vectorial conjunto entre ambas componentes.

El error instantáneo se define como

$$\mathbf{e}_t = \widehat{\mathbf{y}}_t - \mathbf{y}_t = \begin{bmatrix} \widehat{d}_t - d_t \\ \widehat{f}_t - f_t \end{bmatrix}, \quad \|\mathbf{e}_t\|_2 = \sqrt{(\widehat{d}_t - d_t)^2 + (\widehat{f}_t - f_t)^2}.$$

El valor escalar  $\|\mathbf{e}_t\|_2$  representa la distancia euclídea entre el punto predicho y el punto real en el plano fuerza–desplazamiento, constituyendo la base de las métricas bidimensionales.

La **magnitud absoluta media bidimensional (MAE<sub>2D</sub>)** se define como

$$\text{MAE}_{2D} = \frac{1}{T} \sum_{t=1}^T \|\mathbf{e}_t\|_2, \quad \sigma_{\text{MAE}_{2D}} = \sqrt{\frac{1}{T-1} \sum_{t=1}^T (\|\mathbf{e}_t\|_2 - \overline{\|\mathbf{e}\|_2})^2}.$$

Esta métrica mide la distancia promedio entre las trayectorias predicha y real, sin depender de la dirección de los vectores de error. Su desviación  $\sigma_{\text{MAE}_{2D}}$  refleja la variabilidad temporal del ajuste a lo largo del ciclo histerético.

De manera análoga, la **raíz del error cuadrático medio bidimensional (RMSE<sub>2D</sub>)** combina las discrepancias cuadráticas de ambas componentes:

$$\text{RMSE}_{2D} = \sqrt{\frac{1}{T} \sum_{t=1}^T (\widehat{d}_t - d_t)^2 + (\widehat{f}_t - f_t)^2},$$

$$\sigma_{\text{RMSE}_{2D}} = \sqrt{\frac{1}{T-1} \sum_{t=1}^T \left( [(\widehat{d}_t - d_t)^2 + (\widehat{f}_t - f_t)^2] - \overline{(\widehat{d} - d)^2 + (\widehat{f} - f)^2} \right)^2}.$$

A diferencia del MAE<sub>2D</sub>, esta métrica penaliza con mayor intensidad los errores de gran magnitud, proporcionando una estimación global de la discrepancia geométrica entre las curvas reales y predichas.

Dado que las escalas de desplazamiento y fuerza pueden diferir, se define un rango efectivo bidimensional que permite obtener métricas normalizadas:

$$R_{2D} = \sqrt{(\text{máx } d_t - \text{mín } d_t)^2 + (\text{máx } f_t - \text{mín } f_t)^2},$$

a partir del cual se calculan el **error absoluto medio normalizado** y la **raíz del error cuadrático medio normalizado**:

$$\text{NMAE}_{2D} = \frac{\text{MAE}_{2D}}{R_{2D}}, \quad \text{NRMSE}_{2D} = \frac{\text{RMSE}_{2D}}{R_{2D}}.$$

Estas métricas expresan el error en términos relativos al dominio de variación conjunta fuerza–desplazamiento, permitiendo comparar respuestas de distinta amplitud.

Para evaluar la fracción de energía no capturada por el modelo, se emplea el **error cuadrático relativo bidimensional (L2RE<sub>2D</sub>)**, definido como

$$\text{L2RE}_{2D} = \frac{\|\widehat{\mathbf{Y}} - \mathbf{Y}\|_2}{\|\mathbf{Y}\|_2} = \frac{\sqrt{\sum_{t=1}^T [(\widehat{d}_t - d_t)^2 + (\widehat{f}_t - f_t)^2]}}{\sqrt{\sum_{t=1}^T (d_t^2 + f_t^2)}}.$$

Este indicador adimensional cuantifica la fracción de energía total de la respuesta bidimensional que no es explicada por el modelo, de manera análoga al L2RE unidimensional.

Finalmente, el **error de área histerética** constituye una métrica energética directa, asociada a la capacidad del sistema para disipar energía durante los ciclos de carga y descarga. La energía disipada corresponde al área encerrada por la curva fuerza–desplazamiento, cuya estimación discreta se obtiene como:

$$A_{\text{real}} = \sum_{t=1}^{T-1} \frac{1}{2} (f_{t+1} + f_t)(d_{t+1} - d_t), \quad A_{\text{pred}} = \sum_{t=1}^{T-1} \frac{1}{2} (\widehat{f}_{t+1} + \widehat{f}_t)(\widehat{d}_{t+1} - \widehat{d}_t),$$

y el error relativo asociado se calcula como

$$\text{AreaError} = \frac{|A_{\text{pred}} - A_{\text{real}}|}{|A_{\text{real}}|}.$$

Esta métrica cuantifica la discrepancia en la energía disipada entre la curva real y la predicha, siendo especialmente sensible a diferencias en la amplitud, fase o distorsión del ciclo histerético.

### Agregación entre muestras

De manera análoga al caso unidimensional, cada métrica bidimensional se calcula para cada una de las  $N$  muestras o ensayos. El promedio y la desviación muestral se definen como:

$$\overline{M}_{2D} = \frac{1}{N} \sum_{i=1}^N M_{2D,i}, \quad \sigma_{M_{2D}} = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (M_{2D,i} - \overline{M}_{2D})^2}.$$

### Indicadores gráficos de desempeño

Además de las métricas numéricas, es habitual complementar con indicadores gráficos:

- **Predicción vs. serie real:** graficar  $\widehat{y}_t$  y  $y_t$  en el tiempo para evaluar visualmente la calidad del ajuste.
- **Error temporal agregado:** representar  $\frac{1}{N} \sum_{i=1}^N |\widehat{y}_{i,t} - y_{i,t}|$  como función de  $t$ , mostrando cómo varía el error promedio en el tiempo.
- **Diagramas de caja (boxplots):** considerar el error promedio de cada serie  $\{E_i\}_{i=1}^N$  como un valor individual, y representar su distribución mediante gráficos de caja, lo que permite identificar dispersión y valores atípicos.

## Discusión

El uso combinado de métricas globales (RMSE, MAE, L2RE) y representaciones gráficas proporciona una visión más completa del desempeño del modelo. Mientras que los indicadores numéricos permiten comparaciones objetivas, las visualizaciones ayudan a identificar patrones de error que podrían pasar desapercibidos en un análisis puramente cuantitativo.

## 2.8. Redes neuronales informadas por la física (PINNs)

El paradigma de las *Physics-Informed Neural Networks* (PINNs) surgió como respuesta a la necesidad de integrar información proveniente de las leyes físicas en el entrenamiento de redes neuronales. Tradicionalmente, los modelos de aprendizaje profundo se ajustan únicamente a partir de datos observados. Sin embargo, en muchas aplicaciones de ingeniería y ciencias aplicadas, además de los datos experimentales, se dispone de conocimientos previos en forma de ecuaciones diferenciales ordinarias (EDO) o parciales (EDP) que describen el sistema subyacente.

### Formulación general

La idea central de los PINNs es incorporar estas ecuaciones como restricciones adicionales durante el entrenamiento, forzando a la red a producir predicciones que respeten la física del problema. Supongamos que la dinámica de un sistema está gobernada por una ecuación diferencial general

$$\mathcal{N}[u](\mathbf{x}, t) = 0,$$

donde  $\mathcal{N}$  es un operador diferencial y  $u(\mathbf{x}, t)$  la solución desconocida. Una red neuronal  $f_{\theta}(\mathbf{x}, t)$  se utiliza como aproximador de  $u$ , y la función de pérdida se construye con múltiples términos:

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{datos}} + \lambda_{\text{fs}} \mathcal{L}_{\text{física}} + \lambda_{\text{bc}} \mathcal{L}_{\text{condiciones}}.$$

**Término de datos.** Mide el error en los puntos de entrenamiento donde se conocen observaciones:

$$\mathcal{L}_{\text{datos}} = \frac{1}{N_d} \sum_{i=1}^{N_d} |f_{\theta}(\mathbf{x}_i, t_i) - u_i|^2.$$

**Término físico.** Impone que la red satisfaga la ecuación diferencial en un conjunto de puntos de colocación  $\{(\mathbf{x}_j, t_j)\}_{j=1}^{N_c}$ :

$$\mathcal{L}_{\text{física}} = \frac{1}{N_c} \sum_{j=1}^{N_c} |\mathcal{N}[f_{\theta}](\mathbf{x}_j, t_j)|^2.$$

**Condiciones iniciales y de borde.** En problemas bien planteados, además de la ecuación diferencial, se requieren condiciones de borde y/o condiciones iniciales. Estas se incorporan como términos adicionales en la

pérdida:

$$\mathcal{L}_{\text{condiciones}} = \frac{1}{N_{ic}} \sum_{i=1}^{N_{ic}} |f_{\theta}(\mathbf{x}_i, 0) - u_0(\mathbf{x}_i)|^2 + \frac{1}{N_{bc}} \sum_{j=1}^{N_{bc}} |f_{\theta}(\mathbf{x}_j, t_j) - g(\mathbf{x}_j, t_j)|^2,$$

donde  $u_0$  representa las condiciones iniciales y  $g$  las condiciones de contorno prescritas en el dominio espacial.

De este modo, las PINNs aprenden una solución que no sólo ajusta los datos observados, sino que también satisface la ecuación diferencial y respeta las condiciones físicas de frontera, lo que les permite aproximar soluciones únicas incluso en escenarios con datos limitados o ruidosos.

### Variantes solver-informed y guided-informed

Además de las PINNs clásicas, han surgido enfoques relacionados que incorporan el conocimiento físico de maneras distintas. Entre ellos, se encuentran las redes *guided-informed* y *solver-informed*.

Las *guided-informed* incorporan la física mediante términos adicionales en la función de pérdida, como en las PINNs, pero sin modificar la arquitectura de la red. En contraste, las *solver-informed* integran explícitamente estructuras inspiradas en métodos numéricos de resolución de ecuaciones diferenciales. En este caso, la arquitectura de la red se diseña para imitar el comportamiento de un solucionador, por ejemplo representando pasos de integración temporal o esquemas iterativos dentro de la red.

Esta diferencia implica que mientras las *guided-informed* se limitan a guiar el aprendizaje con restricciones físicas, las *solver-informed* adoptan una forma híbrida entre modelo numérico y red neuronal, lo que puede mejorar la estabilidad y la capacidad de generalización en sistemas dinámicos complejos.

## 2.9. Modelos PINN de referencia en dinámica estructural

En esta sección se presentan las bases teóricas de los modelos informados por la física que se adoptan como referencia: las *Physics-guided Convolutional Neural Network* (PhyCNN-MO/PhyCNN-SO) y las arquitecturas *Physics-informed multi-LSTM* (PhyLSTM2/PhyLSTM3), ambas propuestas por Zhang et al. (2020a) y Zhang et al. (2020b). El desarrollo de esta sección se centra en la descripción conceptual y estructural de dichos modelos, mientras que su implementación práctica y la evaluación comparativa se discuten en el Capítulo 3.

### 2.9.1. PhyCNN: red convolucional guiada por la física

La arquitectura PhyCNN (Zhang et al., 2020a) busca construir un metamodelo secuencial que, entrenado con un conjunto limitado de pares entrada–salida (acelerogramas y respuestas estructurales medidas o simuladas), sea capaz de reproducir la cinemática estructural bajo excitaciones sísmicas sin requerir un modelo explícito de elementos finitos durante la inferencia. El objetivo principal del modelo es estimar el comportamiento dinámico no-lineal de una estructura sometida a un movimiento sísmico, empleando una red neuronal convolucional unidimensional (CNN) que aprende las relaciones físicas subyacentes entre la excitación y la respuesta estructural.

La arquitectura se fundamenta en la incorporación explícita de las leyes físicas, en particular la ecuación del movimiento, dentro del proceso de entrenamiento. Este enfoque da origen a una *Physics-Informed Neural*

*Network* (PINN), en la cual el aprendizaje combina información proveniente de datos medidos o simulados con restricciones físicas que aseguran la coherencia del modelo respecto del sistema dinámico real. La información física se introduce como una restricción sobre las salidas de la red: las predicciones de desplazamiento  $\hat{\mathbf{u}}$  y sus derivadas temporales (obtenidas mediante un mecanismo de diferenciación en el grafo computacional, explicado más adelante en esta misma sección) deben satisfacer, en cada punto de colocalización, el residuo de la ecuación de movimiento normalizada por la masa:

$$r(t) = \hat{a}(t) + \hat{g}(t) + a_g(t) = 0, \quad (2.31)$$

donde  $\hat{a}(t)$  representa la segunda derivada temporal de la predicción de desplazamiento  $\hat{u}(t)$ , o la primera derivada de la velocidad predicha  $\hat{v}(t)$ , obtenida mediante la aplicación del operador discreto de diferencias finitas  $\Phi_t \in \mathbb{R}^{T \times T}$ . Por otro lado,  $\hat{g}(t) = \hat{f}_{\text{int}}(t)/m$  denota la fuerza interna normalizada aprendida por la red, que integra los efectos histeréticos, restauradores y de amortiguamiento (véase la Sección 2.2). El proceso de entrenamiento se basa en una función de pérdida compuesta por dos términos: el ajuste a los datos medidos/simulados,  $\mathcal{L}_{\text{data}}$ , y las penalizaciones físicas,  $\mathcal{L}_{\text{fis}}$ . Siguiendo la metodología propuesta por los autores, ambos términos se ponderan de manera equivalente, de modo que el modelo busca simultáneamente minimizar el error respecto a los datos y cumplir con las restricciones físicas impuestas.

Dentro de este esquema general se plantean dos variantes del modelo PhyCNN, que difieren en el número de salidas generadas por la red y en la forma en que incorporan la información física durante el entrenamiento. La primera, denominada **PhyCNN-MO** (*Multiple-Output*), produce de manera simultánea las series de desplazamiento, velocidad y fuerza interna normalizada, representadas como  $\hat{\mathbf{u}}$ ,  $\hat{\mathbf{v}}$  y  $\hat{\mathbf{g}}$ . Esta formulación busca integrar explícitamente las relaciones físicas entre dichas magnitudes, con el propósito de mantener coherencia entre las variables cinemáticas y dinámicas. Sin embargo, esta integración implica una arquitectura más extensa y un mayor número de parámetros entrenables, lo que puede incrementar la complejidad del ajuste y la sensibilidad del modelo a la cantidad y calidad de los datos disponibles.

La segunda variante, **PhyCNN-SO** (*Single-Output*), se centra en predecir únicamente la serie de desplazamientos  $\hat{\mathbf{u}}$ , obteniendo las derivadas temporales mediante el operador discreto de diferenciación  $\Phi_t$  dentro del grafo computacional. Este enfoque reduce significativamente la dimensionalidad de la salida y simplifica el proceso de entrenamiento, aunque su desempeño depende en gran medida de la precisión y estabilidad del esquema numérico de diferenciación utilizado. En consecuencia, la simplificación estructural puede traducirse en una pérdida parcial de información física explícita en comparación con la variante multisalida.

Ambas arquitecturas comparten el principio de incorporar las leyes físicas del sistema estructural como guía del proceso de aprendizaje, empleando redes convolucionales unidimensionales para capturar las dependencias temporales de las señales. No obstante, la efectividad de esta incorporación depende tanto del diseño del mecanismo de diferenciación como de la capacidad del modelo para equilibrar el ajuste a los datos con el cumplimiento de las restricciones físicas. En las subsecciones siguientes se analizan en detalle las arquitecturas PhyCNN-MO y PhyCNN-SO, así como el mecanismo de diferenciación temporal que permite aproximar las derivadas de las señales dentro del modelo.

### Método de diferenciación temporal

Dado que las restricciones físicas impuestas por el residuo de la ecuación de movimiento (Ecuación 2.31) requieren el cálculo de derivadas temporales de las variables cinemáticas (en particular  $\hat{\mathbf{u}}$  y  $\ddot{\mathbf{u}}$ ), el modelo PhyCNN incorpora un mecanismo de diferenciación numérica dentro del propio grafo computacional. Este mecanismo permite obtener derivadas temporales de las señales de salida de la red mediante un operador lineal discreto, con el objetivo de mantener la coherencia entre las predicciones del modelo y las restricciones físicas que definen la ecuación del movimiento. Sin embargo, la precisión de las derivadas obtenidas depende directamente del esquema numérico implementado y del tamaño del paso temporal empleado en la discretización.

Las derivadas temporales se aproximan mediante un esquema mixto de diferencias finitas de segundo orden, que combina aproximaciones unilaterales en los extremos del dominio temporal y centradas en los puntos interiores. Este enfoque presenta, bajo condiciones de paso temporal uniforme, un orden de exactitud teórico  $O(\Delta t^2)$  y permite un cálculo computacionalmente eficiente dentro del grafo de entrenamiento. No obstante, la calidad de las estimaciones puede verse afectada por la amplificación de errores en los bordes y por la sensibilidad del operador a señales ruidosas o con alta frecuencia, lo que introduce un compromiso entre simplicidad y fidelidad numérica en la obtención de las derivadas.

Sea  $\mathbf{u} \in \mathbb{R}^{B \times T \times F}$  el conjunto de señales de entrada o salida, donde  $B$  representa el número de muestras del lote,  $T$  el número de instantes temporales y  $F$  el número de características o grados de libertad. Para un paso temporal uniforme  $\Delta t$ , la derivada discreta se define como:

$$\dot{\mathbf{u}} = \Phi_t \mathbf{u}, \quad (2.32)$$

donde  $\Phi_t \in \mathbb{R}^{T \times T}$  corresponde al operador discreto de diferencias finitas temporales. Su estructura tridiagonal refleja la dependencia local entre instantes consecutivos y puede expresarse como:

$$\Phi_t = \frac{1}{\Delta t} \begin{bmatrix} -\frac{3}{2} & 2 & -\frac{1}{2} & 0 & \cdots & 0 \\ -\frac{1}{2} & 0 & \frac{1}{2} & 0 & \cdots & 0 \\ 0 & -\frac{1}{2} & 0 & \frac{1}{2} & \cdots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & -\frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \cdots & 0 & \frac{1}{2} & -2 & \frac{3}{2} \end{bmatrix}. \quad (2.33)$$

Este operador implementa una diferencia adelantada en el primer punto, una diferencia centrada en el dominio interior y una diferencia atrasada en el último punto, según el siguiente esquema:

- **Primer instante temporal** ( $t_0$ ): diferencia adelantada de segundo orden

$$\dot{u}_{t_0} \approx \frac{-\frac{3}{2}u_{t_0} + 2u_{t_1} - \frac{1}{2}u_{t_2}}{\Delta t}.$$

- **Puntos interiores** ( $1 \leq i \leq T - 2$ ): diferencia centrada

$$\dot{u}_{t_i} \approx \frac{u_{t_{i+1}} - u_{t_{i-1}}}{2\Delta t}.$$

- **Último instante temporal** ( $t_{T-1}$ ): diferencia atrasada de segundo orden

$$\dot{u}_{t_{T-1}} \approx \frac{\frac{3}{2}u_{t_{T-1}} - 2u_{t_{T-2}} + \frac{1}{2}u_{t_{T-3}}}{\Delta t}.$$

En términos prácticos, el operador  $\Phi_t$  actúa como un filtro convolucional fijo de ancho tres, aplicado secuencialmente a lo largo del eje temporal. De este modo, la operación de diferenciación se integra de forma natural en el grafo computacional del modelo, permitiendo la propagación del gradiente durante el entrenamiento sin necesidad de definir derivadas analíticas adicionales.

Cuando se requiere la segunda derivada temporal, se aplica nuevamente el mismo operador sobre la derivada de primer orden, es decir:

$$\ddot{\mathbf{u}} \approx \Phi_t^2 \mathbf{u}. \quad (2.34)$$

Aunque esta aproximación no equivale exactamente a un operador centrado de segunda derivada y presenta menor exactitud en los extremos, permite integrar la diferenciación temporal dentro del grafo computacional de manera eficiente. Sin embargo, la reducción de orden en los bordes y la acumulación de errores discretos pueden afectar la estimación de las derivadas de orden superior, condicionando el cumplimiento exacto de las restricciones físicas en la función de pérdida. Por tanto, el método debe entenderse como un compromiso entre eficiencia numérica y consistencia física.

### PhyCNN-MO: Red Convolucional Guiada por la Física de Múltiples Salidas

La primera configuración propuesta del metamodelo general, denominada **PhyCNN-MO** (*Multiple-Output*), busca estimar simultáneamente las tres variables fundamentales del sistema dinámico estructural: el desplazamiento relativo  $u(t)$ , su velocidad  $\dot{u}(t)$  y la fuerza interna normalizada  $g(t)$ . Esta formulación persigue capturar de manera explícita la relación física entre estas magnitudes, reforzando la coherencia entre las respuestas cinemáticas y dinámicas del modelo.

Desde el punto de vista funcional, la red convolucional se define como una aproximadora no-lineal que mapea la aceleración de entrada del suelo  $a_g(t)$  hacia las tres salidas mencionadas:

$$\widehat{Y} = \text{CNN}_\theta(a_g), \quad \widehat{Y} \equiv [\widehat{\mathbf{u}}, \widehat{\mathbf{v}}, \widehat{\mathbf{g}}],$$

donde  $\theta$  representa el conjunto de parámetros entrenables de la red. El entrenamiento se realiza combinando restricciones basadas en datos (desplazamientos, velocidades y fuerzas internas medidas o simuladas) con restricciones físicas derivadas directamente de la ecuación de movimiento normalizada (Ecuación 2.31).

La entrada al modelo consiste en la aceleración externa del suelo  $\mathbf{a}_g \in \mathbb{R}^{N \times T \times 1}$ , mientras que la salida corresponde a las tres variables del sistema estructural, de dimensión  $\mathbb{R}^{N \times T \times 3}$ , donde:

- $N$ : número de muestras (simulaciones o registros de acelerogramas);
- $T$ : número de pasos temporales por muestra;
- las tres características de salida representan, respectivamente,  $u(t)$ ,  $\dot{u}(t)$  y  $g(t)$ .

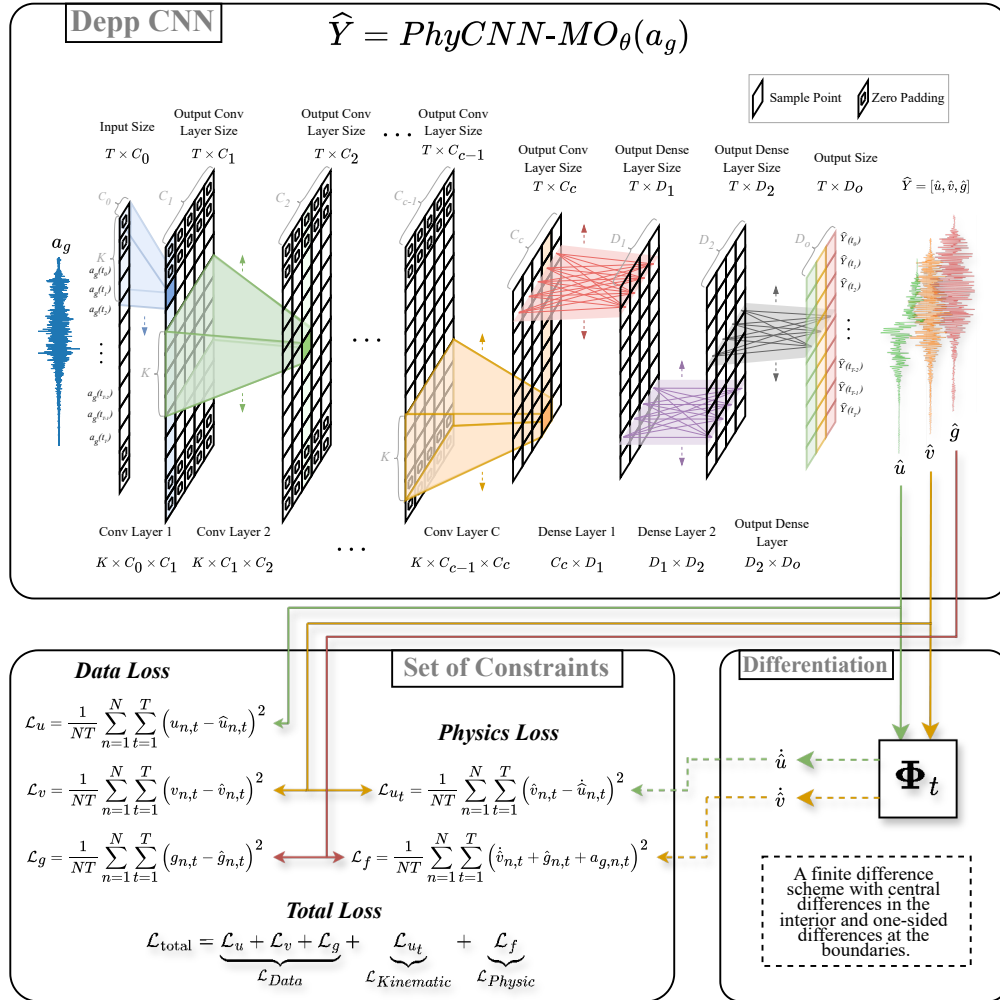


Figura 2.16: Arquitectura y marco de entrenamiento del modelo PhyCNN-MO.

La arquitectura del modelo, ilustrada en la Figura 2.16, está compuesta por cinco capas convolucionales unidimensionales, cada una con 64 filtros y tamaño de kernel  $K = 40$ , todas implementadas con *padding same* para preservar la longitud temporal de la señal. Cada capa aplica una activación ReLU y opera con *stride* unitario, de modo que las dependencias temporales se mantienen completas a lo largo de la secuencia. Estas capas extraen representaciones jerárquicas que capturan patrones dinámicos complejos del movimiento estructural.

La salida convolucional se proyecta luego sobre el espacio de características mediante dos capas densas con 50 unidades cada una, también con activación ReLU, aplicadas de manera independiente en cada instante

temporal (operación tipo *TimeDistributed*). Finalmente, una capa de salida totalmente conectada de tamaño tres genera las predicciones para las tres variables físicas.

Formalmente, para una entrada  $X = \alpha_g$ , la salida del modelo se expresa como:

$$\widehat{Y} = \text{CNN}_\theta(X) \in \mathbb{R}^{N \times T \times 3},$$

dividida según:

$$\widehat{u}(t) = \widehat{Y}[:, :, 0], \quad \widehat{v}(t) = \widehat{Y}[:, :, 1], \quad \widehat{g}(t) = \widehat{Y}[:, :, 2].$$

Para evaluar las restricciones físicas que involucran derivadas temporales, las magnitudes  $\hat{\mathbf{u}}$  y  $\hat{\mathbf{v}}$  se obtienen directamente dentro del grafo computacional mediante un esquema de diferencias finitas mixto de segundo orden, descrito en detalle previamente en la Sección 2.9.1. De este modo, el modelo preserva en principio la diferenciabilidad completa del grafo computacional, permitiendo la propagación del gradiente durante el entrenamiento sin introducir operadores externos o no derivables.

El proceso de entrenamiento de la PhyCNN-MO busca simultáneamente minimizar el error con respecto a los datos observados y asegurar la coherencia física del modelo. La función de pérdida total se compone de cinco términos, agrupados en dos categorías principales: (i) pérdidas de supervisión sobre los datos medidos o simulados, y (ii) pérdidas físicas derivadas de las leyes de la dinámica estructural.

Las pérdidas de supervisión se definen como:

1. **Ajuste en desplazamiento:**

$$\mathcal{L}_u = \frac{1}{NT} \sum_{n=1}^N \sum_{t=1}^T (u_{n,t} - \widehat{u}_{n,t})^2$$

2. **Ajuste en velocidad:**

$$\mathcal{L}_v = \frac{1}{NT} \sum_{n=1}^N \sum_{t=1}^T (v_{n,t} - \widehat{v}_{n,t})^2$$

3. **Ajuste en fuerza interna normalizada:**

$$\mathcal{L}_g = \frac{1}{NT} \sum_{n=1}^N \sum_{t=1}^T (g_{n,t} - \widehat{g}_{n,t})^2$$

Por su parte, las pérdidas físicas se formulan como:

1. **Restricción cinemática:** asegura la consistencia entre la velocidad aprendida y la derivada del desplazamiento predicho:

$$\mathcal{L}_{u_t} = \frac{1}{NT} \sum_{n=1}^N \sum_{t=1}^T (\widehat{v}_{n,t} - \dot{\widehat{u}}_{n,t})^2$$

2. **Restricción dinámica:** impone el cumplimiento de la ecuación del movimiento normalizada:

$$\mathcal{L}_f = \frac{1}{NT} \sum_{n=1}^N \sum_{t=1}^T (\dot{\hat{v}}_{n,t} + \hat{g}_{n,t} + a_{g,n,t})^2$$

De esta forma, la pérdida total queda expresada como:

$$\mathcal{L}_{\text{total}} = \underbrace{(\mathcal{L}_u + \mathcal{L}_v + \mathcal{L}_g)}_{\mathcal{L}_{\text{data}}} + \underbrace{\mathcal{L}_{u_t}}_{\mathcal{L}_{\text{kine}}} + \underbrace{\mathcal{L}_f}_{\mathcal{L}_{\text{phys}}} .$$

El modelo PhyCNN-MO combina aprendizaje supervisado multisalida con penalizaciones basadas en restricciones físicas, incorporando explícitamente la cinemática y la dinámica estructural en el proceso de aprendizaje. Esta estrategia híbrida busca que la red capture relaciones físicas intrínsecas entre desplazamiento, velocidad y fuerza interna, lo que en principio podría favorecer la generalización frente a condiciones de excitación no observadas durante el entrenamiento.

En síntesis, la arquitectura PhyCNN-MO se apoya en la capacidad de las redes convolucionales unidimensionales para representar dependencias temporales complejas, incorporando restricciones físicas que buscan mantener la coherencia dinámica del sistema. Este enfoque ofrece una formulación conceptualmente atractiva por su integración entre aprendizaje de datos y principios físicos, aunque su desempeño práctico depende de la calibración de los términos de pérdida y del control de los errores numéricos asociados al cálculo de derivadas dentro del grafo computacional.

### PhyCNN-SO: Red Convolutiva Guiada por la Física de Salida Única

La variante **PhyCNN-SO** (*Single-Output*) corresponde a una versión simplificada del modelo PhyCNN-MO, diseñada para predecir únicamente el desplazamiento estructural relativo  $\hat{\mathbf{u}}$  de un sistema sometido a excitación sísmica. A diferencia de la arquitectura **PhyCNN-MO**, que estima simultáneamente desplazamientos, velocidades y fuerzas internas, esta versión concentra la capacidad de aprendizaje en la variable de desplazamiento, deduciendo sus derivadas temporales mediante un operador de diferenciación discreto incorporado en el grafo computacional. El enfoque retoma la filosofía de las *Physics-Informed Neural Networks* (PINN) en el sentido de que incorpora información física en la función de pérdida; sin embargo, dicha integración se realiza de forma indirecta, a través de la comparación entre aceleraciones derivadas del modelo y las observadas en los datos, más que mediante la imposición explícita de la ecuación del movimiento.

La entrada al modelo corresponde a la aceleración externa normalizada por la masa, representada como una secuencia temporal:

$$\mathbf{a}_g \in \mathbb{R}^{N \times T \times 1},$$

donde  $N$  es el número de muestras,  $T$  el número de pasos temporales y la última dimensión representa el canal de entrada único. La red produce como salida una única serie temporal de desplazamientos estructurales:

$$\hat{Y} = \text{CNN}_\theta(\mathbf{a}_g), \quad \hat{Y} \equiv \hat{\mathbf{u}},$$

a partir de la cual se obtienen las derivadas temporales mediante el operador discreto de diferencias finitas  $\Phi_t$  (definido en la Sección 2.9.1):

$$\hat{\mathbf{v}} = \Phi_t \hat{\mathbf{u}}, \quad \hat{\mathbf{a}} = \Phi_t^2 \hat{\mathbf{u}}.$$

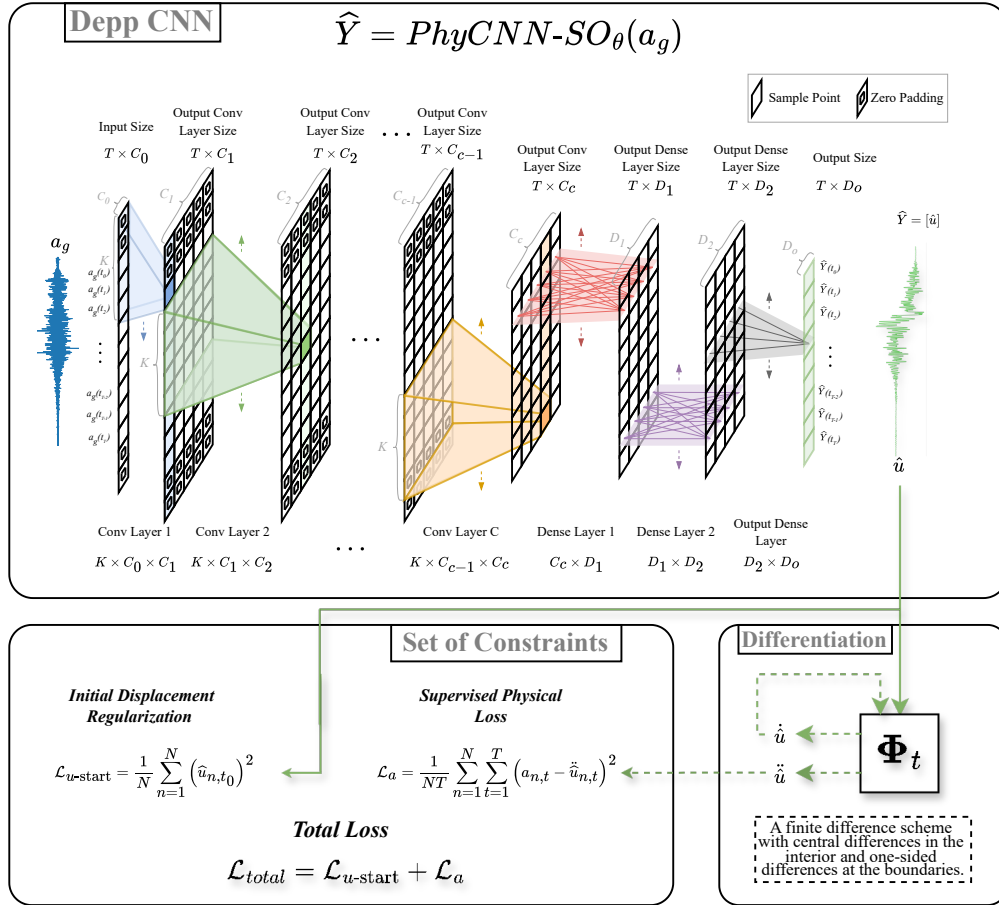


Figura 2.17: Arquitectura de la red PhyCNN-SO.

La arquitectura de la red PhyCNN-SO, ilustrada en la Figura 2.17, conserva una estructura secuencial similar a la de la PhyCNN-MO, compuesta por cinco capas convolucionales unidimensionales. Cada capa utiliza 64 filtros con un tamaño de kernel  $K = 50$  (10 unidades mayor que en PhyCNN-MO), aplicando *padding same* y activación ReLU. Estas capas tienen como propósito extraer características temporales relevantes de la señal de aceleración, lo que potencialmente permite representar dependencias de mediano y largo plazo asociadas al comportamiento dinámico del sistema. Posteriormente, dos capas densas de 50 unidades, también con activación ReLU y aplicadas de manera independiente por paso temporal (*TimeDistributed*), transforman las características extraídas en una representación de nivel superior. Finalmente, una capa densa de salida con una única unidad por instante temporal produce la estimación del desplazamiento estructural:

$$\hat{Y} = \text{CNN}_{\theta}(a_g) \in \mathbb{R}^{N \times T \times 1}.$$

El proceso de entrenamiento combina una penalización física basada en la aceleración estructural estimada y un término de regularización local sobre los desplazamientos iniciales. La función de pérdida total se expresa como:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_a + \mathcal{L}_{u_{\text{start}}}.$$

Los términos que la componen son:

1. **Pérdida física en aceleración:** penaliza la discrepancia entre la aceleración estructural obtenida mediante diferenciación numérica de la salida del modelo sea consistente con la observada en los datos:

$$\mathcal{L}_a = \frac{1}{NT} \sum_{n=1}^N \sum_{t=1}^T (a_{n,t} - \ddot{\hat{u}}_{n,t})^2.$$

2. **Regularización de desplazamiento inicial:** evita inestabilidades numéricas o desplazamientos no físicos en el primer instante de la señal, penalizando valores elevados de  $u(t)$  en el primer paso:

$$\mathcal{L}_{u_{\text{start}}} = \frac{1}{N} \sum_{n=1}^N (\hat{u}_{n,t_0})^2.$$

Aunque el modelo no predice directamente la fuerza interna  $g(t)$ , ésta puede reconstruirse a partir del equilibrio dinámico normalizado de la ecuación de movimiento (Ecuación 2.31). De este modo, la fuerza interna estimada se define como:

$$\hat{g}(t) = -\ddot{\hat{u}}(t) - a_g(t),$$

lo que permite comparar las respuestas generadas por la PhyCNN-SO con las obtenidas mediante el modelo PhyCNN-MO, y evaluar la consistencia del comportamiento dinámico aprendido.

La arquitectura PhyCNN-SO constituye una alternativa de menor complejidad computacional, al reducir el número de variables predichas y simplificar la formulación de la función de pérdida. Esta configuración podría resultar más adecuada en escenarios donde el conjunto de datos disponible es limitado o presenta un alto nivel de ruido, ya que focaliza la capacidad de aprendizaje del modelo en la predicción del desplazamiento estructural. No obstante, la ausencia de salidas explícitas de velocidad y fuerza interna puede limitar su interpretabilidad física y reducir su capacidad para representar interacciones no-lineales complejas. Aun así, la incorporación de restricciones físicas mediante diferenciación numérica busca mantener la coherencia del modelo con las leyes del movimiento, conservando en principio el carácter *physics-informed* que distingue a la familia PhyCNN.

### 2.9.2. PhyLSTM: Redes LSTM multi-bloque informadas por la física

Las arquitecturas **PhyLSTM**<sup>2</sup> y **PhyLSTM**<sup>3</sup>, propuestas por Zhang et al. (2020b), corresponden a un enfoque basado en redes recurrentes *Long Short-Term Memory* (LSTM) orientado al metamodelado de sistemas estructurales no-lineales sometidos a excitaciones sísmicas. Los autores plantean que, cuando las relaciones no-lineales que gobiernan la respuesta estructural son complejas, implícitas o de alto orden, resulta

inviabile formular un modelo reducido con una expresión matemática cerrada que represente con precisión el comportamiento físico del sistema. En tales casos, el desarrollo de modelos informados por la física constituye una alternativa atractiva para capturar de manera aproximada las relaciones dinámicas entre la excitación sísmica y la respuesta estructural.

Dentro de este marco, se asume que el comportamiento restaurador del sistema depende de variables de estado que incluyen el desplazamiento, la velocidad y un parámetro histerético latente que refleja la memoria del material y los efectos de acumulación de daño. Dado que dicho parámetro no puede medirse directamente ni obtenerse mediante simulaciones convencionales, el entrenamiento de un modelo puramente basado en datos resulta insuficiente para caracterizar la dinámica completa del sistema. Como solución, los autores proponen un esquema de aprendizaje profundo informado por la física basado en múltiples redes LSTM acopladas, que busca inferir el estado dinámico completo a partir de información parcial. El enfoque combina datos observados de desplazamiento y velocidad con restricciones físicas derivadas de la ecuación del movimiento, incorporadas en la función de pérdida como penalizaciones, de manera que el modelo aprenda de forma coherente con las leyes físicas subyacentes.

De manera análoga a las formulaciones PhyCNN, las arquitecturas PhyLSTM utilizan el residuo dinámico definido en la Ecuación 2.31 como restricción física durante el entrenamiento. Las derivadas temporales necesarias para evaluar dicho residuo se calculan mediante el mismo mecanismo de diferenciación numérica descrito en la Sección 2.9.1. En esta investigación, estos modelos se consideran como referencias conceptuales y metodológicas para analizar las ventajas y limitaciones de las redes informadas por la física en el contexto del metamodelado estructural no-lineal.

### Bloques LSTM y esquema de entrenamiento

Las arquitecturas PhyLSTM se construyen a partir de *bloques LSTM* secuenciales, cada uno compuesto por una serie de capas recurrentes y densas destinadas a procesar información temporal unidimensional. La estructura general de los bloques se ilustra en la Figura 2.18. Cada bloque incluye tres capas LSTM consecutivas con 100 unidades ocultas, configuradas para retornar secuencias completas, lo que permite conservar la información temporal a lo largo de toda la red. Estas capas emplean la implementación optimizada *CuDNNLSTM*, que mejora el rendimiento computacional en GPU y facilita el entrenamiento de secuencias largas. El estado interno de las celdas se reinicia al comienzo de cada nueva secuencia (*stateful = False*), por lo que los registros se procesan de forma independiente sin transferir información entre ellos. Esta configuración favorece la estabilidad numérica y previene la acumulación de errores entre lotes, aunque restringe la posibilidad de representar procesos dependientes de condiciones iniciales o estados previos dentro de una misma ejecución. La salida de cada capa LSTM se somete posteriormente a una activación ReLU, que introduce no-linealidad en el modelo y reduce el riesgo de saturación del gradiente durante el entrenamiento.

Una diferencia relevante entre las arquitecturas PhyLSTM<sup>2</sup> y PhyLSTM<sup>3</sup> radica en la incorporación de una capa densa intermedia. En los bloques que conforman la PhyLSTM<sup>2</sup>, se incluye una capa totalmente conectada de 100 neuronas entre la última capa LSTM y la capa de salida, mientras que en la PhyLSTM<sup>3</sup> esta capa intermedia se omite. En ambos casos, la capa densa opera de manera distribuida en el tiempo, equivalente a una implementación *TimeDistributed*, es decir, la misma transformación lineal y no-lineal se

aplica independientemente a cada instante temporal de la secuencia. Esta configuración permite mantener la coherencia temporal de las salidas sin necesidad de aplanar la secuencia completa antes de la proyección final. La capa de salida, también densa, varía en tamaño según el número de magnitudes físicas que el bloque debe predecir (por ejemplo, tres salidas para desplazamiento, velocidad y el parámetro de histéresis, o una única salida para la fuerza interna normalizada). En todos los casos, las salidas conservan la longitud temporal de las entradas, generando una predicción por instante de tiempo.

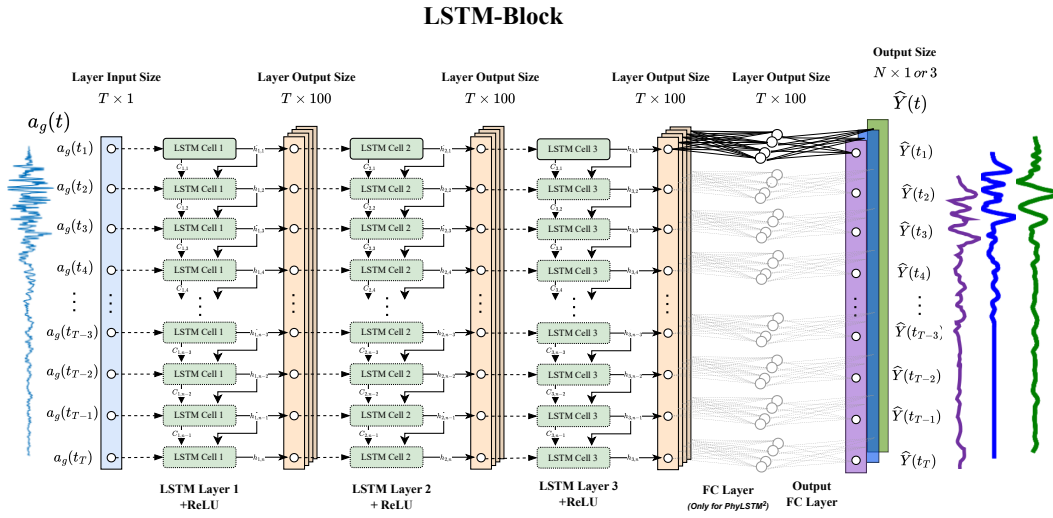


Figura 2.18: Bloque LSTM genérico utilizado en las arquitecturas PhyLSTM. Cada bloque incluye tres capas LSTM (100 unidades) con activación ReLU y una o dos capas densas distribuidas en el tiempo. En los bloques de la PhyLSTM<sup>2</sup> se incorpora una capa densa intermedia de 100 unidades, ausente en la PhyLSTM<sup>3</sup>.

Desde una perspectiva funcional, esta configuración busca equilibrar la capacidad de representación temporal y la eficiencia computacional. El uso de múltiples capas LSTM que devuelven secuencias completas favorece la captura de dependencias de corto y largo plazo, mientras que las capas densas distribuidas en el tiempo permiten una transformación no-lineal local en cada instante, preservando la estructura temporal de la señal. Sin embargo, el procesamiento independiente de cada secuencia limita la posibilidad de acumular información contextual entre registros consecutivos, lo que podría restringir la representación de efectos de memoria de muy largo alcance. En términos prácticos, el diseño de los bloques PhyLSTM refleja un compromiso entre expresividad, estabilidad del entrenamiento y costo computacional.

En cuanto al esquema de entrenamiento, las PhyLSTM difieren de las PINNs tradicionales en la forma en que aplican las restricciones físicas. En lugar de puntos de colocación dentro de un dominio continuo, se emplea un *conjunto de colocación* formado por *registros sísmicos completos*. Se definen dos conjuntos: (i) un *conjunto de entrenamiento supervisado* de tamaño  $N$ , que contiene las excitaciones sísmicas y las respuestas observadas, y (ii) un *conjunto de colocación* de tamaño  $N_c > N$ , compuesto por registros adicionales en los cuales sólo se dispone de la entrada sísmica  $a_g$ . Como convención notacional, las entradas provenientes de este segundo conjunto se denotan  $a_{g,c}$ , y las predicciones asociadas se indican con subíndice  $c$ .

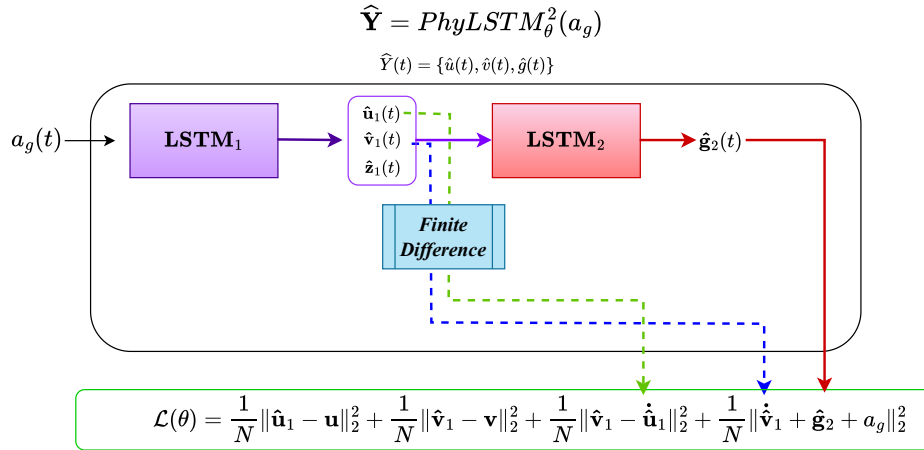


Figura 2.19: Arquitectura PhyLSTM<sup>2</sup>, donde cada bloque LSTM representa el bloque presentado en la sección 2.9.2

Durante el entrenamiento, el primer bloque LSTM<sup>(1)</sup> procesa ambos conjuntos: las predicciones generadas a partir del conjunto supervisado se emplean en las pérdidas basadas en datos, mientras que las obtenidas con el conjunto de colocación se utilizan para evaluar las pérdidas físicas y, en las arquitecturas multibloque, pueden alimentar los bloques subsiguientes. En coherencia con ello, los términos de pérdida asociados a datos se promedian sobre  $N$ , mientras que los términos físicos se calculan sobre  $N_c$ . Las derivadas temporales requeridas para las restricciones físicas se obtienen mediante el mecanismo de diferenciación numérica descrito en la Sección 2.9.1.

### PhyLSTM<sup>2</sup>: Red LSTM informada por la física de dos bloques

La arquitectura **PhyLSTM<sup>2</sup>** mostrada en la Figur 2.19 implementa un esquema compuesto por dos bloques LSTM acoplados, junto con el mecanismo de diferenciación temporal descrito en la Sección 2.9.1.

La entrada del modelo corresponde a la aceleración del suelo normalizada por la masa,  $\mathbf{a}_g \in \mathbb{R}^{N \times T \times 1}$ , donde  $N$  es el número de registros del conjunto de entrenamiento y  $T$  la longitud temporal. El primer bloque recibe  $\mathbf{a}_g$  y genera tres series correspondientes al desplazamiento, la velocidad y la fuerza interna normalizada:

$$\hat{\mathbf{Y}}^{(1)} \equiv [\hat{\mathbf{u}}^{(1)}, \hat{\mathbf{v}}^{(1)}, \hat{\mathbf{g}}^{(1)}] = \text{LSTM}^{(1)}(\mathbf{a}_g; \boldsymbol{\theta}^{(1)}).$$

El segundo bloque utiliza las predicciones del primero para estimar una fuerza restauradora refinada, empleada posteriormente en la restricción dinámica:

$$\hat{\mathbf{Y}}^{(2)} \equiv \hat{\mathbf{g}}^{(2)} = \text{LSTM}^{(2)}(\hat{\mathbf{Y}}^{(1)}; \boldsymbol{\theta}^{(2)}).$$

Durante el entrenamiento, los bloques procesan dos conjuntos de datos con funciones complementarias. El primero corresponde al *conjunto de entrenamiento supervisado* ( $N$  registros), que contiene tanto las

excitaciones sísmicas como las respuestas estructurales observadas, y se utiliza para ajustar las pérdidas basadas en datos. El segundo es el *conjunto de colocación*, compuesto por  $N_c > N$  registros que incluyen únicamente las aceleraciones de entrada  $\mathbf{a}_{g,c}$ , sin información de salida. Este conjunto se emplea exclusivamente para evaluar las pérdidas físicas, permitiendo imponer las restricciones dinámicas sobre un dominio de excitaciones más amplio.

La red LSTM<sup>(1)</sup> procesa ambos conjuntos de forma independiente:

$$\widehat{\mathbf{Y}}^{(1)} = \text{LSTM}^{(1)}(\mathbf{a}_g; \boldsymbol{\theta}^{(1)}), \quad \widehat{\mathbf{Y}}_c^{(1)} = \text{LSTM}^{(1)}(\mathbf{a}_{g,c}; \boldsymbol{\theta}^{(1)}),$$

donde las variables con subíndice  $c$  corresponden a las predicciones generadas para el conjunto de colocación. Estas predicciones alimentan el segundo bloque, encargado de estimar la fuerza interna normalizada asociada:

$$\widehat{\mathbf{Y}}_c^{(2)} \equiv \widehat{\mathbf{g}}_c^{(2)} = \text{LSTM}^{(2)}(\widehat{\mathbf{Y}}_c^{(1)}; \boldsymbol{\theta}^{(2)}).$$

El entrenamiento busca minimizar de manera conjunta las discrepancias con los datos observados y el incumplimiento de las relaciones físicas impuestas. Para ello, las derivadas temporales de las predicciones, en particular  $\dot{\widehat{\mathbf{u}}}^{(1)}$  y  $\dot{\widehat{\mathbf{v}}}^{(1)}$ , se obtienen mediante el esquema de diferencias finitas de segundo orden integrado en el grafo computacional. Las pérdidas asociadas a los datos se calculan sobre el conjunto supervisado ( $N$ ), mientras que las pérdidas físicas se promedian sobre el conjunto de colocación ( $N_c$ ).

Las pérdidas de supervisión incluyen:

### 1. Ajuste en desplazamiento:

$$\mathcal{L}_u = \frac{1}{NT} \sum_{n=1}^N \sum_{t=1}^T (u_{n,t} - \widehat{u}_{n,t}^{(1)})^2,$$

### 2. Ajuste en velocidad:

$$\mathcal{L}_v = \frac{1}{NT} \sum_{n=1}^N \sum_{t=1}^T (v_{n,t} - \widehat{v}_{n,t}^{(1)})^2.$$

Según la formulación propuesta por los autores, no se incorpora supervisión sobre la fuerza interna normalizada, argumentando que su medición experimental directa resulta inviable en estructuras reales. Por su parte, las pérdidas físicas se definen como:

### 1. Restricción cinemática: busca la coherencia entre la velocidad aprendida y la derivada del desplazamiento predicho,

$$\mathcal{L}_{u_t} = \frac{1}{N_c T} \sum_{n=1}^{N_c} \sum_{t=1}^T (\widehat{v}_{c,n,t}^{(1)} - \dot{\widehat{u}}_{c,n,t}^{(1)})^2,$$

### 2. Restricción dinámica: impone el cumplimiento de la ecuación del movimiento normalizada,

$$\mathcal{L}_f = \frac{1}{N_c T} \sum_{n=1}^{N_c} \sum_{t=1}^T (\dot{\widehat{v}}_{c,n,t}^{(1)} + \widehat{g}_{c,n,t}^{(2)} + a_{g,c,n,t})^2.$$

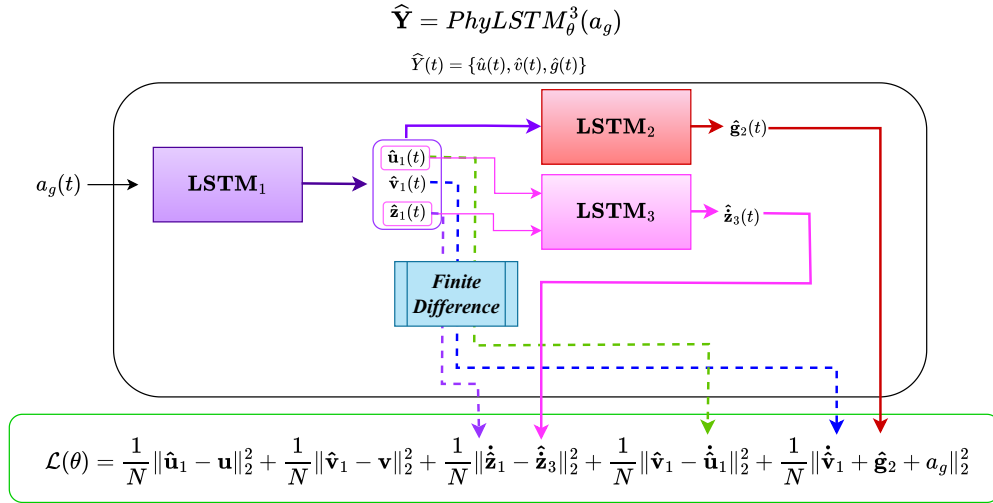


Figura 2.20: Arquitectura  $\text{PhyLSTM}^3$ , donde cada bloque LSTM representa el bloque presentado en la sección 2.9.2

La función de pérdida total queda expresada como:

$$\mathcal{L}_{\text{total}}^{(\text{PhyLSTM}^2)} = \mathcal{L}_u + \mathcal{L}_v + \mathcal{L}_{u_t} + \mathcal{L}_f.$$

Desde una perspectiva analítica, el modelo  $\text{PhyLSTM}^2$  combina aprendizaje supervisado con penalizaciones físicas aplicadas sobre un conjunto ampliado de registros, buscando equilibrar la fidelidad a los datos y la coherencia con las leyes dinámicas. Este planteamiento amplía el rango de excitaciones consideradas durante el entrenamiento, lo que, en principio, podría favorecer la capacidad de generalización en escenarios con información limitada. No obstante, la dependencia del cálculo numérico de las derivadas y la complejidad asociada al acoplamiento de dos redes recurrentes incrementan la sensibilidad del proceso de optimización, pudiendo afectar la estabilidad y reproducibilidad del entrenamiento. En síntesis, la  $\text{PhyLSTM}^2$  constituye un ejemplo representativo del compromiso entre capacidad de representación y coherencia física que caracteriza a los enfoques informados por la física, cuya efectividad depende en gran medida de la calidad del conjunto de colocación y del balance entre los términos de pérdida basados en datos y en restricciones físicas.

### **PhyLSTM<sup>3</sup>: Red LSTM informada por la física de tres bloques**

La arquitectura  $\text{PhyLSTM}^3$  mostrada en la Figura 2.20 extiende la formulación propuesta para la  $\text{PhyLSTM}^2$ , incorporando un tercer bloque destinado a modelar la evolución de una variable latente de histéresis  $\mathbf{z}$  que representa los efectos de memoria del sistema estructural. Esta variable cumple un papel análogo al parámetro interno en los modelos histeréticos de Bouc–Wen, aunque aquí se aprende de manera implícita a partir de los datos durante el entrenamiento de la red.

En esta configuración, el nuevo bloque busca aproximar la derivada temporal de la variable latente ( $\dot{\mathbf{z}}$ ) a partir de las predicciones del primer bloque, mediante la relación:

$$\widehat{\mathbf{Y}}^{(3)} \equiv \widehat{\dot{\mathbf{z}}}^{(3)} = \text{LSTM}^{(3)}(\widehat{\mathbf{v}}^{(1)}, \widehat{\mathbf{z}}^{(1)}; \boldsymbol{\theta}^{(3)}).$$

De este modo, la arquitectura introduce un mecanismo explícito para capturar la dinámica temporal de la variable histerética, integrando dependencias secuenciales de largo plazo que podrían ser relevantes en la representación del comportamiento no-lineal del sistema.

El procedimiento de entrenamiento mantiene la misma estructura general descrita para la PhyLSTM<sup>2</sup>, diferenciando entre el conjunto de entrenamiento supervisado ( $N$  registros) y el conjunto de colocación ( $N_c > N$ ). Las tres redes LSTM<sup>(1)</sup>, LSTM<sup>(2)</sup> y LSTM<sup>(3)</sup> procesan ambos conjuntos de manera análoga, generando para el conjunto de colocación las predicciones:

$$\widehat{\mathbf{Y}}_c^{(1)} = \text{LSTM}^{(1)}(\mathbf{a}_{g,c}; \boldsymbol{\theta}^{(1)}), \quad \widehat{\mathbf{Y}}_c^{(2)} = \text{LSTM}^{(2)}(\widehat{\mathbf{Y}}_c^{(1)}; \boldsymbol{\theta}^{(2)}), \quad \widehat{\mathbf{Y}}_c^{(3)} = \text{LSTM}^{(3)}(\widehat{\mathbf{v}}_c^{(1)}, \widehat{\mathbf{z}}_c^{(1)}; \boldsymbol{\theta}^{(3)}).$$

Las salidas obtenidas para este conjunto se utilizan exclusivamente en las pérdidas físicas, promediadas sobre  $N_c$ , mientras que las pérdidas basadas en datos se calculan sobre el conjunto supervisado ( $N$ ).

Al igual que en el caso anterior, las derivadas temporales se obtienen mediante el esquema de diferenciación numérica descrito en la Sección 2.9.1. Con el fin de integrar el nuevo bloque en el proceso de entrenamiento, se introduce una pérdida adicional que busca la consistencia entre la derivada temporal de la variable latente predicha por el tercer bloque y la derivada numérica obtenida a partir de la predicción del primer bloque:

$$\mathcal{L}_z = \frac{1}{N_c T} \sum_{n=1}^{N_c} \sum_{t=1}^T (\widehat{\dot{\mathbf{z}}}_{c,n,t}^{(3)} - \dot{\mathbf{z}}_{c,n,t}^{(1)})^2.$$

Esta penalización complementa las pérdidas  $\mathcal{L}_u$ ,  $\mathcal{L}_v$ ,  $\mathcal{L}_{u_t}$  y  $\mathcal{L}_f$  definidas previamente, extendiendo el marco de entrenamiento hacia la estimación coherente de una dinámica latente.

La función de pérdida total se expresa como:

$$\mathcal{L}_{\text{total}}^{(\text{PhyLSTM}^3)} = \mathcal{L}_u + \mathcal{L}_v + \mathcal{L}_{u_t} + \mathcal{L}_f + \mathcal{L}_z.$$

La introducción de un tercer bloque busca capturar de manera explícita la evolución temporal de una variable histerética latente, ampliando la capacidad del modelo para representar efectos de memoria y no-linealidades acumulativas. Desde un punto de vista conceptual, esta extensión refuerza la naturaleza física del enfoque, aunque también incrementa su complejidad arquitectónica y el número de parámetros entrenables. El uso del conjunto de colocación permite imponer las restricciones físicas sobre un dominio de entrada más amplio, pero la propagación de errores en las derivadas y la mayor dependencia del ajuste numérico pueden afectar la estabilidad y la reproducibilidad del entrenamiento. En conjunto, la PhyLSTM<sup>3</sup> representa un intento por enriquecer la modelación de la dinámica interna de sistemas no-lineales mediante aprendizaje informado por la física, aunque su desempeño efectivo depende del equilibrio entre precisión numérica, complejidad de la red y disponibilidad de información fiable en los conjuntos de entrenamiento y colocación.

### 2.9.3. Esquema de optimización y criterios de entrenamiento

Tanto las arquitecturas **PhyCNN** como las **PhyLSTM** emplean un esquema de optimización híbrido basado en la combinación de los algoritmos *Adam* y *L-BFGS-B*. El optimizador *Adam* se utiliza durante las primeras etapas del entrenamiento, aplicando un descenso estocástico del gradiente con tasa de aprendizaje constante de  $1 \times 10^{-3}$ , lo que permite una exploración eficiente del espacio de parámetros y una rápida convergencia inicial. Posteriormente, el optimizador *L-BFGS-B*, un método cuasi-Newton de segunda orden, refina los parámetros ajustando la solución hacia un mínimo local más estable mediante la aproximación iterativa del Hessiano. Esta combinación aprovecha la rapidez adaptativa de *Adam* y la precisión del método de optimización de tipo determinista, reduciendo la posibilidad de oscilaciones en torno al mínimo.

En los códigos originales de las arquitecturas propuestas por Zhang et al. (2020a) y Zhang et al. (2020b), el conjunto de validación se define pero no cumple una función activa dentro del proceso de entrenamiento. En consecuencia, no se aplica ningún criterio de *early stopping*, y los pesos finales del modelo corresponden a los obtenidos en la última iteración, guiándose exclusivamente por la pérdida de entrenamiento. Además, el número de épocas se ajusta de manera distinta para cada conjunto de datos.

En este estudio se mantuvo la filosofía general de los trabajos originales, priorizando la reproducibilidad de los resultados, pero se incorporaron algunas modificaciones orientadas a mejorar la estabilidad y la comparabilidad de los modelos. En primer lugar, aunque el conjunto de validación continúa sin intervenir en las decisiones de parada temprana, se utiliza para monitorear el error de validación durante el entrenamiento, con el objetivo de analizar la evolución del sobreajuste sin alterar la optimización del modelo. En segundo lugar, se reemplazó el criterio de conservar los pesos de la última iteración por una estrategia más robusta: almacenar los parámetros correspondientes a la *menor pérdida de entrenamiento* alcanzada. Esta decisión busca evitar que incrementos abruptos de la pérdida en las últimas iteraciones comprometan la calidad del modelo final, mejorando la estabilidad del proceso sin modificar el protocolo de entrenamiento original.

Finalmente, con el propósito de asegurar una base común para la comparación entre arquitecturas y conjuntos de datos, se definió un número estándar de 15 000 épocas para todos los modelos y bases sísmicas utilizadas en este estudio. Este valor garantiza una convergencia adecuada de la pérdida de entrenamiento bajo los criterios establecidos, aun cuando un número elevado de iteraciones no necesariamente implique una mejor generalización. De este modo, las adaptaciones introducidas conservan la coherencia metodológica del enfoque original, pero fortalecen su consistencia experimental y reproducibilidad en el contexto del presente análisis comparativo.

### 2.9.4. Síntesis general y proyección del análisis comparativo

Las arquitecturas **PhyCNN** y **PhyLSTM** representan dos enfoques complementarios dentro del paradigma de aprendizaje profundo informado por la física aplicado a la dinámica estructural. Ambos modelos integran la ecuación del movimiento como restricción durante el entrenamiento, combinando información observacional o simulada y principios físicos para estimar el comportamiento no-lineal de estructuras sometidas a excitaciones sísmicas. Su formulación se apoya en supuestos comunes, como la validez de la ecuación de equilibrio en todo el dominio temporal, la estabilidad del esquema numérico de diferenciación y la representatividad de

los datos empleados. No obstante, difieren en la forma de capturar las dependencias temporales: mientras las PhyCNN utilizan operadores convolucionales de alcance fijo, las PhyLSTM incorporan mecanismos de memoria recurrente que permiten modelar efectos de histéresis y dependencia de historia.

En términos generales, ambas familias ilustran las posibilidades y limitaciones inherentes de los modelos informados por la física: permiten reducir la dependencia de datos exhaustivos, pero su precisión depende del tratamiento numérico de las derivadas, la calibración de los términos de pérdida y la calidad del conjunto de colocación. En el **Capítulo 3** se presenta un análisis comparativo de estas arquitecturas, evaluando su desempeño y capacidad de generalización frente a distintas bases de datos sísmicas y escenarios estructurales.

## 2.10. Síntesis del marco teórico y conexión con capítulos posteriores

El marco desarrollado en este capítulo estableció los fundamentos necesarios para abordar la predicción historia–temporal de la respuesta estructural bajo demandas sísmicas de alta no-linealidad. A partir de la formulación matricial de las ecuaciones de movimiento y su representación en espacio de estados, se destacó el rol de la rigidez tangente y del amortiguamiento dependiente de estado en sistemas histeréticos, así como la utilidad de los balances energéticos para interpretar la disipación y el daño acumulado. La revisión de modelos constitutivos, bilineales idealizados, Bouc–Wen y dispositivos disipativos como los aisladores de péndulo friccional, permitió explicitar los mecanismos de degradación asociados a la pérdida cíclica de rigidez y resistencia, responsables del *drift* residual, indicador estrechamente vinculado al desempeño post–sismo. En paralelo, se sistematizó el flujo del análisis dinámico no-lineal (NTHA) y su coste computacional, junto con la caracterización y preparación de registros naturales y sintéticos, subrayando la variabilidad entre excitaciones y sus implicancias en la generalización de modelos basados en datos. Sobre esta base, se introdujeron las arquitecturas de aprendizaje profundo (MLP, CNN, LSTM y Transformers) y, en particular, los enfoques informados por la física, donde las PINNs, tanto en su vertiente *guided-informed* como *solver-informed*, incorporan sesgos inductivos que restringen las hipótesis de aprendizaje a trayectorias físicamente admisibles, reducen la dependencia de grandes volúmenes de datos y promueven estabilidad numérica en horizontes largos (Raissi et al., 2019; Karniadakis et al., 2021).

En este contexto, las PINNs ofrecen un marco idóneo para analizar el *drift* residual en presencia de no-linealidad intensa, aunque su desempeño depende de la correcta formulación de las pérdidas físicas y del equilibrio entre información empírica y restricciones teóricas. Su principal ventaja radica en la posibilidad de penalizar directamente la ecuación de movimiento y las relaciones constitutivas, lo que fomenta la coherencia entre cinemática, equilibrio y evolución interna, reduciendo la deriva de fase y amplitud típica de modelos puramente *data–driven* en horizontes prolongados. Además, la incorporación de variables latentes, como la fuerza restauradora o los parámetros histeréticos, aportan interpretabilidad y control sobre la evolución del daño, permitiendo extender la función de pérdida con términos específicos (por ejemplo, coherencia cinemática o consistencia de energía) sin comprometer la estabilidad general del modelo.

A partir de estas bases, el **Capítulo 3** implementará y evaluará dos tipos de arquitecturas de referencia informadas por la física: *PhyCNN* y *PhyLSTM* (Zhang et al., 2020a; Zhang et al., 2020b). En cada caso se documentará el preprocesamiento, la partición de datos y las métricas de evaluación empleadas (L2RE,

NRMSE, NMAE y errores absolutos/relativos del *drift* residual). Los ensayos incluirán sistemas SDOF con histéresis representativa y distintos niveles de degradación, con el propósito de analizar de forma sistemática el alcance y las limitaciones de cada arquitectura bajo demandas crecientes .

Sobre esta evaluación comparativa, el **Capítulo 4** presentará un modelo alternativo orientado a subsanar las brechas identificadas en las líneas base. En particular, se introducirán: (i) un término explícito en la función de pérdida asociado al *drift* residual y al equilibrio dinámico final; (ii) una formulación de degradación informada por energía, destinada a penalizar inconsistencias entre disipación modelada y predicha; y (iii) un esquema de avance temporal con sesgo integrador (por ejemplo, un bloque RK–informado o un esquema implícito liviano) que refuerce la estabilidad en secuencias extensas. Se explorará, además, una estrategia de ponderación adaptativa entre términos físicos y de datos, buscando mejorar de manera conjunta la precisión del *drift* residual y la fidelidad transitoria, manteniendo un coste computacional contenido frente al NTHA de referencia. El capítulo culminará con un análisis de sensibilidad destinado a evaluar la robustez y la capacidad de generalización del modelo fuera del dominio de entrenamiento.

En síntesis, este capítulo proporciona la justificación conceptual y técnica que sustenta la adopción de modelos informados por la física, en particular PhyCNN y PhyLSTM, como líneas base para la predicción de respuestas altamente no-lineales con énfasis en el *drift* residual. A la vez, delimita las condiciones, supuestos y vacíos que motivan la formulación del modelo propuesto en el Capítulo 4 y establece el puente metodológico hacia la validación experimental y comparativa desarrollada en el Capítulo 3.

## Capítulo 3

# Estudio Comparativo de Arquitecturas Recurrentes y Convolucionales Informadas por la Física bajo Demandas Sísmicas Severas

### 3.1. Introducción

La predicción precisa de la respuesta dinámica de estructuras ante movimientos sísmicos constituye uno de los desafíos centrales de la ingeniería estructural contemporánea. Los métodos numéricos de integración paso a paso, como los esquemas de Newmark y Runge–Kutta, han sido históricamente la herramienta predominante para capturar el comportamiento no-lineal e hysterético; sin embargo, su costo computacional crece de forma significativa en modelos de gran escala, con formulaciones de elementos finitos, modelos constitutivos complejos o en análisis probabilísticos basados en conjuntos extensos de registros sísmicos. En paralelo, los avances recientes en aprendizaje automático han abierto un nuevo campo de investigación orientado a utilizar modelos de aprendizaje profundo para estimar la respuesta estructural con bajo costo de inferencia, lo que ha impulsado una literatura emergente y cada vez más sofisticada en ingeniería sísmica computacional.

Los primeros desarrollos relevantes en este ámbito se apoyaron en arquitecturas que aprenden directamente el mapeo entrada–salida a partir de datos, sin imponer de manera explícita las leyes de la dinámica estructural. En una etapa inicial, los enfoques basados en máquinas de soporte vectorial (SVM) y redes neuronales artificiales (ANN) demostraron la viabilidad de aproximar no-linealidades (Yinfeng et al., 2008; Ying et al., 2009; Lagaros y Papadrakakis, 2012). Posteriormente, la introducción de redes LSTM por Zhang et al. (2019) representó un avance sustantivo, al lograr predecir respuestas sísmicas no-lineales con mejor precisión y menor tiempo de inferencia que modelos MLP. Estos avances se extendieron hacia la simulación híbrida en tiempo real, donde las LSTM se emplearon como metamodelos computacionales (Bas et al., 2020; Bas y Moustafa, 2020), así como en análisis explícitos de edificios aislados (Mekaoui y Saito, 2022) y en ensayos con interacción suelo–estructura (Al-Subaihawi et al., 2024). De forma complementaria, Xu et al. (2022) propusieron una LSTM recursiva capaz de emular la integración paso a paso para longitudes y tasas de muestreo arbitrarias, logrando reducciones sustantivas de costo frente a los métodos de elementos finitos, aunque manteniendo su naturaleza *black-box*. Por otro lado, Ning et al. (2023) evaluaron comparativamente arquitecturas LSTM, WaveNet y CNN bidimensionales para la predicción de historias temporales no-lineales en sistemas estructurales de

diferente complejidad (SDOF, marcos de acero y puentes multicomponente), demostrando que la combinación de convoluciones causales y unidades recurrentes amplía el campo de dependencia temporal y mejora la representación de los desplazamientos residuales frente a las LSTM convencionales. De forma complementaria, Liao et al. (2023) incorporaron mecanismos de atención al marco recurrente clásico, proponiendo una red *Attention-based LSTM* (AttLSTM) capaz de ponderar componentes informativos dentro de las secuencias sísmicas para mejorar el aprendizaje a partir de datos limitados. Su aplicación a puentes tipo viga y atirantados demostró una mayor precisión y estabilidad frente a las LSTM convencionales, lo que evidencia el potencial de los esquemas con atención para capturar dependencias no estacionarias y características no-lineales de las respuestas estructurales.

En paralelo, se desarrollaron predictores de parámetros globales, como derivas o desplazamientos máximos, a partir de descriptores de la acción sísmica (Oh et al., 2020; Nguyen et al., 2022). En una fase más reciente, arquitecturas del tipo Transformer comenzaron a emplearse en dinámica estructural, tanto para estimar respuesta elastoplástica y clasificar daño (Chen et al., 2023), como para predecir trayectorias con derivas residuales mediante configuraciones híbridas Transformer–LSTM (Zhang et al., 2024). Si bien estos modelos no incorporan restricciones físicas en la función de pérdida, destacan por su capacidad para capturar dependencias de largo alcance, aunque a un costo computacional superior al de arquitecturas recurrentes o convolucionales más ligeras. En conjunto, los modelos puramente *data-driven* han demostrado alta eficiencia para aproximar dinámicas complejas; sin embargo, su desempeño depende críticamente de la disponibilidad y cobertura de datos y, al carecer de restricciones físicas explícitas, no garantizan el cumplimiento de las leyes de la dinámica. En consecuencia, su capacidad de generalización se debilita al extrapolar hacia escenarios no representados en el entrenamiento, en particular cuando emergen histéresis y derivas residuales en la respuesta estructural.

Con el propósito de superar estas limitaciones, ha emergido en los últimos años una corriente que integra conocimiento físico en el aprendizaje profundo (Raissi et al., 2019; Karniadakis et al., 2021). Este enfoque, denominado *physics-informed machine learning*, introduce la ecuación de movimiento como restricción adicional en la función de pérdida, de modo que las predicciones no solo se ajusten a los datos observados, sino que respeten las leyes fundamentales de la dinámica estructural. En 2020, las contribuciones de Zhang et al. (2020a) y Zhang et al. (2020b) consolidaron esta tendencia al proponer, respectivamente, la *PhyCNN* y la red *multi-LSTM*, ambas con pérdidas físicas diseñadas para capturar la dinámica estructural a partir de datos escasos, incluyendo además el modelado de variables latentes como las fuerzas restauradoras. En ese mismo año, Yu et al. (2020) desarrollaron un esquema híbrido RNN–MLP con términos físicos, reforzando empíricamente la pertinencia de incorporar conocimiento de la física en el entrenamiento. Trabajos posteriores ampliaron esta línea denominada *guided-informed* donde los modelos ajustan la función de pérdida a la ecuación de movimiento, tanto en regímenes lineales como no-lineales, validados mediante evidencia numérica y experimental (Liu et al., 2023). En una vertiente cercana, se han propuesto esquemas de reconstrucción de respuesta a partir de señales medidas, en particular mediante la fusión de deformaciones y aceleraciones para inferir desplazamientos (Xu et al., 2024).

En paralelo conceptual, pero con una filosofía arquitectónica distinta, se han consolidado los enfoques *solver-informed*, que incorporan explícitamente operadores de avance temporal en la red para emular integradores numéricos y mejorar la interpretabilidad y estabilidad en trayectorias largas. Dentro de este conjunto se

incluyen propuestas inspiradas en integradores explícitos e implícitos, fórmulas de Runge–Kutta y esquemas recurrentes hibridados en espacio de estados (Sadeghi Eshkevari et al., 2021; Zhai et al., 2023; Su et al., 2024; Xing et al., 2024; Zhai et al., 2024). De forma complementaria, las *Neural ODEs* y los marcos residuales de aprendizaje profundo ofrecen rutas alternativas para la identificación estructural y la predicción autoregresiva de largo alcance, aunque a costa de supuestos más exigentes sobre los estados observables (Lai et al., 2021; Guo et al., 2023).

En este contexto, pueden distinguirse dos familias conceptuales principales: (i) las *PINNs guided-informed*, que mantienen arquitecturas de propósito general (CNN o LSTM) pero incorporan la ecuación de movimiento como restricción blanda en la función de pérdida, siendo Zhang et al. (2020a) y Zhang et al. (2020b) sus referentes más influyentes; y (ii) las *solver-informed*, que emulan explícitamente integradores numéricos en la propia arquitectura. El presente trabajo se concentra exclusivamente en la primera familia, y en particular en la evaluación empírica de las propuestas *PhyCNN* y *multi-LSTM* de Zhang, consideradas aquí como modelos de referencia dentro de los enfoques *guided-informed*. Los modelos *solver-informed*, las *Neural ODEs* y las arquitecturas con atención se discuten únicamente para contextualizar el estado del arte, pero no forman parte del conjunto de modelos evaluados.

A pesar de los avances comunicados en la reducción de la dependencia de datos y en la mejora de la precisión, persiste la interrogante respecto de la capacidad de las arquitecturas *guided-informed* cuando se enfrentan a condiciones de demanda que desafían sus supuestos de entrenamiento. En particular, interesa examinar su generalización frente a registros sísmicos de larga duración y respuestas con histéresis y derivas residuales, rasgos que no siempre estuvieron presentes en las validaciones originales. Esta brecha resulta especialmente relevante en el contexto chileno, donde los registros sísmicos extensos pueden inducir residuales significativos en sistemas aislados o en osciladores no-lineales representativos, poniendo a prueba la extrapolación de modelos entrenados en dominios más acotados.

Ante este escenario, se plantea una evaluación comparativa de los modelos *PhyCNN* y *multi-LSTM* bajo escenarios progresivamente más demandantes, con el propósito de determinar su margen de validez, identificar fallas y orientar extensiones específicas para abordarlas. En consecuencia, este capítulo busca responder la siguiente pregunta central:

*¿Hasta qué punto los modelos PINNs guided-informed de referencia en la literatura, específicamente PhyCNN y multi-LSTM, son capaces de predecir con precisión las respuestas dinámicas de sistemas estructurales hysteréticos sometidos a terremotos chilenos de larga duración, y cuáles son sus limitaciones al enfrentar escenarios con desplazamientos residuales y comportamientos no-lineales complejos?*

El análisis se desarrolla sobre tres conjuntos de datos representativos: (i) un oscilador SDOF no-lineal no hysterético tipo Duffing excitado con registros reales; (ii) un sistema Bouc–Wen SDOF simplificado bajo ruido blanco filtrado; y (iii) un sistema Bouc–Wen SDOF representativo de un edificio aislado sísmicamente sometido a terremotos chilenos de larga duración. La motivación de esta evaluación es doble: por un lado, delimitar de forma transparente las condiciones bajo las cuales estos modelos funcionan adecuadamente en dominios lineales o sin residuales; y por otro, identificar las fallas asociadas a cambios del eje de oscilación

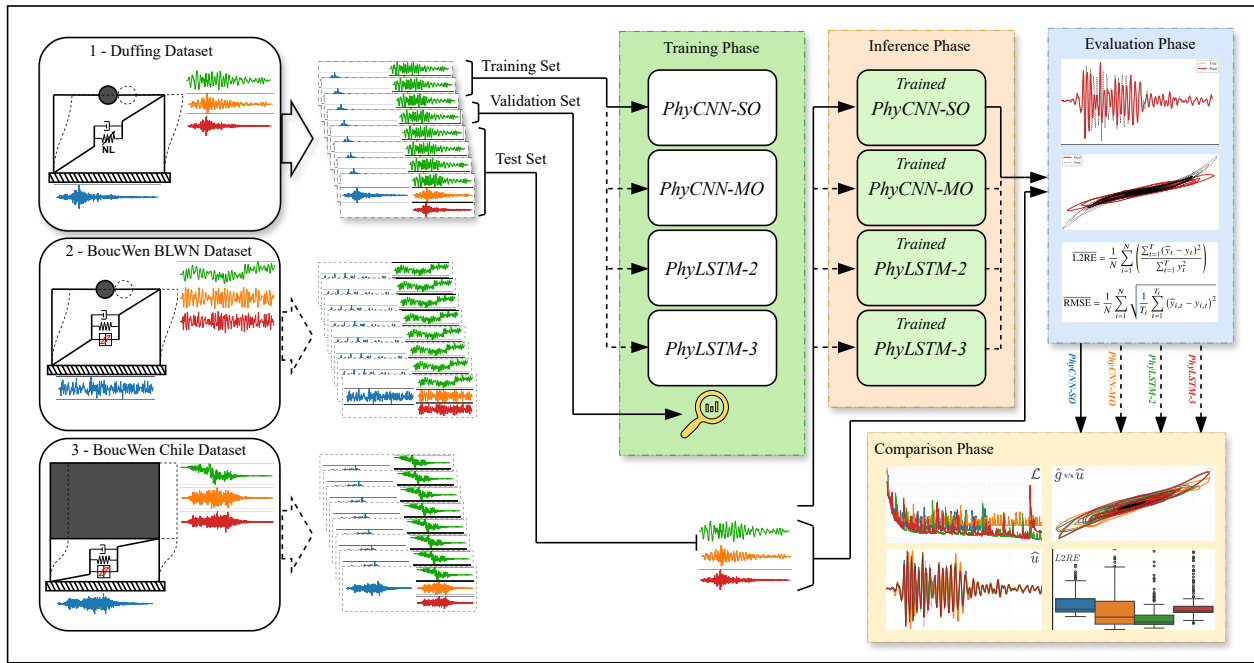


Figura 3.1: Esquema general de la metodología. Se consideran tres conjuntos de datos (Duffing, Bouc–Wen BLWN y Bouc–Wen Chile) que alimentan las fases de entrenamiento, inferencia y evaluación de cuatro arquitecturas *guided-informed*.

que orienten el desarrollo de mejoras metodológicas. En términos más amplios, clarificar estas condiciones de validez resulta esencial para evaluar la aplicabilidad de las técnicas *physics-informed* en el análisis del riesgo y en el diseño sismorresistente de estructuras.

Este capítulo se organiza del siguiente modo. La Sección 3.2 presenta la formulación adoptada, el modelo *surrogado* secuencia–a–secuencia, las pérdidas (de datos, física y cinemática), las arquitecturas *guided-informed* consideradas y las métricas/indicadores de evaluación. La Sección 3.3 describe los conjuntos de datos y sistemas analizados; a continuación, las Secciones 3.4–3.6 desarrollan los resultados y el análisis comparativo para cada escenario (Duffing; Bouc–Wen con BLWN; Bouc–Wen representativo con registros chilenos). Finalmente, la Sección 3.7 sintetiza los hallazgos, discute limitaciones y plantea recomendaciones.

## 3.2. Metodología

La metodología desarrollada tiene por objetivo evaluar el desempeño de distintas arquitecturas neuronales informadas por la física en la predicción historia–temporal de respuestas estructurales no-lineales sometidas a excitaciones sísmicas. El procedimiento general se ilustra en la Figura 3.1, donde se resumen las etapas de construcción de bases de datos, entrenamiento, inferencia, evaluación y comparación de resultados.

Para el análisis comparativo se definieron tres sistemas estructurales de un grado de libertad (SDOF), seleccionados de acuerdo con su capacidad de representar distintos niveles de no-linealidad y mecanismos de disipación de energía. El primero corresponde a un oscilador no histerético tipo Duffing, el segundo a un

sistema histerético Bouc–Wen excitado mediante ruido blanco filtrado (BLWN), y el tercero a un sistema Bouc–Wen representativo de un edificio aislado sísmicamente, sometido a registros reales de terremotos chilenos. Cada sistema fue analizado numéricamente para generar su respectiva base de datos, compuesta por pares de señales de excitación sísmica y respuestas estructurales asociadas, incluyendo desplazamientos, velocidades y fuerzas internas normalizadas.

Estas tres bases de datos se utilizaron de manera independiente para entrenar y evaluar cuatro arquitecturas neuronales informadas por la física: *PhyCNN–SO*, *PhyCNN–MO*, *PhyLSTM–2* y *PhyLSTM–3*. Cada modelo se implementó siguiendo fielmente las configuraciones propuestas por sus autores, conforme al código fuente público disponible, manteniendo la estructura y función de pérdida originales salvo ajustes específicos que se detallan más adelante.

El proceso de entrenamiento, inferencia y evaluación se llevó a cabo en un entorno de cómputo con las siguientes características: sistema operativo Ubuntu 24.04.3 LTS, procesador Intel(R) Core(TM) i5–12400 (12th Gen), y tarjeta gráfica NVIDIA GeForce RTX 3060 con 12 GB de memoria dedicada.

Durante la fase de entrenamiento se empleó una fracción reducida del total de registros disponibles, con el fin de representar un escenario de limitada disponibilidad de datos, tal como suele ocurrir en la práctica de la ingeniería estructural. El conjunto de validación, de menor tamaño que el de entrenamiento, se utilizó únicamente con fines de monitoreo del proceso de ajuste, sin intervenir en los criterios de parada temprana ni en el guardado de pesos, en concordancia con la configuración original de los modelos base. Posteriormente, la fase de inferencia se realizó sobre el conjunto de prueba, compuesto por el mayor número de registros, a partir de los cuales cada modelo predijo las tres variables dinámicas principales: desplazamiento relativo, velocidad y fuerza interna normalizada.

La etapa de evaluación consistió en comparar las predicciones con sus valores de referencia mediante métricas cuantitativas y análisis gráficos. Entre las métricas empleadas se incluyen el error cuadrático medio normalizado (NRMSE) y el error relativo cuadrático logarítmico (L2RE), los cuales permiten cuantificar la precisión temporal de las respuestas. Adicionalmente, se representaron las historias de desplazamiento, velocidad y fuerza, así como las curvas de histéresis desplazamiento–fuerza, con el fin de evaluar la capacidad de los modelos para reproducir comportamientos no-lineales y desplazamientos residuales. Finalmente, los resultados de las cuatro arquitecturas fueron comparados entre sí para cada base de datos, identificando fortalezas y limitaciones relativas.

### 3.2.1. Formulación física y modelo secuencia–a–secuencia

El análisis parte de la ecuación de movimiento de un sistema estructural de un grado de libertad (SDOF) en coordenadas relativas  $u(t)$  sometido a la aceleración del terreno  $a_g(t)$ :

$$m \ddot{u}(t) + c \dot{u}(t) + f_{\text{int}}(u(t), \dot{u}(t), z(t); \psi) = -m a_g(t), \quad (3.1)$$

donde  $m$  y  $c$  representan la masa y el coeficiente de amortiguamiento, respectivamente;  $f_{\text{int}}$  es la fuerza interna resultante, que puede incluir rigidez lineal o no-lineal, y efectos histeréticos descritos mediante variables

internas  $z(t)$ ; y  $\psi$  agrupa los parámetros físicos del modelo constitutivo (por ejemplo, rigideces, ratios posfluencia o exponentes del modelo de Bouc–Wen).

Cuando el amortiguamiento se incluye dentro de las fuerzas internas, la ecuación (3.1) adopta la forma compacta

$$m \ddot{u}(t) + f_{\text{int}}(u(t), \dot{u}(t), z(t); \psi) = -m a_g(t). \quad (3.2)$$

El cumplimiento de esta relación puede verificarse mediante el *residuo dinámico*,

$$r(t) = m \ddot{u}(t) + f_{\text{int}}(u(t), \dot{u}(t), z(t); \psi) + m a_g(t), \quad r(t) \equiv 0, \quad (3.3)$$

que, al normalizar por la masa, se reescribe como

$$r_{\text{norm}}(t) = \ddot{u}(t) + g(u(t), \dot{u}(t), z(t); \psi) + a_g(t), \quad g(u, \dot{u}, z; \psi) = \frac{f_{\text{int}}(u, \dot{u}, z; \psi)}{m}. \quad (3.4)$$

La condición  $r_{\text{norm}}(t) \equiv 0$  caracteriza el equilibrio dinámico del sistema y será la base de la restricción física impuesta a las redes neuronales durante el entrenamiento.

A partir de esta formulación, se define un modelo *surrogado* que busca aproximar el operador dinámico que transforma la excitación sísmica en la respuesta estructural. En términos generales, el modelo se plantea como una función no-lineal parametrizada por  $\theta$ :

$$\mathcal{N}_\theta : \mathbb{R}^{n_x \times T} \longrightarrow \mathbb{R}^{n_y \times T}, \quad \hat{\mathbf{Y}} = \mathcal{N}_\theta[\mathbf{X}],$$

donde la entrada  $\mathbf{X} = \mathbf{a}_g$  corresponde a la secuencia de aceleración del terreno, y la salida  $\hat{\mathbf{Y}}$  agrupa las respuestas estructurales predichas: desplazamiento relativo  $\hat{\mathbf{u}}$ , velocidad  $\hat{\mathbf{v}}$ , aceleración  $\hat{\mathbf{a}}$  y fuerza interna normalizada  $\hat{\mathbf{g}}$ , todas definidas en el dominio temporal.

El modelo se entrena a partir de un conjunto de registros sísmicos y sus correspondientes respuestas numéricas de referencia. Estos registros se dividen en tres subconjuntos: entrenamiento, validación y prueba. El entrenamiento utiliza un total de  $N_{\text{tr}}$  registros con etiquetas conocidas, mientras que un conjunto adicional de  $N_c$  registros sin respuesta (puntos de colación) se emplea para imponer el cumplimiento de las restricciones físicas. De este modo, el modelo aprovecha tanto información observacional como conocimiento físico, combinando ajuste a datos y regularización basada en la ecuación de movimiento.

El proceso de optimización de los parámetros  $\theta$  se guía por la minimización de una función de pérdida total compuesta por tres términos complementarios:

$$\mathcal{L}(\theta) = \underbrace{\frac{1}{N_{\text{tr}}T} \sum_{e,t} (\hat{y}_t^{(e)} - y_t^{(e)})^2}_{\mathcal{L}_{\text{dat}}} + \underbrace{\frac{1}{N_c T} \sum_{e,t} (\hat{a}_t^{(e)} + \hat{g}_t^{(e)} + a_{g,t}^{(e)})^2}_{\mathcal{L}_{\text{fis}}} + \underbrace{\frac{1}{N_c T} \sum_{e,t} (\hat{v}_t^{(e)} - \dot{u}_t^{(e)})^2}_{\mathcal{L}_{\text{cin}}}, \quad (3.5)$$

donde:  $\mathcal{L}_{\text{dat}}$  penaliza la discrepancia entre las predicciones y las respuestas de referencia,  $\mathcal{L}_{\text{fis}}$  impone el cumplimiento de la ecuación de movimiento en los puntos de colación, y  $\mathcal{L}_{\text{cin}}$  refuerza la coherencia cinemática entre los desplazamientos predichos y sus derivadas temporales.

En conjunto, estos términos definen una estrategia *guided-informed*, en la cual las leyes de la dinámica estructural actúan como una restricción blanda sobre las predicciones de la red. Esto permite incorporar información física incluso en ausencia de datos etiquetados, mejorando la estabilidad y la generalización del modelo frente a condiciones no vistas.

El cálculo de las derivadas temporales necesarias para la evaluación de la pérdida cinemática se implementa mediante un operador discreto  $\Phi_t$ , basado en diferencias finitas mixtas de segundo orden:

$$\dot{\mathbf{u}} = \Phi_t \mathbf{u}, \quad \ddot{\mathbf{u}} = \Phi_t^2 \mathbf{u}.$$

Este esquema utiliza diferencias adelantadas al inicio, centradas en el dominio interior y atrasadas en los extremos, asegurando estabilidad numérica y consistencia con las restricciones físicas. Al estar implementado dentro del grafo computacional, el operador  $\Phi_t$  conserva la diferenciabilidad del modelo y permite la propagación del gradiente durante el entrenamiento, sin requerir expresiones analíticas de las derivadas.

### 3.2.2. Redes Neuronales Convolucionales 1D y LSTM

Con la formulación física y el esquema secuencia–a–secuencia definidos en la Sección 3.2.1, es necesario un aproximador capaz de transformar las aceleraciones del terreno en variables de respuesta, preservando la coherencia temporal y garantizando estabilidad numérica durante el proceso de inferencia. En este contexto, las redes neuronales profundas constituyen una familia de modelos capaces de aprender representaciones jerárquicas de los datos mediante la composición de múltiples capas no-lineales.

Entre las distintas arquitecturas existentes, las redes convolucionales unidimensionales (CNN 1D) y las redes de memoria a largo corto plazo (LSTM) destacan por su capacidad para modelar relaciones temporales y dependencias locales en secuencias. En esta subsección se describen los fundamentos teóricos de ambas estructuras, que serán utilizadas como aproximadores dentro del marco general propuesto.

#### Redes Neuronales Convolucionales 1D

Las *Convolutional Neural Networks* (CNN) aplican operaciones de convolución sobre los datos de entrada con el objetivo de extraer características locales relevantes. En una CNN unidimensional, los filtros se desplazan a lo largo de una sola dimensión, permitiendo capturar patrones espaciales o temporales dependientes del orden de los datos.

Formalmente, una convolución unidimensional transforma una señal multicanal  $\mathbf{X} \in \mathbb{R}^{D \times T}$  en mapas de características  $\mathbf{Y} \in \mathbb{R}^{D' \times T'}$  mediante la aplicación de un conjunto de filtros  $\mathbf{W} \in \mathbb{R}^{D' \times D \times K}$  y un sesgo  $\mathbf{b} \in \mathbb{R}^{D'}$ . En este contexto,  $D$  denota el número de canales de entrada,  $T$  la longitud de la secuencia,  $K$  el tamaño del filtro y  $D'$  el número de filtros o canales de salida. El proceso puede expresarse como:

$$y_i^{(d')} = \sigma \left( \sum_{d=1}^D \sum_{\tau=0}^{K-1} W_{d,\tau}^{(d')} X_{d,i+\tau} + b^{(d')} \right),$$

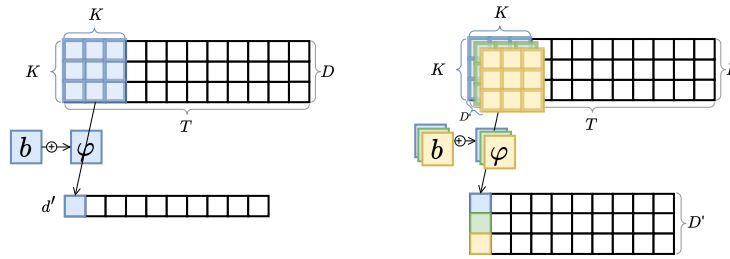


Figura 3.2: Capa convolucional 1D con múltiples entradas: un único filtro (izquierda) y múltiples filtros (derecha).

donde  $\sigma(\cdot)$  representa una función de activación no-lineal aplicada punto a punto, y  $(\cdot)^{(d')}$  indica el canal de salida correspondiente al filtro  $d'$ .

Cada filtro convolucional actúa como un detector de patrones, extrayendo características específicas de la entrada, mientras que la superposición de múltiples filtros permite representar información de diferentes niveles de abstracción. Para evitar que la longitud de la salida  $T'$  se reduzca tras la convolución, es habitual aplicar un procedimiento denominado *padding*, que consiste en extender la secuencia de entrada mediante la adición de ceros (u otros valores neutros) en sus extremos. De esta forma, la operación de convolución produce mapas de características con la misma longitud que la secuencia original, manteniendo la alineación temporal entre entrada y salida.

La Figura 3.2 ilustra el funcionamiento de una capa convolucional 1D, mostrando tanto el caso de un único filtro como el de múltiples filtros aplicados a una misma entrada.

## Redes LSTM

Las redes *Long Short-Term Memory* (LSTM), introducidas por Hochreiter y Schmidhuber (1997), constituyen una extensión de las redes neuronales recurrentes (RNN) diseñada para modelar dependencias temporales de corto y largo alcance. Su estructura interna incorpora mecanismos de control mediante compuertas que regulan el flujo de información dentro de la celda de memoria.

Cada celda LSTM contiene tres compuertas principales: de entrada ( $\mathbf{i}_t$ ), de olvido ( $\mathbf{f}_t$ ) y de salida ( $\mathbf{o}_t$ ), además del estado de la celda ( $\mathbf{c}_t$ ). Dadas una entrada  $\mathbf{x}_t \in \mathbb{R}^D$  y el estado oculto previo  $\mathbf{h}_{t-1} \in \mathbb{R}^{D'}$ , la dinámica interna de la celda puede describirse como:

$$\begin{aligned} \mathbf{i}_t &= \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i), & \mathbf{f}_t &= \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f), \\ \mathbf{o}_t &= \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o), & \tilde{\mathbf{c}}_t &= \tanh(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-1} + \mathbf{b}_c), \end{aligned}$$

y

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t, \quad \mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t),$$

donde  $\odot$  denota el producto elemento a elemento,  $\sigma(\cdot)$  es la función sigmoide y  $\tanh(\cdot)$  la tangente hiperbólica.

Es importante notar que la misma celda LSTM, definida por los parámetros  $\{\mathbf{W}_*, \mathbf{U}_*, \mathbf{b}_*\}$ , se aplica de manera recurrente a todos los pasos temporales  $t = 1, \dots, T$  de la secuencia de entrada. Esta reutilización

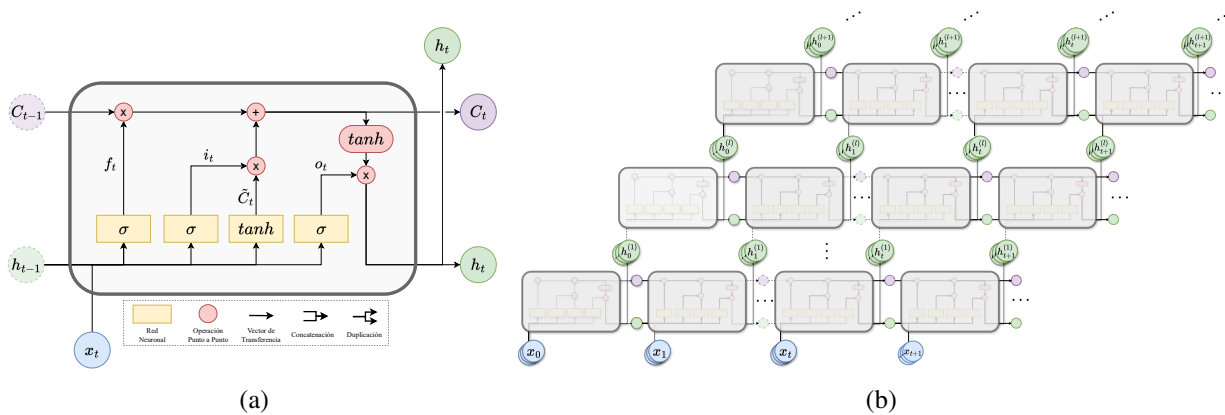


Figura 3.3: Esquema de LSTM: **(a)** Celda LSTM elemental: compuertas de entrada ( $i_t$ ), olvido ( $f_t$ ) y salida ( $o_t$ ) que regulan el estado de memoria  $c_t$ . y **(b)** Arquitectura profunda: apilamiento de capas para modelar dependencias de largo alcance.

de parámetros implica que la celda comparte la misma estructura y pesos en cada instante, generando así secuencias de salida  $\{\mathbf{h}_t\}_{t=1}^T$  y de estado de celda  $\{\mathbf{c}_t\}_{t=1}^T$  de la misma longitud que la secuencia de entrada. De esta forma, la LSTM procesa la información de manera temporalmente coherente, preservando las dependencias entre los distintos pasos. La Figura 3.3a representa esquemáticamente una celda LSTM, mostrando el flujo de información a través de sus compuertas.

Cuando se apilan múltiples capas LSTM ( $L > 1$ ), la capacidad de modelado se incrementa al permitir una jerarquía de representaciones temporales. En tal caso, el flujo de información entre capas puede formalizarse como:

$$\mathbf{h}_t^{(l)}, \mathbf{c}_t^{(l)} = \text{LSTM}^{(l)}(\mathbf{h}_{t-1}^{(l)}, \mathbf{c}_{t-1}^{(l)}, \mathbf{x}_t^{(l)}), \quad \mathbf{x}_t^{(l)} = \begin{cases} \mathbf{x}_t, & l = 1, \\ \mathbf{h}_t^{(l-1)}, & l > 1. \end{cases}$$

La Figura 3.3 muestra el esquema general de la arquitectura LSTM, incluyendo una versión profunda con múltiples capas apiladas.

### 3.2.3. Indicadores de desempeño y métricas de evaluación

Sea  $e_t = \hat{y}_t - y_t$  el error puntual en el instante  $t$ . Para comparar errores de señales de distinta escala, las métricas se expresan en forma adimensional mediante la normalización respecto al rango dinámico de la señal real,  $y_{\max} - y_{\min}$ . Todas las métricas se calculan sobre el conjunto de prueba considerando la secuencia completa de salida. La evaluación se basa en tres métricas normalizadas que reflejan la precisión y consistencia del modelo.

La **raíz del error cuadrático medio normalizado (NRMSE)** se define como

$$\text{NRMSE} = \frac{\text{RMSE}}{y_{\max} - y_{\min}}, \quad \text{RMSE} = \sqrt{\frac{1}{T} \sum_{t=1}^T e_t^2},$$

y representa la magnitud cuadrática media del error relativa al rango de la señal. Penaliza errores de gran amplitud y es sensible a picos locales. La desviación temporal asociada,  $\sigma_{\text{NRMSE}} = \sigma_{\text{RMSE}} / (y_{\text{máx}} - y_{\text{mín}})$ , cuantifica la variabilidad interna del error a lo largo de la secuencia.

El **error absoluto medio normalizado (NMAE)** se obtiene como

$$\text{NMAE} = \frac{\text{MAE}}{y_{\text{máx}} - y_{\text{mín}}}, \quad \text{MAE} = \frac{1}{T} \sum_{t=1}^T |e_t|,$$

y mide la magnitud promedio del error en términos relativos al rango de la señal. Es menos sensible a valores atípicos que la NRMSE, y su desviación  $\sigma_{\text{NMAE}}$  refleja la estabilidad temporal del ajuste.

Por último, el **error cuadrático relativo (L2RE)** se expresa como

$$\text{L2RE} = \frac{\|\hat{\mathbf{y}} - \mathbf{y}\|_2}{\|\mathbf{y}\|_2} = \frac{\sqrt{\sum_{t=1}^T e_t^2}}{\sqrt{\sum_{t=1}^T y_t^2}},$$

comparando la energía del error con la energía total de la señal real. Su carácter global y adimensional permite evaluar la fracción de energía no explicada por el modelo, siendo  $\text{L2RE} \ll 1$  indicativo de alta fidelidad relativa. A diferencia de las métricas normalizadas por el rango, el L2RE es inherentemente adimensional, al expresar la energía del error como fracción de la energía total de la señal real.

Cuando el modelo se aplica a  $N$  secuencias distintas, cada métrica  $E_i$  se calcula individualmente por cada muestra y se agregan el promedio y la desviación muestral:

$$\bar{E} = \frac{1}{N} \sum_{i=1}^N E_i, \quad \sigma_E = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (E_i - \bar{E})^2}.$$

El promedio  $\bar{E}$  representa el desempeño medio del modelo, mientras que  $\sigma_E$  caracteriza la dispersión del error entre diferentes realizaciones, reflejando su estabilidad estadística.

Para el caso de respuestas bidimensionales, las métricas anteriores se extienden considerando el error vectorial conjunto entre fuerza y desplazamiento.

### Evaluación bidimensional de desempeño

En respuestas no-lineales, la precisión del modelo se evalúa sobre las trayectorias fuerza–desplazamiento, donde  $\mathbf{y}_t = [u_t, g_t]^\top$  y  $\hat{\mathbf{y}}_t = [\hat{u}_t, \hat{g}_t]^\top$ . El error instantáneo vectorial es

$$\mathbf{e}_t = \hat{\mathbf{y}}_t - \mathbf{y}_t, \quad \|\mathbf{e}_t\|_2 = \sqrt{(\hat{u}_t - u_t)^2 + (\hat{g}_t - g_t)^2}.$$

El rango efectivo bidimensional se define como

$$R_{2D} = \sqrt{(\text{máx } u_t - \text{mín } u_t)^2 + (\text{máx } g_t - \text{mín } g_t)^2},$$

y permite expresar las métricas en forma normalizada.

La  $\text{NRMSE}_{2D}$  y la  $\text{NMAE}_{2D}$  se calculan respectivamente como

$$\text{NRMSE}_{2D} = \frac{\text{RMSE}_{2D}}{R_{2D}}, \quad \text{NMAE}_{2D} = \frac{\text{MAE}_{2D}}{R_{2D}},$$

con

$$\text{RMSE}_{2D} = \sqrt{\frac{1}{T} \sum_{t=1}^T \left[ (\hat{u}_t - u_t)^2 + (\hat{g}_t - g_t)^2 \right]}, \quad \text{MAE}_{2D} = \frac{1}{T} \sum_{t=1}^T \sqrt{(\hat{u}_t - u_t)^2 + (\hat{g}_t - g_t)^2}.$$

Ambas cuantifican la discrepancia geométrica entre la trayectoria real y la predicha, siendo la  $\text{NRMSE}_{2D}$  más sensible a desviaciones puntuales. Sus desviaciones temporales,  $\sigma_{\text{NRMSE}_{2D}}$  y  $\sigma_{\text{NMAE}_{2D}}$ , reflejan la consistencia del ajuste dentro del ciclo histerético.

Finalmente, el **error cuadrático relativo bidimensional ( $\text{L2RE}_{2D}$ )** compara la energía de la respuesta predicha con la real:

$$\text{L2RE}_{2D} = \frac{\sqrt{\sum_{t=1}^T \left[ (\hat{u}_t - u_t)^2 + (\hat{g}_t - g_t)^2 \right]}}{\sqrt{\sum_{t=1}^T (u_t^2 + g_t^2)}}.$$

Un valor bajo de  $\text{L2RE}_{2D}$  indica una representación precisa tanto en amplitud como en fase del comportamiento histerético.

Al igual que en el caso unidimensional, los valores reportados corresponden al promedio  $\bar{E}_{2D}$  y su desviación muestral  $\sigma_{E_{2D}}$ , que resumen el desempeño medio y la variabilidad entre muestras.

### Indicadores gráficos de desempeño

Además de las métricas numéricas, se emplean indicadores gráficos que permiten interpretar visualmente el comportamiento del modelo y complementar la evaluación cuantitativa.

- **Predicción vs. serie real:** se representa la señal predicha  $\hat{y}(t)$  junto a la señal de referencia  $y(t)$  en el dominio temporal, lo que permite identificar desviaciones locales, desfases o errores sistemáticos a lo largo del registro.
- **Distribución del error entre series:** el error promedio por secuencia  $\{E_i\}_{i=1}^N$  se resume mediante diagramas de caja (*boxplots*), permitiendo analizar la consistencia del modelo entre distintos registros e identificar posibles valores atípicos o muestras con desempeño degradado.
- **Densidad espectral de potencia (PSD):** para evaluar la preservación del contenido frecuencial, se comparan las densidades espectrales de potencia de la señal real y de la predicha,  $S_{yy}(f)$  y  $S_{\hat{y}\hat{y}}(f)$ , respectivamente. Este gráfico evidencia si el modelo reproduce adecuadamente la distribución de energía en el dominio de frecuencias.
- **Curvas de fuerza–desplazamiento:** en sistemas no-lineales, se comparan las trayectorias reales y predichas en el plano fuerza–desplazamiento, representadas por pares  $(u_t, g_t)$  y  $(\hat{u}_t, \hat{g}_t)$ , respectivamente.

La superposición de ambas curvas permite inspeccionar la capacidad del modelo para reproducir la forma, amplitud y fase del ciclo histerético.

Estos indicadores gráficos facilitan la identificación de patrones sistemáticos y posibles sesgos que podrían no reflejarse directamente en las métricas globales, proporcionando una visión más completa sobre la estabilidad, la fidelidad dinámica y la capacidad de generalización del modelo.

### 3.2.4. Arquitecturas *guided-informed*

Con el objetivo de comparar distintos esquemas de modelado físico-informado, se desarrollan cuatro arquitecturas complementarias: dos convolucionales (*PhyCNN-SO* y *PhyCNN-MO*) y dos recurrentes (*PhyLSTM<sup>2</sup>* y *PhyLSTM<sup>3</sup>*). Todas ellas se entrenan bajo la función de pérdida global definida en la Ecuación (3.5) (o variaciones de esta) y comparten el mismo esquema de diferenciación temporal para imponer las restricciones cinemáticas y dinámicas. Las arquitecturas convolucionales procesan la información de forma jerárquica y local, mientras que las recurrentes modelan dependencias temporales de largo alcance mediante celdas LSTM.

#### PhyCNN-MO (multisalida)

La red *PhyCNN-MO* es una arquitectura convolucional unidimensional diseñada para predecir simultáneamente el desplazamiento  $\hat{\mathbf{u}}$ , la velocidad  $\hat{\mathbf{v}}$  y la fuerza interna normalizada  $\hat{\mathbf{g}}$ . Su estructura, ilustrada en la Figura 3.4, está compuesta por cinco capas convolucionales 1D con 64 filtros y tamaño de kernel 40, manteniendo la longitud de las secuencias mediante *padding*. A continuación, se incorporan dos capas densas de 50 unidades aplicadas a cada paso temporal (*TimeDistributed*). Todas las capas, excepto la de salida, emplean una función de activación ReLU. La capa final genera tres canales correspondientes a las tres variables de salida.

Durante el entrenamiento, las pérdidas de datos, cinemática y dinámica se combinan de acuerdo con el esquema mostrado en la Figura 3.4, donde las derivadas temporales se obtienen numéricamente para aplicar las restricciones físicas.

#### PhyCNN-SO (salida única)

La arquitectura *PhyCNN-SO* constituye una versión simplificada de la anterior, diseñada para predecir únicamente el desplazamiento estructural  $\hat{\mathbf{u}}$ . Las restantes variables se reconstruyen mediante relaciones físicas: la velocidad como  $\hat{\mathbf{v}} = \Phi_t \hat{\mathbf{u}}$  y la fuerza interna normalizada como  $\hat{\mathbf{g}} = -\Phi_t^2 \hat{\mathbf{u}} - \mathbf{a}_g$ . La Figura 3.5 muestra su estructura, análoga a la de la *PhyCNN-MO*, pero con kernel de tamaño 50 y una única salida. En este caso, la función de pérdida no incorpora directamente la ecuación de movimiento como restricción, pero incluye la física mediante una penalización física sobre la aceleración derivada y una regularización sobre el desplazamiento inicial.

#### Bloque LSTM

Las arquitecturas recurrentes se construyen a partir de un bloque base LSTM, mostrado en la Figura 3.6. Este bloque está conformado por tres capas LSTM con 100 unidades ocultas cada una y activación ReLU, seguidas

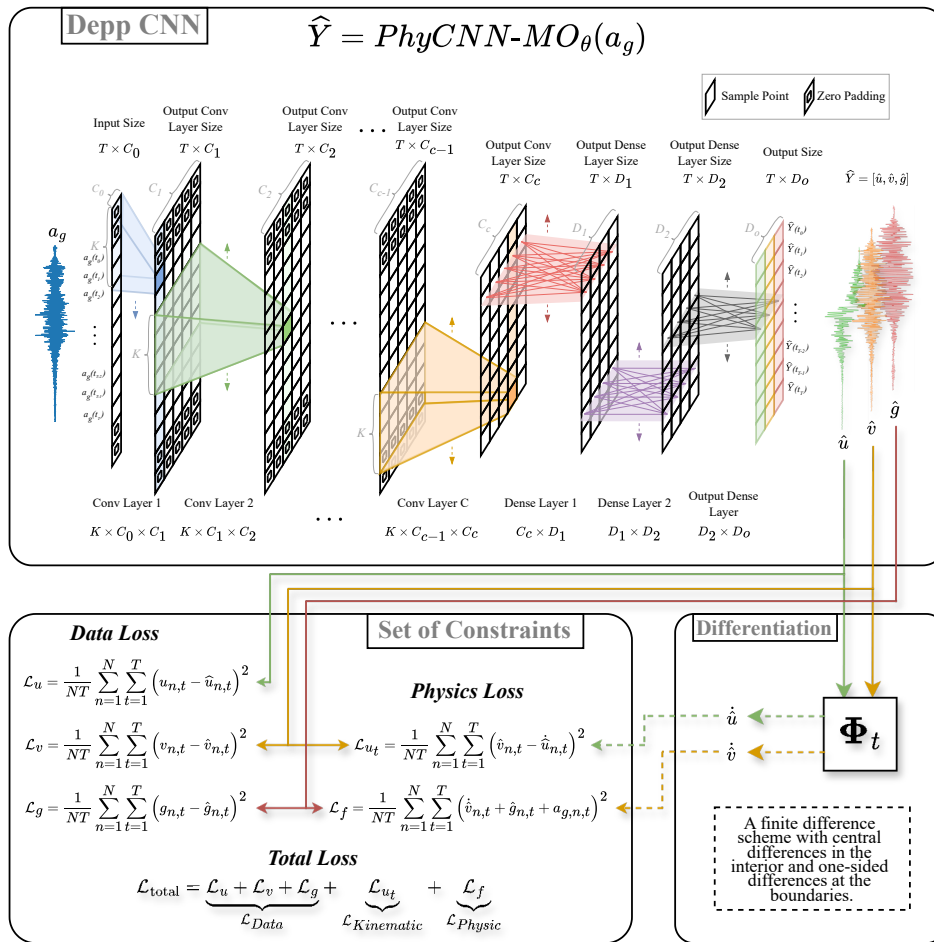


Figura 3.4: Arquitectura y marco de entrenamiento del modelo *PhyCNN-MO*. La red predice simultáneamente  $\hat{u}$ ,  $\hat{v}$ ,  $\hat{g}$  e impone restricciones cinemáticas y dinámicas mediante diferenciación numérica.

de una capa densa *TimeDistributed* de 100 unidades (presente únicamente en la arquitectura *PhyLSTM*<sup>2</sup>). La capa de salida genera entre uno y tres canales según el tipo de bloque. Cada bloque procesa secuencias completas y preserva la dimensionalidad temporal, lo que permite la integración directa de derivadas y la aplicación de restricciones físicas. Sobre este bloque se construyen las arquitecturas *PhyLSTM*<sup>2</sup> y *PhyLSTM*<sup>3</sup>, descritas a continuación.

### PhyLSTM<sup>2</sup>

La arquitectura *PhyLSTM*<sup>2</sup> implementa dos bloques LSTM en cascada. El primer bloque predice el desplazamiento  $\hat{u}$ , la velocidad  $\hat{v}$  y el parámetro de histéresis  $\hat{z}$  a partir de la excitación sísmica  $a_g$ , mientras que el segundo bloque estima la fuerza interna normalizada  $\hat{g}$  a partir de las tres secuencias generadas previamente. La función de pérdida incluye términos de ajuste de datos y penalizaciones asociadas a las restricciones cinemática y dinámica, según se ilustra en la Figura 3.7.

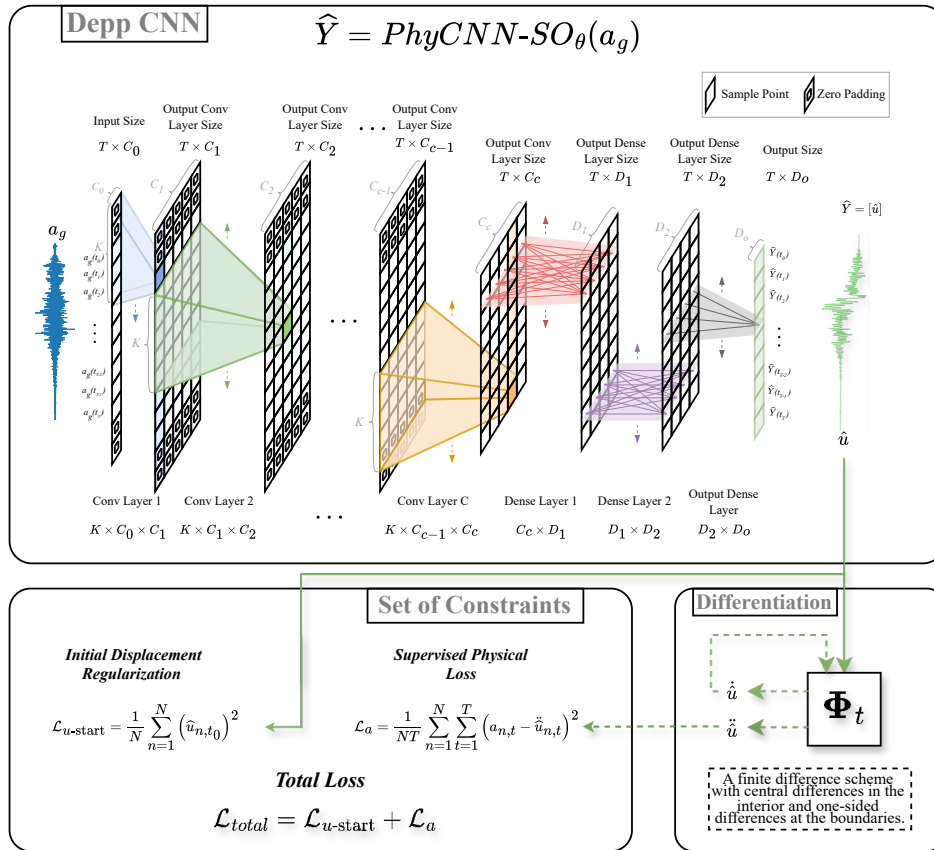


Figura 3.5: Arquitectura de la red *PhyCNN-SO*. Predice únicamente el desplazamiento estructural y aplica una restricción física supervisada sobre la aceleración derivada.

### PhyLSTM<sup>3</sup>

La arquitectura *PhyLSTM<sup>3</sup>* extiende el esquema anterior mediante la incorporación de un tercer bloque LSTM dedicado a la predicción de la evolución de una variable histerética latente  $\hat{\mathbf{z}}$ . Este bloque recibe como entrada el desplazamiento  $\hat{\mathbf{u}}$  y la variable de histéresis  $\hat{\mathbf{z}}$  estimada por el primer bloque. Durante el entrenamiento, se incluye una nueva restricción que penaliza la inconsistencia entre la derivada numérica  $\Phi_t \hat{\mathbf{z}}$  y la predicción directa  $\hat{\mathbf{z}}$ , garantizando así la coherencia temporal del parámetro histerético. La arquitectura completa se muestra en la Figura 3.8.

#### 3.2.5. Esquema de optimización y criterios de entrenamiento

Tanto las arquitecturas *PhyCNN* como las *PhyLSTM* emplean un esquema de optimización híbrido basado en la combinación de los algoritmos *Adam* y *L-BFGS-B*. El optimizador *Adam* se utiliza durante las primeras etapas del entrenamiento, aplicando un descenso estocástico del gradiente con tasa de aprendizaje constante de  $1 \times 10^{-3}$ , lo que permite una exploración eficiente del espacio de parámetros y una rápida convergencia inicial.

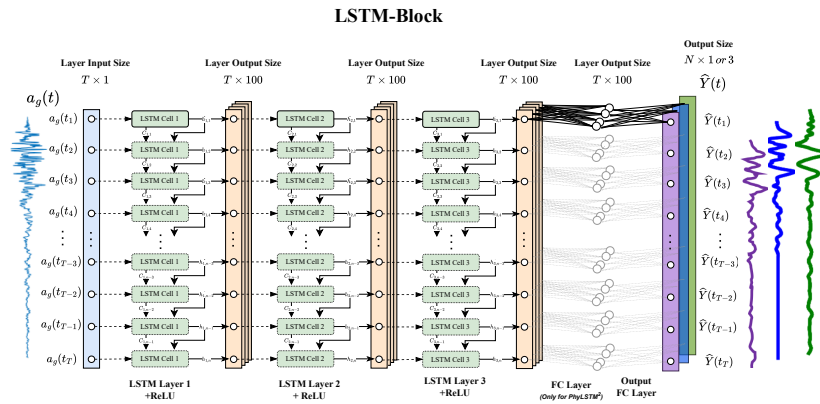


Figura 3.6: Bloque LSTM genérico utilizado en las arquitecturas *PhyLSTM*. Contiene tres capas LSTM (100 unidades) con activación ReLU y una capa densa *TimeDistributed*.

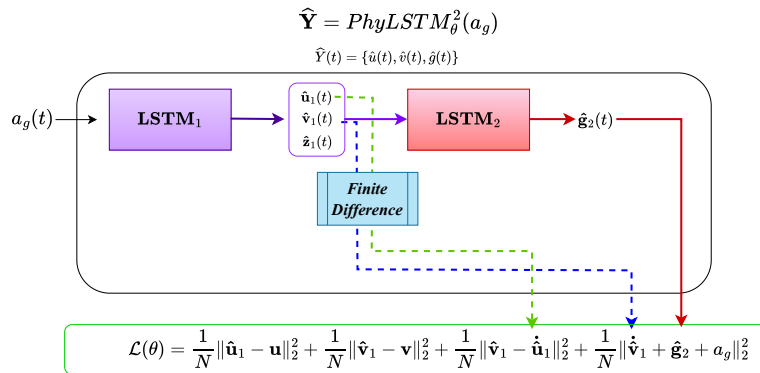


Figura 3.7: Arquitectura *PhyLSTM*<sup>2</sup>. Consta de dos bloques LSTM conectados en serie; el segundo bloque ajusta la fuerza interna  $\hat{\mathbf{g}}^{(2)}$  e impone la restricción dinámica.

Posteriormente, el optimizador *L-BFGS-B*, un método cuasi-Newton de segunda orden, refina los parámetros ajustando la solución hacia un mínimo local más estable mediante la aproximación iterativa del Hessiano.

En los códigos originales de las arquitecturas propuestas por Zhang et al. (2020a) y Zhang et al. (2020b), el conjunto de validación se define pero no cumple una función activa dentro del proceso de entrenamiento. En consecuencia, no se aplica ningún criterio de *early stopping*, y los pesos finales del modelo corresponden a los obtenidos en la última iteración, guiándose exclusivamente por la pérdida de entrenamiento. Además, el número de épocas se ajusta de manera distinta para cada conjunto de datos.

En este estudio se mantuvo la filosofía general de los trabajos originales, priorizando la reproducibilidad de los resultados, pero se incorporaron algunas modificaciones orientadas a mejorar la estabilidad y la comparabilidad de los modelos. En primer lugar, aunque el conjunto de validación continúa sin intervenir en las decisiones de parada temprana, se utiliza para monitorear el error de validación durante el entrenamiento, con el objetivo de analizar la evolución del sobreajuste sin alterar la optimización del modelo. En segundo lugar, se reemplazó el criterio de conservar los pesos de la última iteración por una estrategia más robusta:

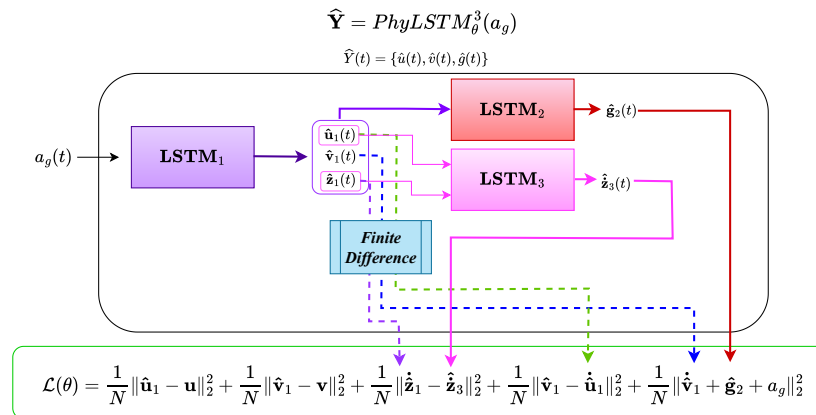


Figura 3.8: Arquitectura  $\text{PhyLSTM}^3$ . Introduce un tercer bloque LSTM para modelar la dinámica de la variable histerética latente  $\mathbf{z}$ , reforzando la consistencia temporal y ampliando la capacidad de representar efectos de memoria no-lineales.

almacenar los parámetros correspondientes a la *menor pérdida de entrenamiento* alcanzada. Esta decisión busca evitar que incrementos abruptos de la pérdida en las últimas iteraciones comprometan la calidad del modelo final, mejorando la estabilidad del proceso sin modificar el protocolo de entrenamiento original.

Finalmente, con el propósito de asegurar una base común para la comparación entre arquitecturas y conjuntos de datos, se definió un número estándar de 15 000 épocas para todos los modelos y bases sísmicas utilizadas en este estudio. Este valor garantiza una convergencia adecuada de la pérdida de entrenamiento bajo los criterios establecidos, aun cuando un número elevado de iteraciones no necesariamente implique una mejor generalización. De este modo, las adaptaciones introducidas conservan la coherencia metodológica del enfoque original, pero fortalecen su consistencia experimental y reproducibilidad en el contexto del presente análisis comparativo.

### 3.3. Casos de Estudio

El proceso de entrenamiento y validación de los modelos se llevó a cabo utilizando tres conjuntos de datos numéricos de creciente complejidad dinámica. Cada uno de ellos corresponde a la respuesta temporal de un sistema estructural de un grado de libertad (SDOF) sometido a excitaciones sísmicas o estocásticas, calculada mediante integración directa de la ecuación del movimiento. Los conjuntos se seleccionaron de modo que permitieran evaluar la capacidad de generalización de los modelos *guided-informed* bajo distintos regímenes de no-linealidad y tipos de excitación.

### 3.4. Escenario I: Oscilador no histerético tipo Duffing con registros reales

El primer conjunto de datos, originalmente propuesto por Zhang et al. (2020a), corresponde a un oscilador de Duffing caracterizado por una fuerza restauradora no-lineal de tipo cúbico. Su ecuación de movimiento se

expresa como:

$$m \ddot{\mathbf{u}} + c \dot{\mathbf{u}} + k_1 \mathbf{u} + k_2 \mathbf{u}^3 = -m \mathbf{a}_g,$$

donde  $m = 1.0 \text{ kg}$ ,  $c = 1.0 \text{ N.s/m}$ ,  $k_1 = 20.0 \text{ N/m}$  y  $k_2 = 200.0 \text{ N/m}$ . Estos parámetros definen un sistema con rigidez positiva y comportamiento de endurecimiento (*hardening*), sin dependencia de la historia de carga, tal como se ilustra en la Figura 3.9.

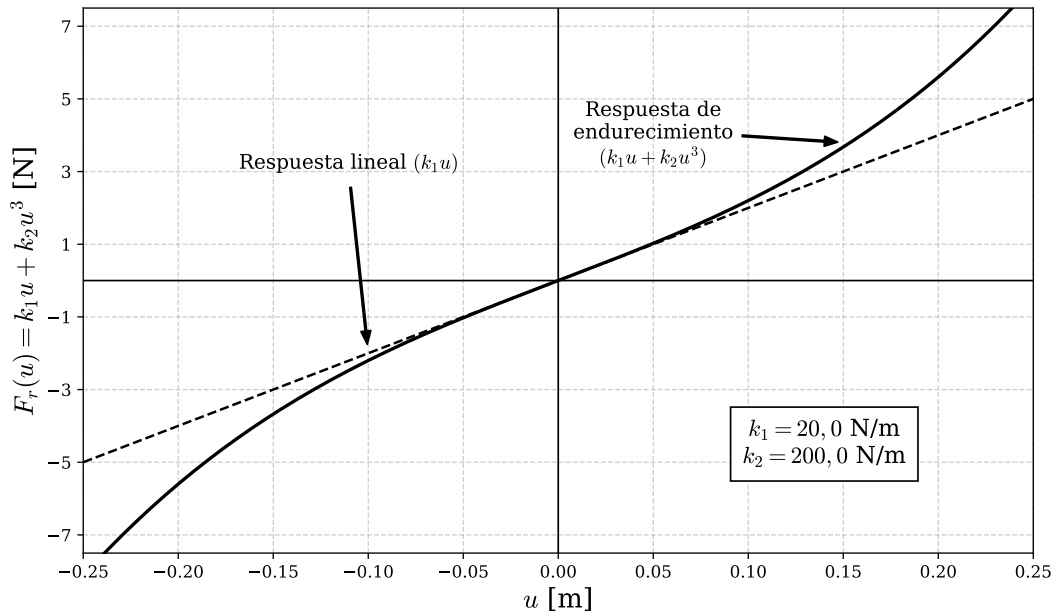


Figura 3.9: Esquema de la fuerza restauradora vs. desplazamiento para el oscilador de Duffing (Escenario I). Basado en los parámetros técnicos de Zhang et al. (2020a).

Las excitaciones empleadas corresponden a registros sísmicos reales seleccionados de la *PEER Strong Motion Database*, representativos de una probabilidad de excedencia del 10% en 50 años. Cada acelerograma tiene una duración de 50 s y una frecuencia de muestreo de 20 Hz, resultando en un total de  $T = 1001$  instantes discretos.

El conjunto completo comprende 100 simulaciones independientes. De ellas, ocho se emplearon para entrenamiento, dos para validación y las 90 restantes se reservaron para prueba. Para el ajuste de las restricciones físicas se utilizó un conjunto de colocación construido a partir de los 10 registros de entrenamiento y validación junto con 50 registros adicionales seleccionados del conjunto de prueba, los cuales solo fueron considerados para el ajuste de la restricción física del modelo.

Las variables de salida incluyen el desplazamiento  $\mathbf{u}$ , la velocidad  $\dot{\mathbf{u}}$  y la aceleración relativa  $\ddot{\mathbf{u}}$  (equivalente a la fuerza inercial normalizada por la masa), todas expresadas en unidades del Sistema Internacional. Este escenario constituye el caso base del estudio, caracterizado por una no-linealidad suave y la ausencia de efectos histeréticos, lo que permite aislar el desempeño de los modelos bajo una dinámica esencialmente no histerética.

### 3.4.1. Desempeño de PhyCNN y Multi-LSTM en el Escenario I

El desempeño predictivo de las arquitecturas se evaluó sobre el conjunto de prueba considerando las tres variables de salida: desplazamiento  $\hat{\mathbf{u}}$ , velocidad  $\hat{\mathbf{v}}$  y fuerza interna normalizada  $\hat{\mathbf{g}}$ . Como métrica principal de comparación se adoptó el error relativo cuadrático medio (L2RE), complementado por la raíz del error cuadrático medio normalizado (NRMSE) y el error absoluto medio normalizado (NMAE), a fin de proporcionar una medida consistente de precisión y dispersión.

Para los modelos basados en LSTM, se consideró una reconstrucción alternativa de la fuerza interna normalizada a partir de la derivada temporal de la velocidad predicha:

$$\hat{\mathbf{g}} = -(\mathbf{a}_g + \dot{\hat{\mathbf{v}}}),$$

debido probablemente a que la salida directa de fuerza en PhyLSTM-2 y PhyLSTM-3 mostró un desempeño deficiente por falta de supervisión de datos sobre esta variable durante el entrenamiento.

#### Comparación global de métricas.

Para evaluar el desempeño de los modelos, resulta fundamental analizar tanto las tendencias generales como los valores numéricos exactos, una práctica estándar para garantizar la reproducibilidad y permitir un *benchmarking* riguroso. En este contexto, la Tabla 3.1 resume las métricas promedio y su dispersión (media  $\pm$  desviación estándar) obtenidas para cada modelo y variable de salida. Cabe destacar que la distribución visual de estos errores se analizará en detalle en la Figura 3.10 de la sección posterior.

En términos globales, las arquitecturas basadas en LSTM muestran los menores valores de L2RE en desplazamiento y velocidad, con una reducción promedio cercana al 75 % respecto de las variantes convolucionales. Este comportamiento evidencia la mayor capacidad de las redes recurrentes para capturar la dinámica temporal del oscilador no histerético y preservar la coherencia entre estados consecutivos. En contraste, para la fuerza interna normalizada, las PhyCNN mantienen un desempeño más estable, mientras que las PhyLSTM exhiben una degradación significativa en la predicción directa de dicha variable. No obstante, la reconstrucción de la fuerza a partir de las derivadas de la velocidad permite recuperar parcialmente la consistencia física de las PhyLSTM, reduciendo el error global en más de un 40 % respecto de su estimación directa.

Tabla 3.1: Métricas de desempeño promedio (media  $\pm$  desviación estándar) sobre el conjunto de prueba para el oscilador de Duffing. La métrica principal es L2RE, complementada por NRMSE y NMAE.

Modelo	L2RE	NRMSE	NMAE
<i>Desplazamiento</i>			
PhyCNN–MO	0.529 $\pm$ 0.189	0.063 $\pm$ 0.029	0.037 $\pm$ 0.018
PhyCNN–SO	0.552 $\pm$ 0.283	0.066 $\pm$ 0.041	0.037 $\pm$ 0.025
PhyLSTM–2	<b>0.108 <math>\pm</math> 0.124</b>	<b>0.013 <math>\pm</math> 0.015</b>	<b>0.005 <math>\pm</math> 0.006</b>
PhyLSTM–3	0.142 $\pm$ 0.150	0.017 $\pm$ 0.019	0.007 $\pm$ 0.008
<i>Velocidad</i>			
PhyCNN–MO	0.507 $\pm$ 0.162	0.055 $\pm$ 0.023	0.032 $\pm$ 0.015
PhyCNN–SO	0.466 $\pm$ 0.210	0.050 $\pm$ 0.027	0.028 $\pm$ 0.017
PhyLSTM–2	<b>0.104 <math>\pm</math> 0.129</b>	<b>0.011 <math>\pm</math> 0.015</b>	<b>0.004 <math>\pm</math> 0.006</b>
PhyLSTM–3	0.142 $\pm$ 0.161	0.016 $\pm$ 0.019	0.006 $\pm$ 0.008
<i>Fuerza interna normalizada</i>			
PhyCNN–MO	<b>0.516 <math>\pm</math> 0.159</b>	<b>0.051 <math>\pm</math> 0.021</b>	<b>0.028 <math>\pm</math> 0.014</b>
PhyCNN–SO (reconstruida)	0.530 $\pm$ 0.260	0.052 $\pm$ 0.034	0.029 $\pm$ 0.021
PhyLSTM–2 (predicha)	1.007 $\pm$ 0.015	0.099 $\pm$ 0.024	0.060 $\pm$ 0.023
PhyLSTM–2 (reconstruida)	0.587 $\pm$ 0.733	0.062 $\pm$ 0.101	0.029 $\pm$ 0.051
PhyLSTM–3 (predicha)	0.981 $\pm$ 0.005	0.097 $\pm$ 0.023	0.056 $\pm$ 0.022
PhyLSTM–3 (reconstruida)	0.618 $\pm$ 0.729	0.065 $\pm$ 0.100	0.030 $\pm$ 0.051
<i>Fuerza–Desplazamiento (L2RE–2D)</i>			
PhyCNN–MO	0.523 $\pm$ 0.175	0.057 $\pm$ 0.025	0.034 $\pm$ 0.017
PhyCNN–SO	0.545 $\pm$ 0.271	0.060 $\pm$ 0.037	0.035 $\pm$ 0.025
PhyLSTM–2	<b>0.413 <math>\pm</math> 0.516</b>	<b>0.047 <math>\pm</math> 0.071</b>	<b>0.022 <math>\pm</math> 0.036</b>
PhyLSTM–3	0.443 $\pm$ 0.513	0.050 $\pm$ 0.071	0.023 $\pm$ 0.036

En particular, PhyLSTM–2 alcanza los menores errores tanto en desplazamiento como en velocidad (L2RE  $\approx$  0.11), mientras que la reconstrucción de la fuerza interna en las arquitecturas recurrentes reduce el error promedio desde  $\approx$  1.0 hasta  $\approx$  0.6, equiparándolo al rango observado en las variantes convolucionales.

### Distribución de errores.

La Figura 3.10 muestra la distribución de L2RE para desplazamiento, velocidad y fuerza interna normalizada; en los modelos PhyLSTM se presenta además la alternativa reconstruida de la fuerza (a partir de  $\hat{\mathbf{v}}$ ). Esta reconstrucción se utiliza porque las predicciones directas de la fuerza presentaron errores excesivos y menor coherencia física.

En cada diagrama, la línea horizontal continua dentro de la caja representa la mediana, mientras que el rombo con líneas segmentadas indica la media. Los límites inferior y superior de la caja corresponden a los

percentiles 25 y 75, respectivamente; los bigotes cubren hasta 1.5 veces el rango intercuartílico y los puntos fuera de los bigotes son valores atípicos.

El eje vertical se limitó a un máximo de 1.25 para mejorar la legibilidad, aunque se identificaron valores atípicos extremos con L2RE cercanos a 3 e incluso 5, especialmente en las reconstrucciones de la fuerza de PhyLSTM. A pesar de estos casos, la mayor parte de las distribuciones se concentra por debajo de 0.6, con dispersiones más amplias en las arquitecturas convolucionales. Las PhyLSTM presentan medianas y medias consistentemente menores en desplazamiento y velocidad, reflejando menor dispersión y mayor estabilidad del error. En la fuerza interna normalizada, las PhyCNN mantienen una distribución más compacta, mientras que las PhyLSTM exhiben mayor variabilidad entre percentiles; sin embargo, alcanzan algunos de los valores mínimos más bajos (colas inferiores  $<0.2$ ); el cuarto grupo "Disp-Force" corresponde al error idimensional  $L2RE_{2D}$  sobre la trayectoria fuerza-desplazamiento, no directamente comparable en magnitud con el L2RE escalar de las otras variables, y se interpreta solo como tendencia por arquitectura: aquí PhyLSTM-2 presenta la menor mediana y un IQR reducido, PhyLSTM-3 es competitiva aunque con algunos atípicos altos, y las PhyCNN muestran medianas mayores y bigotes más extensos.

En conjunto, el desempeño de PhyLSTM es estable en la mayoría de los registros, con alta precisión en los mejores casos y cierta sensibilidad localizada a combinaciones particulares de entrada.

### **Evolución de la función de pérdida.**

La Figura 3.11 muestra la evolución de la función de pérdida durante el entrenamiento de los cuatro modelos, tanto para entrenamiento (líneas sólidas) como para validación (líneas discontinuas). El eje vertical se representa en escala logarítmica.

En todos los modelos se observa un descenso rápido de la pérdida en las primeras épocas, seguido de una reducción más gradual con oscilaciones locales. Las arquitecturas basadas en LSTM alcanzan valores finales del orden de  $10^{-3}$  a  $10^{-4}$ , inferiores a los de PhyCNN (cerca de  $10^{-2}$ ). En las variantes convolucionales se aprecia un estancamiento temprano, especialmente en validación, mientras que las PhyLSTM mantienen una reducción sostenida y sin señales de sobreajuste.

Cabe destacar que los conjuntos de datos son deliberadamente pequeños (ocho registros para entrenamiento y dos para validación), por lo que la pérdida de validación es una estimación de alta varianza. Aun bajo este régimen de datos escasos, las tendencias son consistentes: en PhyLSTM la validación desciende en paralelo al entrenamiento y no presenta repuntes, lo que sugiere buena generalización; en PhyCNN, en cambio, la validación se estabiliza pronto y queda por encima de la de entrenamiento, una brecha de generalización que difícilmente puede atribuirse solo al tamaño muestral y que es coherente con una capacidad limitada para capturar dependencias temporales.

### **Comparación temporal de las respuestas.**

La Figura 3.12 presenta las respuestas temporales para los registros de mejor y peor desempeño (según L2RE promedio de los modelos). En los mejores casos, todas las arquitecturas reproducen adecuadamente fase y amplitud, con una concordancia casi exacta en PhyLSTM-2 y PhyLSTM-3 para desplazamiento y

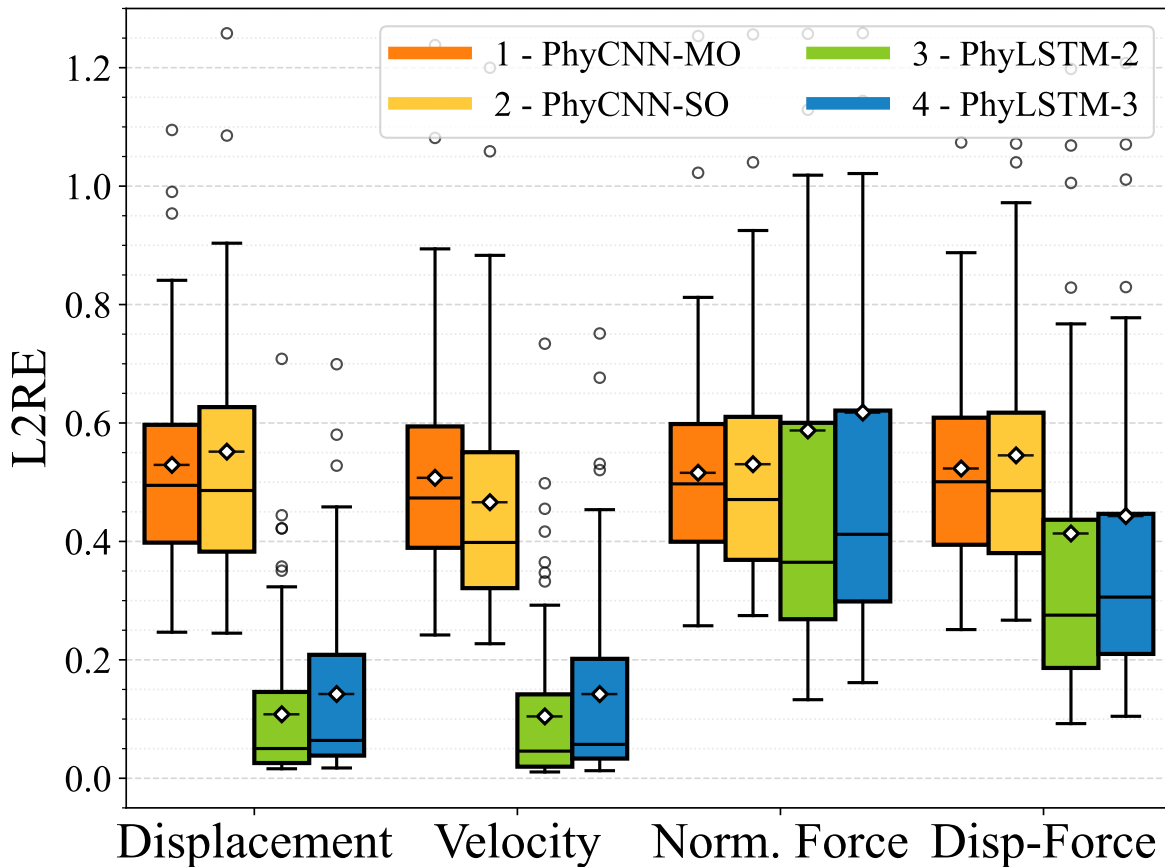


Figura 3.10: Distribución de L2RE para las tres variables de salida en el oscilador de Duffing (conjunto de prueba), el cuarto grupo corresponde al error bidimensional  $L2RE_{2D}$  en el plano fuerza–desplazamiento. El eje vertical se recorta en 1.25 para mejorar la legibilidad.

velocidad. Las PhyCNN exhiben pequeñas desviaciones locales, sobreestimación leve de amplitud y mayor contenido de alta frecuencia, estos mismos patrones se ven reproducidos por las PhyLSTM para la predicción de fuerza normalizada.

En los peores casos, las PhyCNN presentan errores acumulados de fase y mayor dispersión durante intervalos de alta energía. Las PhyLSTM mantienen la forma general en desplazamiento y velocidad, pero con errores de fase locales y sobre/ subestimación de picos. En fuerza, la predicción directa de LSTM es deficiente; la reconstrucción mejora la coherencia pero puede amplificar ruido de alta frecuencia al derivar señales discretas.

### Análisis espectral.

La Figura 3.13 muestra las PSD de desplazamiento, velocidad y fuerza interna normalizada (en PhyLSTM calculada por reconstrucción). En la banda dominante asociada a la frecuencia natural del sistema, las PhyLSTM siguen con precisión la posición y amplitud del pico y su decaimiento, mientras que las PhyCNN tienden a sobreestimar la energía en el rango alto de frecuencias ( $\geq 6$  Hz, con Nyquist en 10 Hz). En la fuerza

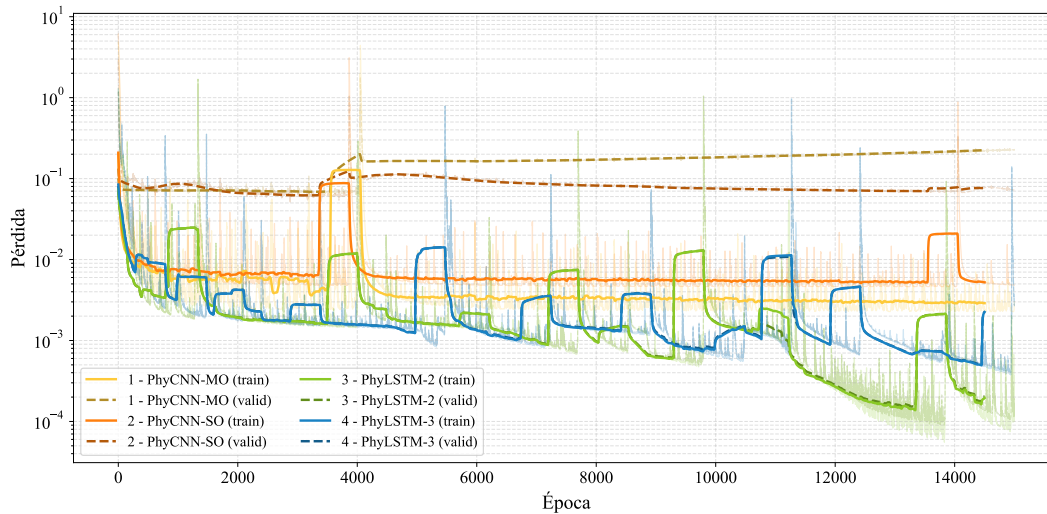


Figura 3.11: Evolución de la pérdida durante entrenamiento y validación (suavizada con ventana móvil de 500 épocas). Eje vertical en escala logarítmica.

reconstruida, las PhyLSTM preservan el máximo espectral, aunque con un leve exceso en la cola; por su parte, PhyCNN–SO ajusta mejor el decaimiento de alta frecuencia.

Dado que las señales no son estrictamente estacionarias, las PSD se presentan como referencia global y se complementan con espectrogramas (STFT) que revelan la evolución tiempo–frecuencia. En ambos casos se utilizó ventana Hann; para las PSD se emplearon segmentos de 200 muestras con 75 % de solape, mientras que para los espectrogramas se usaron segmentos de 160 muestras con 80 % de solape; adicionalmente, se incluye un mapa de error PSD definido como  $10 \log_{10} S_{\hat{x}} - 10 \log_{10} S_x$  (dB).

La Figura 3.14 compara el espectrograma de desplazamiento real con el de PhyCNN–SO (representativo de los modelos PhyCNN) y su error PSD. Se observa un exceso persistente de energía en la banda alta (aprox. 6–10 Hz) durante prácticamente toda la duración, coherente con las colas de la PSD; en torno a 0–2 Hz el error es cercano a 0 dB, indicando un seguimiento adecuado de la banda dominante. Este comportamiento sugiere que la operación convolucional introduce componentes de alta frecuencia que, aunque de baja amplitud en el dominio temporal, elevan el piso espectral. En casos desfavorables, estas componentes pueden amplificarse y afectar la amplitud aparente de la respuesta, como en el peor desempeño de desplazamiento mostrado en la Fig. 3.12.

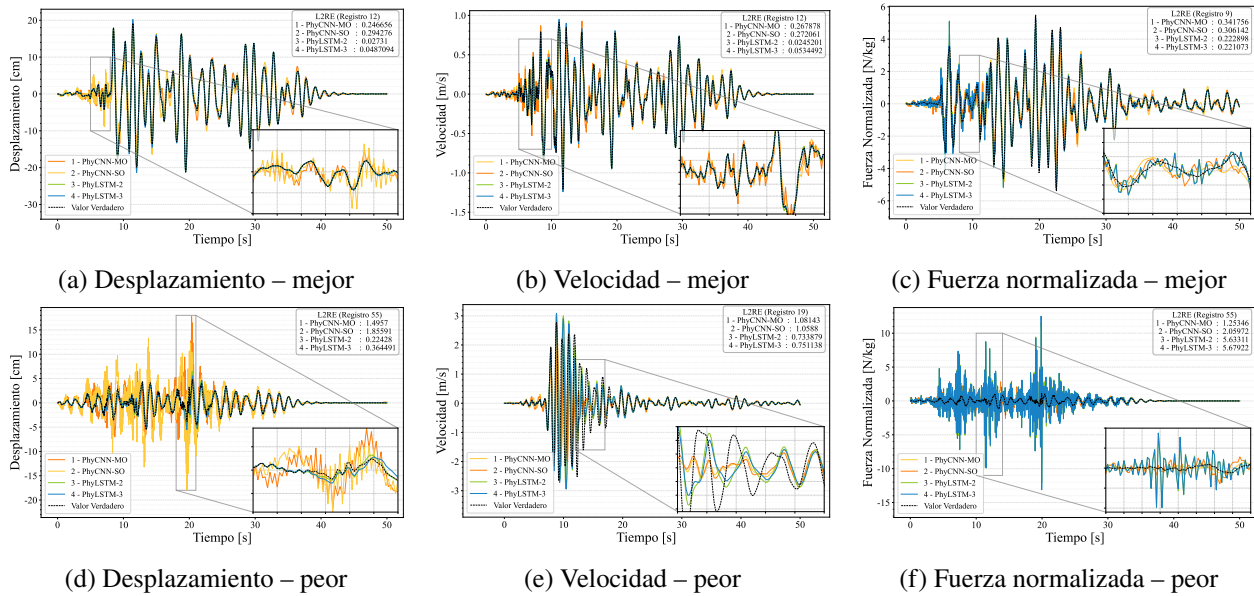


Figura 3.12: Comparación temporal de las predicciones para los registros de mejor (fila superior) y peor (fila inferior) desempeño. La referencia se muestra con línea discontinua.

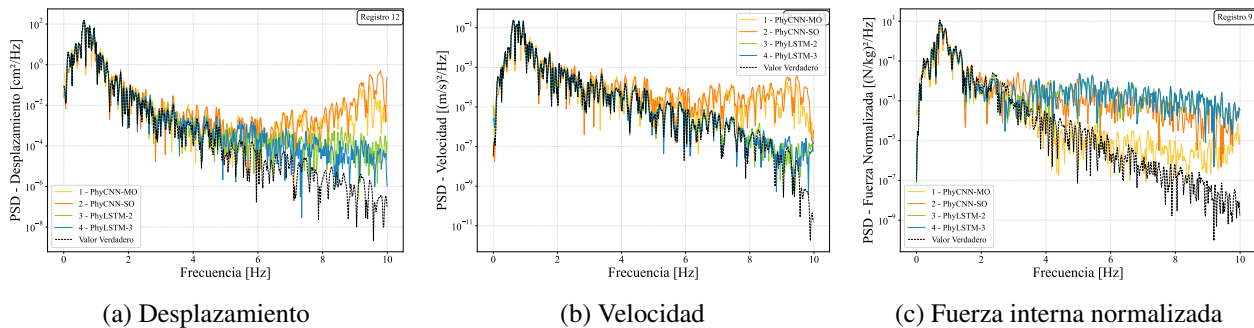


Figura 3.13: PSD de desplazamiento, velocidad y fuerza interna normalizada para los registros de mejor desempeño (eje vertical logarítmico).

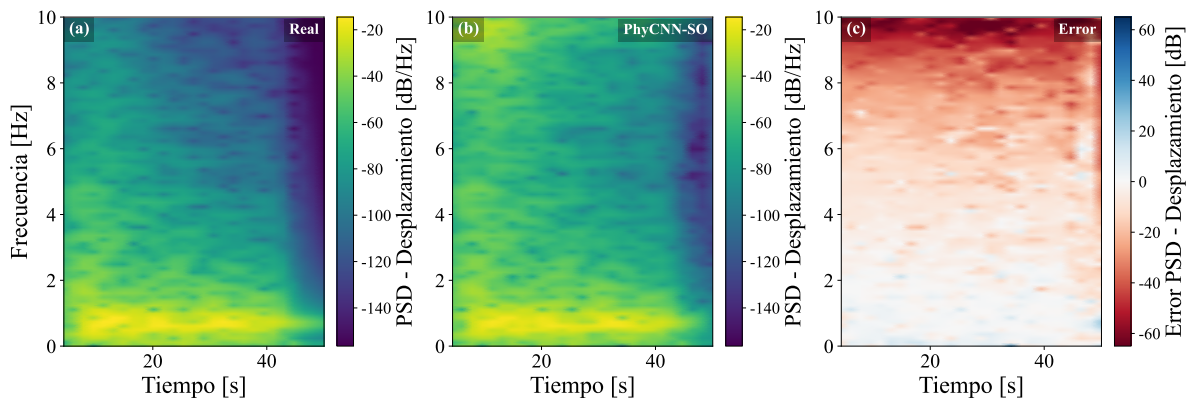


Figura 3.14: Espectrograma de desplazamiento para el registro con mejor desempeño (registro 12): (a) real, (b) PhyCNN-SO, (c) error PSD (dB).

La Figura 3.15 muestra el espectrograma de **fuerza interna normalizada** (reconstruida) para PhyLSTM-2 (representativo de los modelos PhyLSTM). El error PSD se concentra en frecuencias medias–altas (aprox. 3–10 Hz) a lo largo de toda la secuencia, lo que es coherente con (i) la derivación numérica implicada en la reconstrucción a partir de  $\hat{v}$ , que amplifica componentes de alta frecuencia, y (ii) la ausencia de supervisión directa de la fuerza durante el entrenamiento, que puede desanclar su suavidad. Aunque estos aportes son pequeños en el dominio temporal, explican el sesgo en las colas de las PSD y parte del error en  $g$ . En casos desfavorables, pueden también amplificar la amplitud aparente de la respuesta, como en el peor desempeño de fuerza mostrado en la Fig. 3.12.

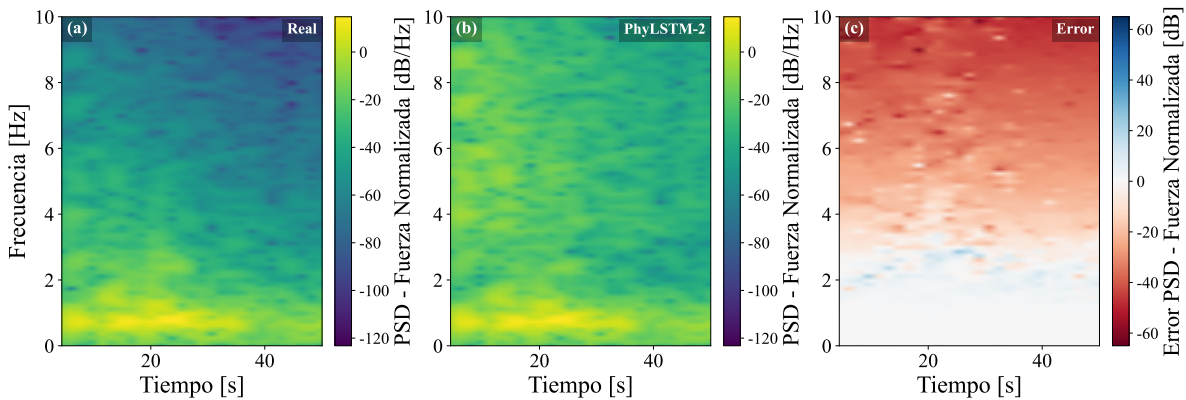


Figura 3.15: Espectrograma de fuerza interna normalizada para el registro con mejor desempeño (registro 9): (a) real, (b) PhyLSTM-2 (reconstruida), (c) *error PSD* (dB).

En síntesis, los espectrogramas confirman que el **exceso de alta frecuencia** es un fenómeno persistente: en PhyCNN aparece principalmente en el desplazamiento (6–10 Hz), mientras que en PhyLSTM se manifiesta en la fuerza reconstruida (3–10 Hz). Esto coincide con las colas de las PSD y con las diferencias observadas en las métricas: buen ajuste en la banda dominante y pequeñas discrepancias acumuladas en bandas altas que elevan los errores.

### Relación fuerza–desplazamiento.

La Figura 3.16 muestra las curvas fuerza–desplazamiento para el registro de mejor desempeño (según L2RE-2D promedio de los modelos). Todas las arquitecturas reproducen la morfología global, preservando la simetría respecto del equilibrio y la pendiente efectiva en el tramo central. Las PhyLSTM-2 y PhyLSTM-3 siguen con mayor fidelidad la curvatura de los lazos y la amplitud en los extremos, cerrando la trayectoria con menor error. En contraste, las PhyCNN describen lazos más cortos, presentando una ligera subestimación del desplazamiento, especialmente en los máximos y un mayor contenido de alta frecuencia que dentella la trayectoria, haciéndola más ruidosa y con una rigidización aparente en los extremos, aunque sin formación de lazos histeréticos.

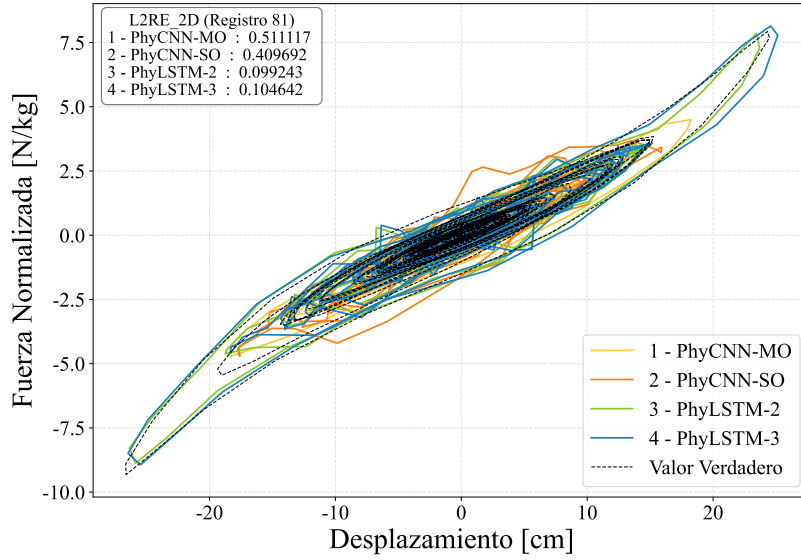


Figura 3.16: Curvas fuerza–desplazamiento correspondientes al registro de mejor desempeño (índice 81). La trayectoria refleja una relación no-lineal suave sin histéresis. El error asociado es **L2RE–2D**.

### Síntesis del Escenario I.

En conjunto, los resultados del escenario de Duffing confirman que las arquitecturas *guided-informed* basadas en LSTM ofrecen un desempeño superior en la predicción de desplazamientos y velocidades, mientras que las variantes convolucionales resultan más estables para la estimación directa de fuerzas internas. La estrategia de reconstrucción propuesta para PhyLSTM permite equilibrar ambas capacidades, recuperando la coherencia física del modelo sin requerir ajustes adicionales en el entrenamiento. La inclusión de espectrogramas corrobora que los sesgos observados en las colas de las PSD se asocian a inyección de alta frecuencia persistente a lo largo de la señal.

### 3.5. Escenario II: Bouc–Wen SDOF simplificado bajo ruido blanco filtrado

El segundo conjunto de datos corresponde a la formulación Bouc–Wen empleada por Zhang et al. (2020b), en la cual un sistema SDOF presenta comportamiento histerético con disipación cíclica de energía. La ecuación de movimiento se define como:

$$m \ddot{\mathbf{u}} + c \dot{\mathbf{u}} + \alpha k_0 \mathbf{u} + (1 - \alpha) k_0 u_y \mathbf{z} = -m \mathbf{a}_g, \quad (3.6)$$

donde la variable interna  $\mathbf{z}$  describe la evolución del comportamiento inelástico mediante la siguiente relación diferencial:

$$\dot{\mathbf{z}} = u_y^{-1} \left( A \dot{\mathbf{u}} - \beta |\dot{\mathbf{u}}| |\mathbf{z}|^{n-1} \mathbf{z} - \gamma \dot{\mathbf{u}} |\mathbf{z}|^n \right). \quad (3.7)$$

Las propiedades adoptadas para este sistema son: masa  $m = 500$  kg, amortiguamiento  $c = 0.35$  kN·s/m y rigidez inicial  $k_0 = 25$  kN/m. Los parámetros del modelo histerético se fijaron en  $A = 1$ ,  $\beta = 2$ ,  $\gamma = 2$  y  $n = 3$ ,

con frecuencia natural  $f_n = 1.13$  Hz y parámetro de transición  $\alpha = 0.5$  (en esta formulación,  $u_y$  se emplea únicamente como escala de referencia para la variable interna).

Las excitaciones corresponden a señales de ruido blanco limitado en banda (*band-limited white noise*, BLWN) de distintas intensidades, con una duración de 30 s y frecuencia de muestreo de 50 Hz, resultando en  $T = 1501$  instantes discretos. En total se generaron 100 realizaciones independientes mediante simulación temporal no-lineal, de las cuales ocho se destinaron al entrenamiento, dos a validación y las 90 restantes al conjunto de prueba. Para el ajuste de las restricciones físicas se utilizó un conjunto de colocación conformado por los 10 registros de entrenamiento y validación junto con 50 registros adicionales seleccionados del conjunto de prueba, los cuales solo fueron considerados para el ajuste de la restricción física del modelo.

Este escenario introduce un grado moderado de no-linealidad e histéresis, permitiendo evaluar la capacidad de las arquitecturas para reproducir la disipación energética y los bucles de histéresis característicos de sistemas con memoria material.

### 3.5.1. Desempeño de PhyCNN y Multi-LSTM en Escenario II

La Tabla 3.2 resume las métricas promedio obtenidas para el conjunto de prueba del escenario Bouc–Wen ante ruido blanco, expresadas como media  $\pm$  desviación estándar. En desplazamiento y velocidad, las arquitecturas PhyLSTM muestran los menores valores de error, con reducciones del orden del 18 % y del 70–80 % respectivamente respecto de las PhyCNN. En particular, PhyLSTM–2 presenta el mejor desempeño en desplazamiento (L2RE =  $0.779 \pm 0.155$ ), mientras que ambas variantes recurrentes alcanzan errores prácticamente idénticos en velocidad (L2RE  $\approx 0.045$ ).

Para la fuerza interna normalizada, la diferencia entre predicción directa y reconstrucción resulta significativa. Las predicciones directas de las PhyLSTM arrojan errores cercanos a la unidad, lo que indica un desacoplamiento entre la fuerza estimada y las excitaciones de entrada, probablemente debido a la falta de ajuste a los datos durante el entrenamiento. Sin embargo, la reconstrucción de la fuerza a partir de las derivadas de la velocidad reduce el error más de un orden de magnitud, alcanzando valores de L2RE  $\approx 0.06$ – $0.07$ , lo que representa una mejora notable frente a cualquier otra formulación. En las redes convolucionales, la versión PhyCNN–SO obtiene un desempeño ligeramente mejor que la PhyCNN–MO, con una dispersión menor de los errores.

Finalmente, en la relación fuerza–desplazamiento, las PhyLSTM presentan los menores errores promedio y una menor dispersión, evidenciando una mejor representación de los lazos de histéresis.

Tabla 3.2: Métricas de desempeño promedio (media  $\pm$  desviación estándar) sobre el conjunto de prueba del sistema Bouc–Wen ante ruido blanco. La métrica principal es L2RE, complementada por NRMSE y NMAE. La fila *Fuerza–Desplazamiento* reporta L2RE–2D.

Modelo	L2RE	NRMSE	NMAE
<i>Desplazamiento</i>			
PhyCNN–MO	1.289 $\pm$ 0.361	0.316 $\pm$ 0.098	0.260 $\pm$ 0.089
PhyCNN–SO	0.781 $\pm$ 0.128	0.205 $\pm$ 0.085	0.170 $\pm$ 0.078
PhyLSTM–2	<b>0.779 <math>\pm</math> 0.155</b>	<b>0.200 <math>\pm</math> 0.078</b>	<b>0.161 <math>\pm</math> 0.070</b>
PhyLSTM–3	0.908 $\pm$ 0.244	0.229 $\pm$ 0.086	0.188 $\pm$ 0.077
<i>Velocidad</i>			
PhyCNN–MO	0.387 $\pm$ 0.061	0.074 $\pm$ 0.012	0.059 $\pm$ 0.009
PhyCNN–SO (derivada)	0.303 $\pm$ 0.045	0.058 $\pm$ 0.008	0.046 $\pm$ 0.007
PhyLSTM–2	<b>0.045 <math>\pm</math> 0.010</b>	<b>0.009 <math>\pm</math> 0.002</b>	<b>0.006 <math>\pm</math> 0.001</b>
PhyLSTM–3	0.045 $\pm$ 0.010	0.009 $\pm$ 0.002	0.006 $\pm$ 0.001
<i>Fuerza interna normalizada</i>			
PhyCNN–MO	0.347 $\pm$ 0.053	0.090 $\pm$ 0.013	0.072 $\pm$ 0.010
PhyCNN–SO (reconstruida)	0.299 $\pm$ 0.046	0.078 $\pm$ 0.011	0.062 $\pm$ 0.009
PhyLSTM–2 (predicha)	1.000 $\pm$ 0.003	0.260 $\pm$ 0.012	0.220 $\pm$ 0.013
PhyLSTM–2 (reconstruida)	0.069 $\pm$ 0.008	0.018 $\pm$ 0.002	0.014 $\pm$ 0.001
PhyLSTM–3 (predicha)	1.002 $\pm$ 0.002	0.261 $\pm$ 0.012	0.220 $\pm$ 0.013
PhyLSTM–3 (reconstruida)	<b>0.060 <math>\pm</math> 0.008</b>	<b>0.016 <math>\pm</math> 0.002</b>	<b>0.012 <math>\pm</math> 0.001</b>
<i>Fuerza–Desplazamiento (Histéresis) (L2RE–2D)</i>			
PhyCNN–MO	0.819 $\pm$ 0.202	0.208 $\pm$ 0.053	0.179 $\pm$ 0.047
PhyCNN–SO	0.534 $\pm$ 0.094	0.139 $\pm$ 0.038	0.122 $\pm$ 0.036
PhyLSTM–2	<b>0.472 <math>\pm</math> 0.112</b>	<b>0.122 <math>\pm</math> 0.038</b>	<b>0.099 <math>\pm</math> 0.035</b>
PhyLSTM–3	0.547 $\pm$ 0.154	0.141 $\pm$ 0.045	0.116 $\pm$ 0.040

### Distribución de errores.

La Figura 3.17 presenta la distribución del error relativo cuadrático medio (L2RE) para desplazamiento, velocidad y fuerza interna normalizada del sistema Bouc–Wen bajo ruido blanco limitado en banda, además se muestra, la distribución del índice bidimensional L2RE–2D asociado a la relación fuerza–desplazamiento. Aun cuando las métricas difieren en definición, aquí se contrastan de forma directa para evaluar de manera complementaria la precisión en variables individuales y la consistencia geométrica del ciclo histerético.

En desplazamiento, los errores promedio son mayores que en las demás variables, coherente con la dependencia de historia del modelo Bouc–Wen: el estado no depende solo del instante, sino de la trayectoria previa, con plastificación acumulada y cambios del eje de oscilación. En este contexto, PhyLSTM–2 concentra las menores medianas de L2RE y baja dispersión; PhyCNN–SO muestra una distribución más compacta, aunque

sin alcanzar los mínimos de PhyLSTM–2; y PhyCNN–MO presenta medianas altas y colas largas, evidenciando limitaciones para capturar la evolución con memoria. Esta diferencia entre las variantes convolucionales es consistente con su diseño: PhyCNN–SO está especializada en predecir únicamente el desplazamiento y derivar el resto de variables a partir de él, mientras que PhyCNN–MO debe aprender múltiples salidas simultáneamente, lo que diluye su capacidad para afinar la predicción de  $u(t)$ .

En velocidad y fuerza interna normalizada, las PhyLSTM exhiben errores claramente menores y distribuciones más compactas (medianas y rangos intercuartílicos reducidos), lo que es consistente con una representación más estable de la dinámica instantánea y del equilibrio inercial. En contraste, las PhyCNN presentan dispersiones mayores (medianas en torno a 0.3–0.4), compatibles con pequeñas incoherencias entre  $u$ ,  $\dot{u}$  y  $g$  a lo largo de la secuencia. En el panel fuerza–desplazamiento (L2RE–2D), PhyLSTM–2 mantiene la mediana más baja y colas controladas, mientras que PhyCNN–MO concentra los errores más altos y la mayor variabilidad, consecuencia de su dificultad para predecir el desplazamiento. Estos patrones refuerzan que la memoria explícita de las LSTM favorece la preservación de relaciones derivativas y acopladas, aun cuando el seguimiento de  $u(t)$  sea más desafiante.

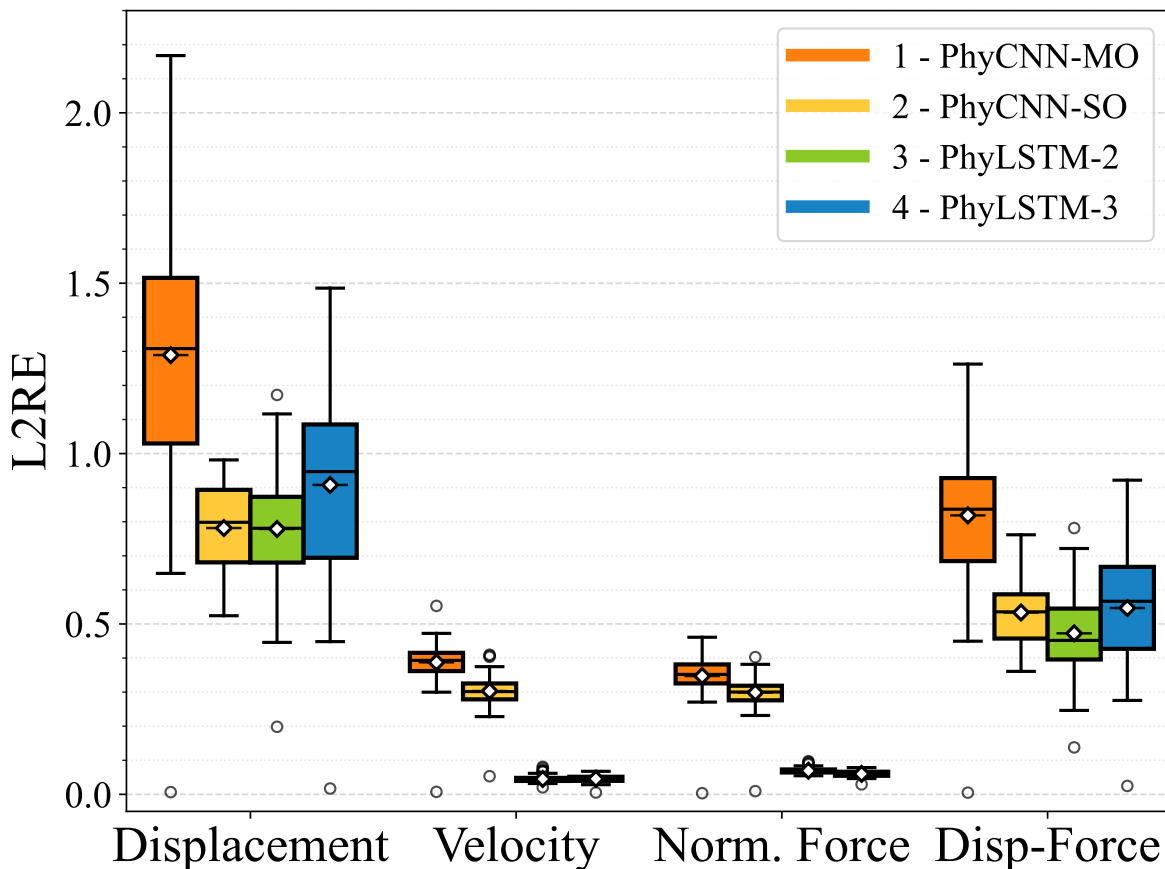


Figura 3.17: Distribución del error relativo cuadrático medio (L2RE) para desplazamiento, velocidad y fuerza interna normalizada en el conjunto de prueba del sistema Bouc–Wen ante ruido blanco filtrado. El eje vertical se recorta en 2.3 para mejorar la legibilidad.

### Evolución de la función de pérdida.

La Figura 3.18 muestra la evolución de la función de pérdida durante el entrenamiento y la validación de las cuatro arquitecturas. En una primera fase todas las redes descienden con rapidez y, luego, convergen más lentamente. En entrenamiento, las PhyLSTM alcanzan valores del orden de  $10^{-4}$ – $10^{-3}$ , mientras que las PhyCNN se estabilizan alrededor de  $10^{-2}$ . En validación, tanto PhyLSTM–2 como PhyLSTM–3 se estabilizan cerca de  $10^{-2}$ , cercano a las PhyCNN pero sin continuar la caída observada en entrenamiento. Este comportamiento debe leerse con cautela porque el conjunto de validación es muy reducido (2 registros frente a 8 de entrenamiento) y su composición puede sesgar la métrica: por ejemplo, la presencia de trayectorias con cambios del eje de oscilación, deriva residual o mayor amplitud podría fijar una "meseta" de validación independiente del progreso en entrenamiento. En conjunto, las curvas sugieren que las LSTM optimizan más en entrenamiento, mientras que la generalización queda condicionada tanto por el tamaño de validación como por las características específicas de esos registros. Las PhyCNN, además, exhiben una brecha temprana entre entrenamiento y validación, consistente con un estancamiento prematuro. Las oscilaciones intermedias visibles en todas las curvas pueden deberse a la heterogeneidad del lote o a ajustes del optimizador, aunque no es posible atribuirles una causa única con la evidencia disponible.

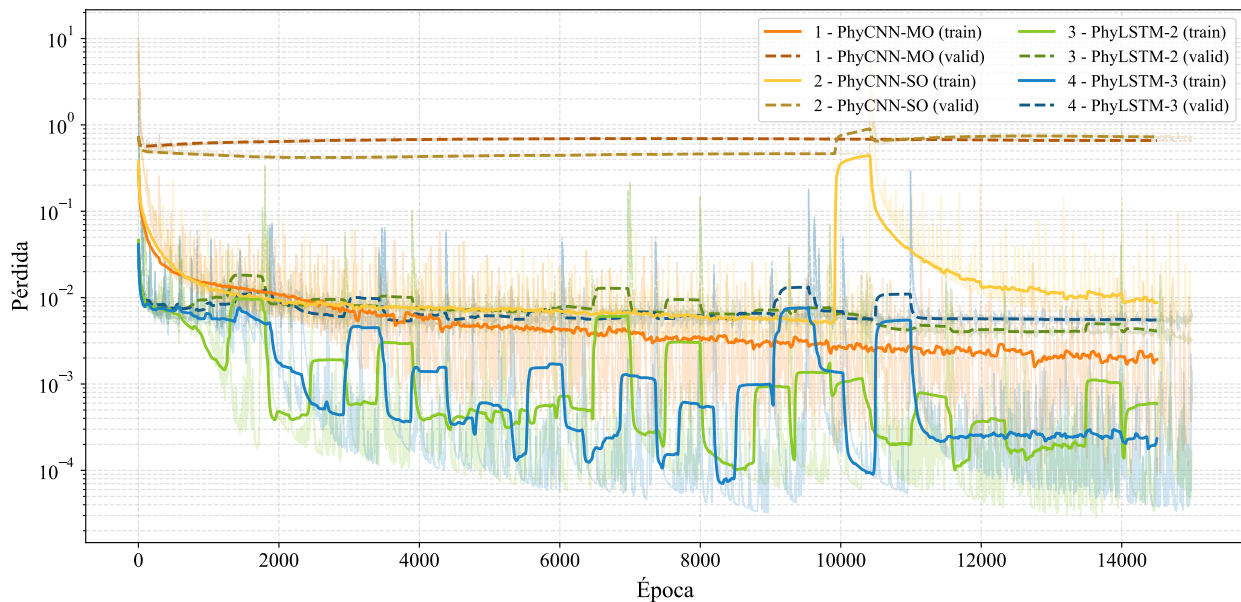


Figura 3.18: Curvas de pérdida suavizadas (ventana móvil de 500 épocas) durante el entrenamiento y validación de los modelos para el sistema Bouc–Wen. El eje vertical se representa en escala logarítmica.

### Análisis temporal de las predicciones.

La Figura 3.19 compara las series temporales simuladas con las predichas por los cuatro modelos para desplazamiento, velocidad y fuerza interna normalizada, en los registros de mejor y peor desempeño (según L2RE promedio de los modelos). En los mejores casos, PhyLSTM–3 y PhyCNN–MO se superponen prácticamente con la referencia en las tres variables; PhyLSTM–2 y PhyCNN–SO ajustan bien velocidad y

fuerza, con diferencias más visibles en desplazamiento (ligera sobreestimación de amplitud en PhyLSTM–2 y subestimación del desplazamiento residual en PhyCNN–SO). En los peores casos, PhyCNN–SO rinde mejor y PhyCNN–MO concentra las mayores derivas acumulativas; las PhyLSTM conservan la forma general y la tendencia del desplazamiento acumulado, con pequeñas discrepancias en amplitud, mostrando un desempeño similar a PhyCNN–SO. En síntesis, desplazamiento es la magnitud más desafiante, por su dependencia de historia, mientras que velocidad y fuerza muestran errores mucho menores y una coherencia dinámica similar entre modelos, favoreciendo a las arquitecturas con memoria explícita.

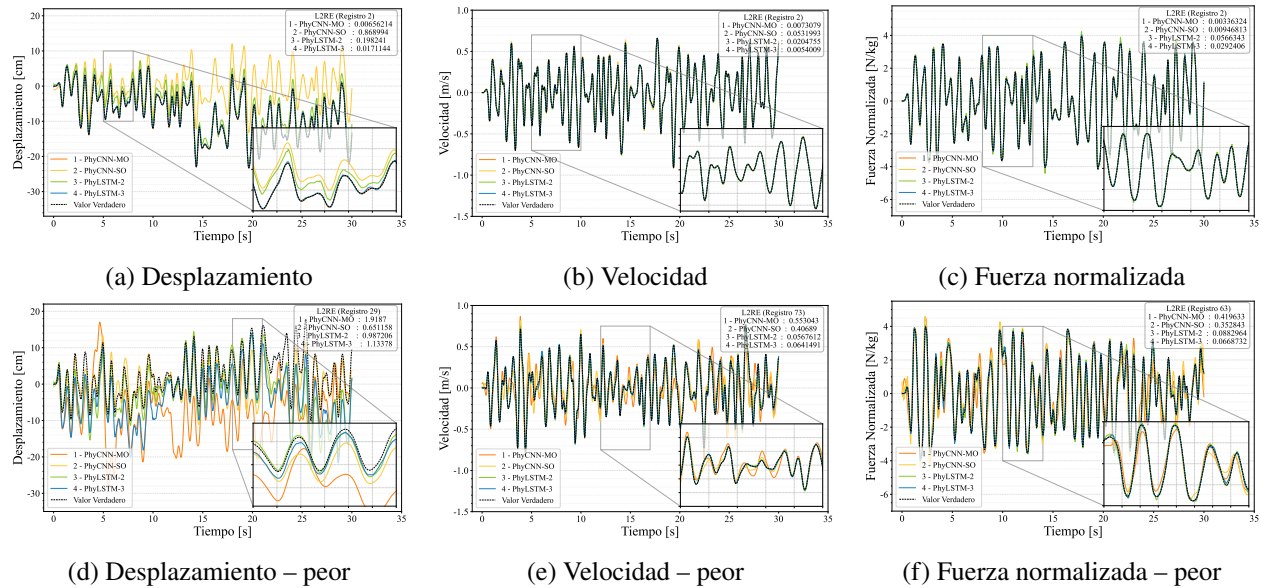


Figura 3.19: Comparación tiempo–historia de desplazamiento, velocidad y fuerza interna normalizada para los registros de mejor (fila superior) y peor (fila inferior) desempeño en el escenario Bouc–Wen ante ruido blanco. Las líneas continuas representan las respuestas de referencia y las líneas de color las predicciones de cada modelo.

### Análisis espectral.

La Figura 3.20 presenta las densidades espectrales de potencia (PSD) para el registro de mejor desempeño. En desplazamiento, todas las arquitecturas reproducen la banda dominante de baja frecuencia; sin embargo, muestran un exceso de energía en frecuencias medias–altas ( $f \gtrsim 5$  Hz), más marcado en las PhyLSTM y, en menor medida, en las PhyCNN, lo que sugiere una ligera rugosidad de alta frecuencia en las trazas temporales. En el extremo opuesto del espectro, la PhyCNN–SO presenta un déficit de energía en la frecuencia más baja respecto de la referencia, consistente con su dificultad para reproducir *desplazamientos residuales* y cambios del eje de oscilación: el contenido cuasi–estático asociado a la deriva se concentra en la vecindad de 0 Hz, de modo que una deriva subestimada se manifiesta como menor PSD en bajas frecuencias.

En velocidad y fuerza interna normalizada las diferencias espectrales se atenúan: las curvas de todos los modelos siguen la localización de las frecuencias dominantes y el decaimiento inicial; persiste, no obstante, un piso ligeramente más alto en altas frecuencias para las variantes recurrentes, mientras que las PhyCNN,

mantienen colas más regulares. En conjunto, los modelos capturan correctamente la dinámica principal, y las discrepancias se concentran fuera de la banda de mayor interés ingenieril.

Dado que las señales no son estrictamente estacionarias, las PSD se presentan como referencia global y se complementan con espectrogramas (STFT) que revelan la evolución tiempo–frecuencia. En ambos casos se utilizó ventana Hann; para las PSD se emplearon segmentos de 200 muestras con 75 % de solape, mientras que para los espectrogramas se usaron segmentos de 160 muestras con 80 % de solape; adicionalmente, se incluye un mapa de error PSD definido como  $10 \log_{10} S_{\hat{x}} - 10 \log_{10} S_x$  (dB).

La Figura 3.21 compara el espectrograma de desplazamiento real con los de PhyCNN–SO y PhyLSTM–2, junto con sus mapas de error PSD (dB). En ambos modelos se aprecia un exceso persistente de energía en bandas altas ( $\geq 10$ –15 Hz) a lo largo del registro, consistente con las colas observadas en la PSD; este exceso es más marcado en PhyLSTM–2, que eleva el nivel de fondo desde frecuencias algo más bajas ( $\approx 5$  Hz). En conjunto, las PSD y los *STFT*, junto con las predicciones tiempo–historia, indican que las discrepancias en altas frecuencias se deben principalmente a efectos de estimación (ventaneo, solape, discretización y derivadas) y al sesgo inductivo de cada arquitectura, más que a una deficiente captura de la dinámica principal del sistema.

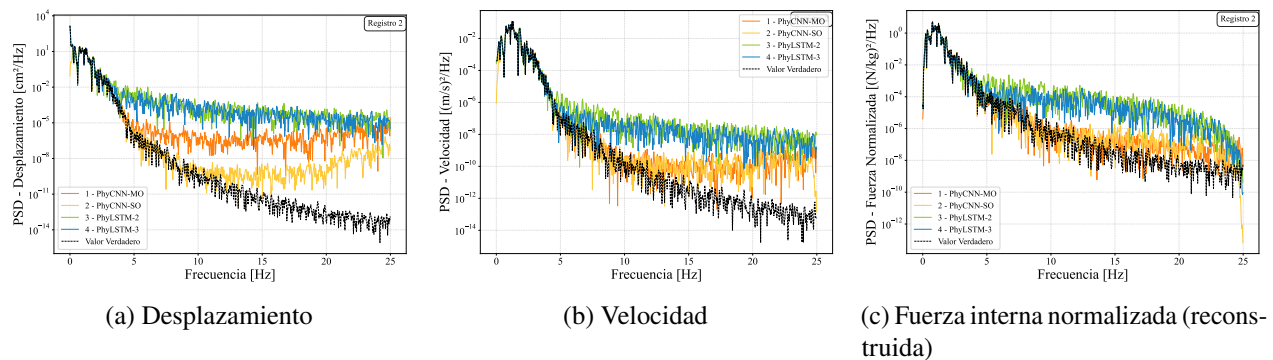


Figura 3.20: Espectros de densidad de potencia (PSD) de desplazamiento, velocidad y fuerza interna normalizada para los registros de mejor desempeño en el sistema Bouc–Wen ante ruido blanco. Las líneas discontinuas son las respuestas de referencia y las curvas de color las predicciones de cada modelo.

### Relación fuerza–desplazamiento (Bouc–Wen).

La Figura 3.22 muestra las curvas de histéresis fuerza–desplazamiento obtenidas para el registro de mejor desempeño (según L2RE-2D promedio de los modelos). Todas las arquitecturas reproducen la forma global del lazo, manteniendo la pendiente inicial y el cierre elástico; las diferencias se concentran en la amplitud y regularidad de los ciclos externos. PhyLSTM–3 y PhyCNN–MO logran lazos muy próximos a la referencia. PhyLSTM–2 sigue la trayectoria con ligera dispersión en ciclos intermedios (pequeños desfases y ruido), conservando la pendiente promedio y el eje de equilibrio, mientras que PhyCNN–SO muestra el mayor desajuste (lazo desplazado, centrado en cero), asociado a subestimación del desplazamiento residual.

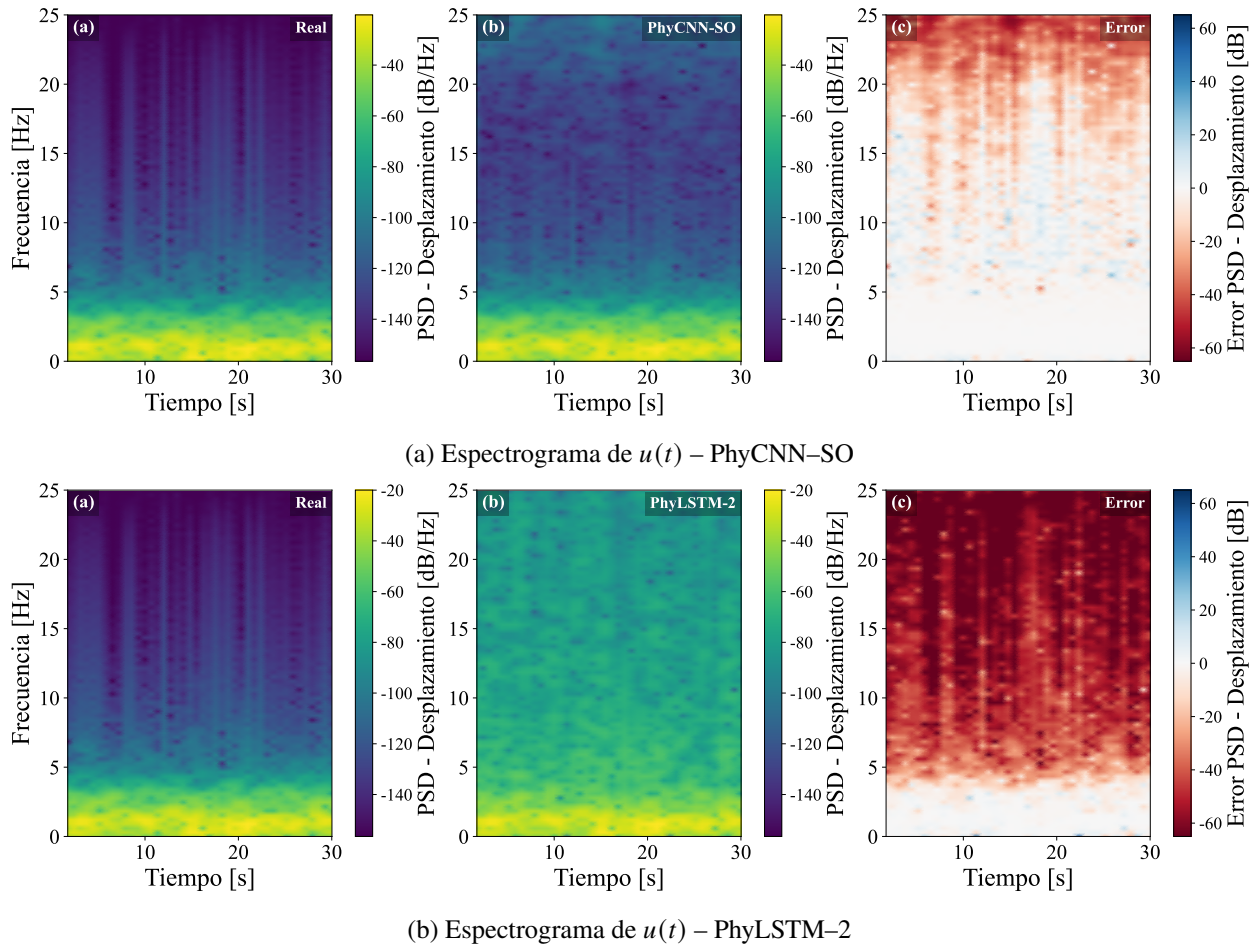


Figura 3.21: Espectrogramas de desplazamiento para el registro con mejor desempeño (registro 2): (a) real, (b) PhyCNN–SO/PhyLSTM–2, (c) *error PSD* (dB).

## Síntesis del Escenario II

El análisis del sistema Bouc–Wen excitado por ruido blanco limitado en banda permite evaluar el desempeño de las arquitecturas guiadas por la física en un entorno controlado pero fuertemente no-lineal, donde las interacciones entre desplazamiento, velocidad y fuerza restauradora están mediadas por efectos de memoria e histéresis. Globalmente, las PhyLSTM alcanzan los menores errores promedio y mayor consistencia estadística que las PhyCNN. En desplazamiento, la variable más compleja por acumulación de historia, PhyLSTM–2 logra menores medianas y dispersión; PhyCNN–SO resulta competitiva pero con limitaciones para reproducir derivas residuales, mientras que PhyCNN–MO presenta los mayores errores. En velocidad y fuerza reconstruida las diferencias entre arquitecturas se estrechan y los errores son muy bajos, principalmente en las arquitecturas PhyLSTM, lo que indica que capturan la dinámica instantánea, diferenciándose en su capacidad de memoria. Finalmente, las curvas de histéresis y las métricas de L2RE-2D refuerzan que las PhyLSTM preservan mejor la geometría y el área de los lazos, validando la pertinencia de incorporar mecanismos explícitos de memoria

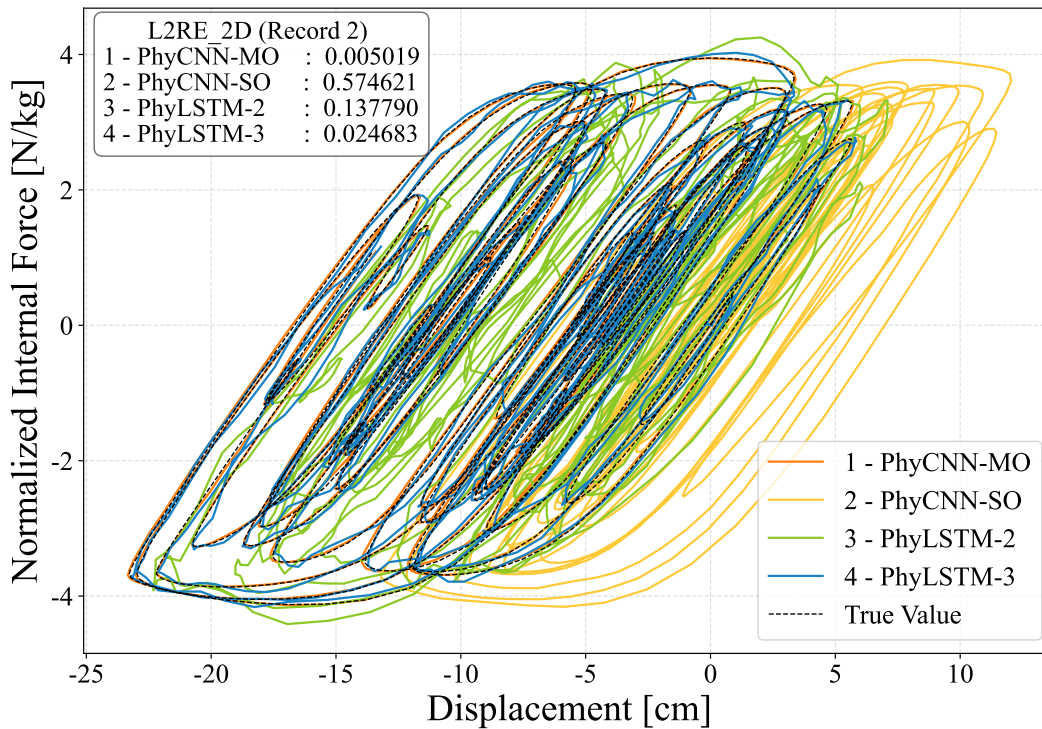


Figura 3.22: Curvas fuerza–desplazamiento correspondientes al registro de mejor desempeño del sistema Bouc–Wen bajo excitación de ruido blanco filtrado.

temporal en arquitecturas guiadas por la física para sistemas no-lineales con acumulación de plastificación y disipación cíclica.

### 3.6. Escenario III: Bouc–Wen SDOF representativo de edificio aislado bajo sismos chilenos de larga duración

El tercer conjunto de datos, desarrollado específicamente para este trabajo, representa la respuesta dinámica de un sistema SDOF que modela el comportamiento de un aislador sísmico tipo *Friction Pendulum System* (FPS) sometido a registros reales de terremotos chilenos. El sistema se caracteriza por una masa total de  $m = 1.988$  ton, período efectivo  $T_{ef} = 2.5$  s y rigidez efectiva  $k_{ef} = 12.56$  kN/mm. El desplazamiento de fluencia se fija en  $u_y = 6$  cm, con rigidez inicial  $k_0 = 71.75$  kN/mm y razón post-fluencia  $\alpha = 0.01$  ( $k_p = 0.72$  kN/mm). Estas propiedades definen un sistema fuertemente no-lineal, con amplias deformaciones laterales y potenciales desplazamientos residuales significativos.

Las excitaciones consisten en acelerogramas reales registrados durante eventos sísmicos de distintas magnitudes en territorio chileno. Los registros se obtuvieron desde el portal *SiberRisk* (Castro et al., 2022) y fueron seleccionados a partir de los siguientes criterios: (i) distancia hipocentral menor a 100 km, (ii) magnitud de momento  $M_w > 6.0$ , y (iii) desplazamiento espectral, para un período de aislación  $T = 2.5$  s y amortiguamiento del 5 %, superior a 3 cm (equivalente a  $u_y/2$ ). El cumplimiento de estas condiciones arrojó

un total de 270 registros, los cuales fueron utilizados como entrada al modelo Bouc–Wen. Cabe destacar que en las simulaciones se emplearon los registros en su forma original, sin procesamiento adicional.

La simulación del sistema se implementó en *Simulink* mediante un bloque en espacio de estados que integra un subsistema encargado de generar la variable de histéresis  $z(t)$  del modelo Bouc–Wen. Este subsistema recibe como entrada la velocidad  $\dot{u}(t)$  y reproduce la evolución temporal de la variable interna histerética. La salida  $z(t)$ , junto con el registro sísmico  $a_g(t)$ , se introducen al bloque principal de espacio de estados, permitiendo simular la respuesta completa del sistema bajo excitaciones sísmicas. La configuración general de este esquema se ilustra en la Figura 3.23.

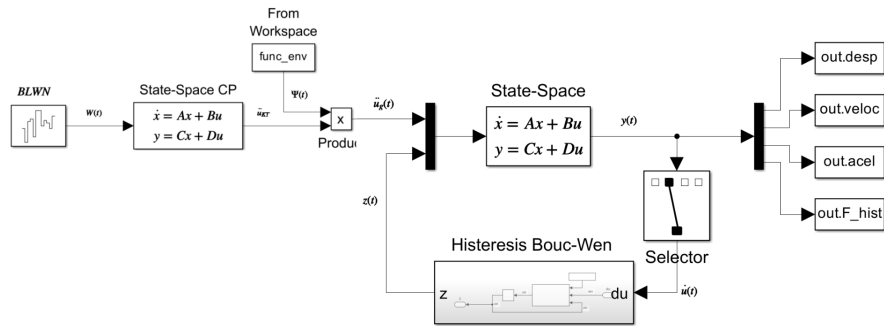


Figura 3.23: Diagrama de bloques del sistema SDOF con modelo Bouc–Wen completo en *Simulink*, mostrando la integración del modelo de histéresis y las entradas sísmicas.

Con el fin de validar la correcta implementación del modelo Bouc–Wen, se realizó una prueba inicial aplicando una señal sinusoidal al subsistema de histéresis. El diagrama correspondiente se presenta en la Figura 3.24, donde se observa la aplicación de la velocidad  $\dot{u}(t)$  y la generación de la variable  $z(t)$  mediante una función programada en *MATLAB*. Dicha función evalúa la ecuación diferencial definida en la Ecuación 3.7, cuya integración en el tiempo produce  $z(t)$ , retroalimentando el sistema en un lazo cerrado.

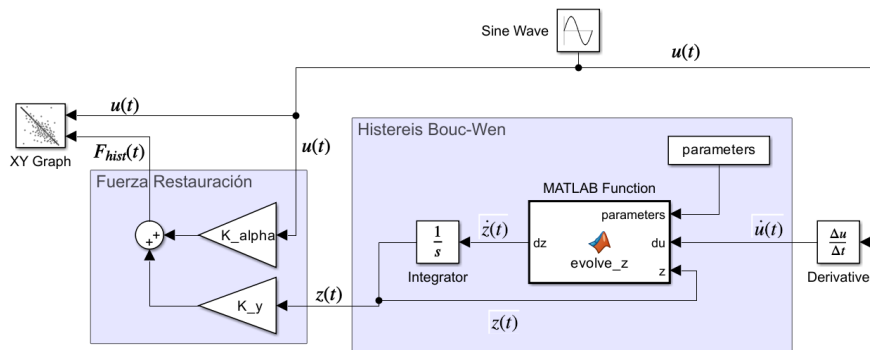


Figura 3.24: Diagrama de bloques de la implementación inicial del modelo Bouc–Wen en *Simulink* con entrada sinusoidal.

El sistema completo, representado en la Figura 3.23, permite procesar los registros sísmicos y obtener una simulación detallada y realista del comportamiento dinámico del sistema bajo excitaciones reales. Las simulaciones se ejecutaron con un paso de integración reducido,  $\Delta t = 0.001$  s, ejecutándose todas las

simulaciones en aproximadamente 10 min. Posteriormente, tanto las excitaciones como las respuestas fueron submuestreadas a un paso temporal de  $\Delta t = 0.05$  s. Durante este proceso se verificó la preservación del contenido espectral mediante la comparación de densidades espectrales de potencia, asegurando que el submuestreo no introdujera pérdida relevante de información dinámica. De esta manera, se conformó una base de datos compuesta por 270 registros, cada uno con 6793 instantes discretos. La Figura 3.25 muestra la distribución geográfica de los eventos y estaciones asociadas a los registros empleados. La selección de registros no consideró una caracterización de sitio ni un ajuste espectral a un objetivo de diseño específico, dado que el propósito de este escenario no es reproducir las condiciones de un emplazamiento real, sino evaluar el desempeño de los modelos *guided-informed* frente a excitaciones reales de alta duración y variabilidad. El uso directo de registros chilenos permite preservar la diversidad natural del contenido frecuencial y energético de los movimientos característicos de la región, lo que resulta más apropiado para analizar la capacidad de generalización y robustez de las arquitecturas bajo estudio.

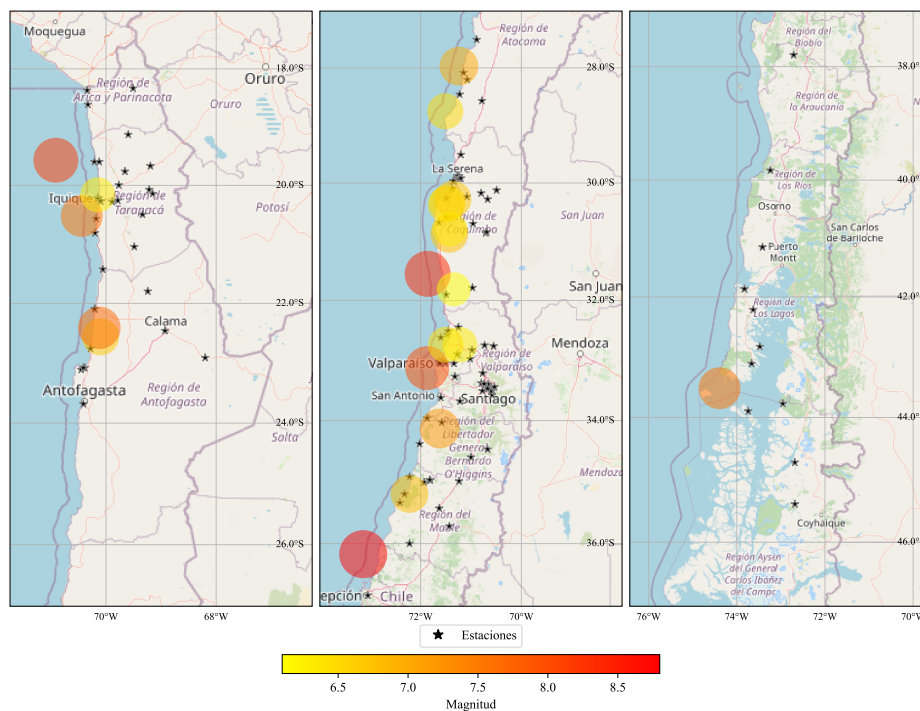


Figura 3.25: Mapa de los eventos sísmicos registrados (círculos) y estaciones sismológicas (cuadrados). Se incluyen 21 eventos observados por 120 estaciones, generando un total de 270 registros correspondientes a ambas direcciones de movimiento.

Del conjunto total, 78 registros presentan desplazamiento residual, definido como aquel cuyo desplazamiento final supera el 10% del desplazamiento máximo del registro, y 192 no lo presentan, representando un 28.89% de casos con deriva permanente. Con el fin de mantener un equilibrio entre ambos tipos de respuesta, los subconjuntos de entrenamiento, validación y prueba se generaron mediante una selección estratificada que asegura una proporción cercana a 50% de registros residuales y 50% no residuales en cada conjunto. Las proporciones globales se fijaron en 10% para entrenamiento, 5% para validación y 85% para prueba, resul-

tando en 27, 13 y 230 registros, respectivamente. El número reducido de datos de entrenamiento se estableció deliberadamente para representar escenarios realistas de limitada disponibilidad de información experimental o numérica, frecuente en el contexto estructural, donde la obtención de respuestas no-lineales o histeréticas resulta costosa, y evaluar así la capacidad de generalización del enfoque *guided-informed*. Adicionalmente, se definió un conjunto de colocación de 90 registros, entrenamiento, validación y 50 adicionales de prueba, utilizado exclusivamente para el ajuste de las restricciones físicas del modelo, manteniendo la imparcialidad de la evaluación final.

### 3.6.1. Desempeño de PhyCNN y Multi-LSTM en aislamiento sísmico con registros chilenos

#### Comparación global de métricas.

La Tabla 3.3 resume las métricas promedio (media  $\pm$  desviación estándar) sobre el conjunto de prueba. En comparación con los escenarios anteriores, las magnitudes de error son sensiblemente mayores, lo que refleja la combinación de no-linealidades severas, trayectorias no simétricas y derivas residuales prolongadas.

En desplazamiento, los errores promedio se sitúan entre  $L2RE = 1.6\text{--}3.4$ , con dispersiones amplias que evidencian alta variabilidad entre registros. Las PhyLSTM muestran promedios algo menores que las PhyCNN, pero con colas largas: logran ajustes muy buenos en algunos casos y se degradan bajo respuestas extremas con deriva y asimetrías. En velocidad, las cuatro arquitecturas exhiben un comportamiento más uniforme ( $L2RE \approx 0.8\text{--}0.9$ ), sin diferencias claras. En fuerza interna normalizada, las PhyLSTM *predichas* tienden a concentrar errores cercanos a la unidad: la falta de supervisión directa de la fuerza durante el entrenamiento induce sesgo y desacoplo respecto de las excitaciones. Las *fuerzas reconstruidas* a partir de derivadas de la velocidad presentan mayor varianza: combinan casos de muy buen ajuste con otros de pobre desempeño; las PhyCNN mantienen errores intermedios ( $L2RE \approx 0.6\text{--}0.7$ ) y menor dispersión. En la relación fuerza–desplazamiento, medida con  $L2RE\text{-}2D$ , los errores se ubican entre 1.5 y 2.6 con alta dispersión; PhyCNN–SO y PhyLSTM–2 muestran las medias más bajas, sin embargo PhyLSTM–2 presenta una mayor desviación.

Tabla 3.3: Métricas de desempeño promedio (media  $\pm$  desviación estándar) sobre el conjunto de prueba para el sistema Bouc–Wen representativo de edificio aislado bajo registros sísmicos chilenos reales. La métrica principal es L2RE, complementada por NRMSE y NMAE. La fila *Fuerza–Desplazamiento* reporta L2RE–2D.

Modelo	L2RE	NRMSE	NMAE
<i>Desplazamiento</i>			
PhyCNN–MO	3.404 $\pm$ 2.461	0.218 $\pm$ 0.100	0.167 $\pm$ 0.088
PhyCNN–SO	2.231 $\pm$ 2.086	0.155 $\pm$ 0.116	0.083 $\pm$ 0.066
PhyLSTM–2	<b>1.579 <math>\pm</math> 3.636</b>	<b>0.125 <math>\pm</math> 0.249</b>	<b>0.104 <math>\pm</math> 0.243</b>
PhyLSTM–3	1.847 $\pm$ 2.647	0.138 $\pm$ 0.179	0.115 $\pm$ 0.172
<i>Velocidad</i>			
PhyCNN–MO	0.861 $\pm$ 0.596	0.042 $\pm$ 0.026	0.016 $\pm$ 0.010
PhyCNN–SO (derivada)	0.848 $\pm$ 0.693	0.041 $\pm$ 0.030	0.014 $\pm$ 0.011
PhyLSTM–2	<b>0.835 <math>\pm</math> 0.542</b>	<b>0.043 <math>\pm</math> 0.025</b>	<b>0.015 <math>\pm</math> 0.009</b>
PhyLSTM–3	0.943 $\pm$ 0.602	0.048 $\pm$ 0.028	0.018 $\pm$ 0.010
<i>Fuerza interna normalizada</i>			
PhyCNN–MO	0.727 $\pm$ 0.487	0.040 $\pm$ 0.025	0.015 $\pm$ 0.011
PhyCNN–SO (reconstruida)	<b>0.607 <math>\pm</math> 0.488</b>	<b>0.033 <math>\pm</math> 0.025</b>	<b>0.012 <math>\pm</math> 0.010</b>
PhyLSTM–2 (predicha)	1.005 $\pm$ 0.021	0.060 $\pm$ 0.018	0.026 $\pm$ 0.011
PhyLSTM–2 (reconstruida)	1.362 $\pm$ 0.896	0.077 $\pm$ 0.047	0.027 $\pm$ 0.018
PhyLSTM–3 (predicha)	1.369 $\pm$ 0.482	0.078 $\pm$ 0.021	0.059 $\pm$ 0.023
PhyLSTM–3 (reconstruida)	1.434 $\pm$ 0.977	0.080 $\pm$ 0.051	0.028 $\pm$ 0.019
<i>Fuerza–desplazamiento (Histéresis) (L2RE–2D)</i>			
PhyCNN–MO	2.568 $\pm$ 1.756	0.154 $\pm$ 0.070	0.119 $\pm$ 0.061
PhyCNN–SO	1.713 $\pm$ 1.547	0.109 $\pm$ 0.082	0.058 $\pm$ 0.043
PhyLSTM–2	<b>1.577 <math>\pm</math> 2.793</b>	<b>0.108 <math>\pm</math> 0.173</b>	<b>0.082 <math>\pm</math> 0.170</b>
PhyLSTM–3	1.753 $\pm$ 2.004	0.117 $\pm$ 0.123	0.088 $\pm$ 0.119

### Distribución de errores y robustez.

La Figura 3.26 muestra la distribución del L2RE para desplazamiento, velocidad y fuerza interna normalizada, junto con el L2RE–2D para la relación histerética fuerza–desplazamiento. El eje vertical se acota a 0–10 para resaltar la zona central, aunque existen valores atípicos que superan ampliamente ese rango en varios registros con no linealidades importantes.

En desplazamiento, los modelos PhyCNN (MO y SO) exhiben medianas más altas y cajas menos compactas; las PhyLSTM reducen la mediana y el IQR, en especial PhyLSTM–2, pero presentan numerosas colas superiores y atípicos, lo que desplaza la media fuera del rango intercuartílico (asimetría por casos con muy mal desempeño).

En velocidad, las cuatro arquitecturas presentan un comportamiento muy similar: medianas próximas y dispersiones comparables, sin diferencias sistemáticas relevantes entre familias de modelos.

En fuerza interna normalizada, el patrón se invierte: la PhyCNN–SO alcanza la mediana más baja y la caja más compacta, seguida por PhyCNN–MO; las variantes PhyLSTM presentan medianas más altas y mayor dispersión, coherente con la sensibilidad de la fuerza reconstruida/predicha a errores de derivación en series largas y a la ausencia de supervisión directa de  $g(t)$  durante el entrenamiento.

En el plano fuerza–desplazamiento (L2RE–2D), PhyLSTM–2 conserva percentiles inferiores competitivos (mejores mínimos) pero con variabilidad amplia; PhyCNN–SO muestra una mediana comparable, mientras que PhyCNN–MO concentra los mayores valores y la mayor dispersión.

Nótese que L2RE y L2RE–2D no son magnitudes directamente comparables; deben interpretarse de manera conjunta para contrastar tendencias por arquitectura.

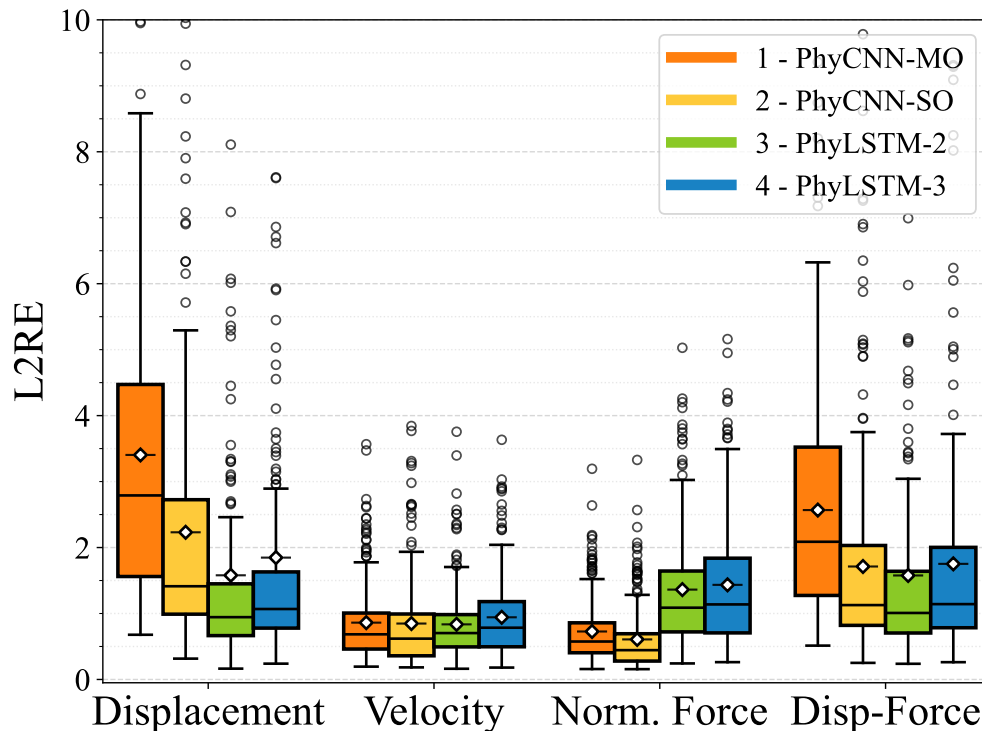


Figura 3.26: Distribución del L2RE en desplazamiento, velocidad y fuerza interna normalizada para el sistema Bouc–Wen representativo de un edificio aislado.

### Evolución de la función de pérdida.

La Figura 3.27 muestra la evolución de la función de pérdida durante el entrenamiento y la validación de las cuatro arquitecturas. En una primera fase todas las redes descienden con rapidez y, luego, convergen más lentamente. En **entrenamiento**, las PhyLSTM alcanzan valores del orden de  $10^{-4}$  (PhyLSTM–3 más baja, seguida de PhyLSTM–2), mientras que las PhyCNN se estabilizan alrededor de  $10^{-3}$ . En **validación**, **PhyLSTM–2** y **PhyLSTM–3** descienden hasta el entorno de  $10^{-3}$  pero se mantienen por encima de sus

curvas de entrenamiento; las PhyCNN, en cambio, forman mesetas cercanas a  $10^{-2}$  desde etapas tempranas. Este comportamiento debe leerse con cautela porque, aunque el conjunto total es amplio, la validación usa solo el 5 % de los datos (13 registros, frente al 10 % para entrenamiento, 27 registros), y su composición puede sesgar la métrica: por ejemplo, trayectorias con cambios del eje de oscilación, deriva residual o mayor amplitud pueden fijar una “meseta” de validación independiente del progreso en entrenamiento. En conjunto, las curvas sugieren que las LSTM optimizan más en entrenamiento y mejoran la validación, mientras que la generalización queda condicionada por el tamaño de validación y las características de esos registros. Las PhyCNN, además, exhiben una brecha temprana entre entrenamiento y validación, consistente con un estancamiento prematuro. Las oscilaciones intermedias visibles en todas las curvas pueden deberse a la heterogeneidad de los lotes o a ajustes del optimizador, aunque no es posible atribuirles una causa única con la evidencia disponible.

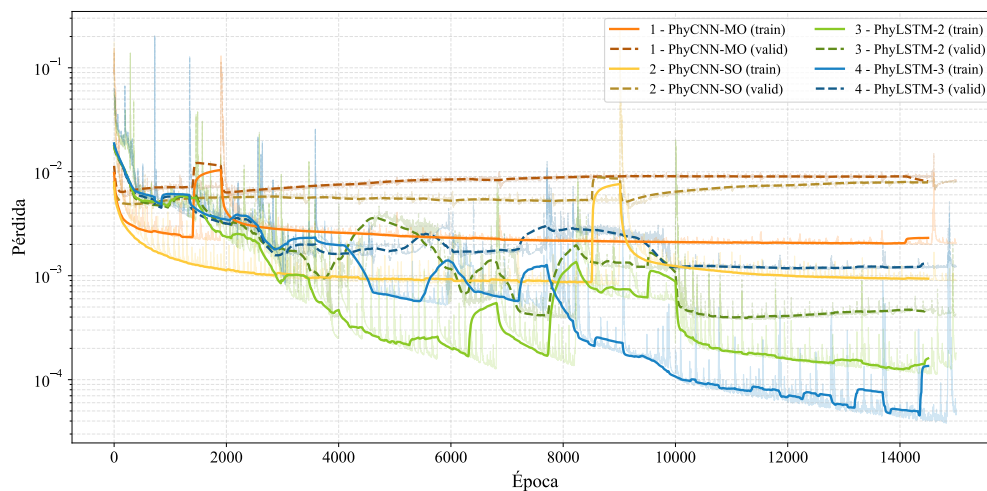


Figura 3.27: Evolución de la pérdida (suavizada con ventana de 500 épocas) durante entrenamiento y validación del escenario chileno.

### Análisis temporal de las predicciones.

La Figura 3.28 reúne, para cada variable, el registro de mejor y peor desempeño (según el L2RE promedio de los modelos). En los mejores casos, las PhyLSTM siguen con alta fidelidad la fase y la amplitud; en desplazamiento todas las arquitecturas muestran mayor dificultad para ajustarse a la referencia respecto de los escenarios previos: existe sobreestimación de picos en PhyLSTM y PhyCNN, aunque es menos marcada en las primeras. Las PhyCNN exhiben un sesgo de amplitud algo mayor y pequeñas derivas locales. En velocidad y fuerza, las cuatro curvas se solapan durante el pulso principal, pero persiste una leve sobreestimación de picos y episodios de ajuste local deficiente, más notorios en PhyCNN–MO. En fuerza normalizada, las predicciones de PhyLSTM aparecen algo más ruidosas y oscilantes que la referencia (y que las PhyCNN), manteniendo también cierta sobrestimación en los máximos.

En los peores casos se observan patrones sistemáticos que explican los L2RE elevados. En desplazamiento, aparece una deriva espuria junto con sesgos de amplitud: las PhyLSTM introducen una deriva sostenida

(residual no presente en la referencia), lo que sugiere que generalizan indebidamente patrones de registros con deriva real y los trasladan a casos sin deriva; además, presentan sobrestimaciones locales. Las PhyCNN, por su parte, tienden a desbordarse en amplitud e incorporar componentes de alta frecuencia. Estos sesgos se acumulan a lo largo del registro y explican los errores muy altos. En velocidad, los cuatro modelos muestran un desempeño similar: siguen el pulso principal pero pierden fase de manera progresiva y sobrestiman los máximos; adicionalmente, las PhyCNN exhiben oscilaciones alrededor de la trayectoria de referencia. En fuerza normalizada, las PhyCNN resultan más cercanas a la referencia en este ejemplo, con trazas más suaves y menor oscilación en los picos, mientras que las PhyLSTM presentan mayor ruido y desfases en los máximos. En conjunto, estas series confirman que el escenario chileno tensiona la generalización: las LSTM capturan mejor la dinámica en los buenos casos, pero son sensibles a derivas espurias o a no reproducir correctamente la deriva real cuando existe; las PhyCNN muestran un comportamiento más regular, aunque con sesgos de amplitud y dificultad para capturar desplazamientos residuales. En fuerza, las PhyCNN pueden aventajar a las PhyLSTM cuando estas últimas amplifican oscilaciones o incurrir en pequeños desfases.

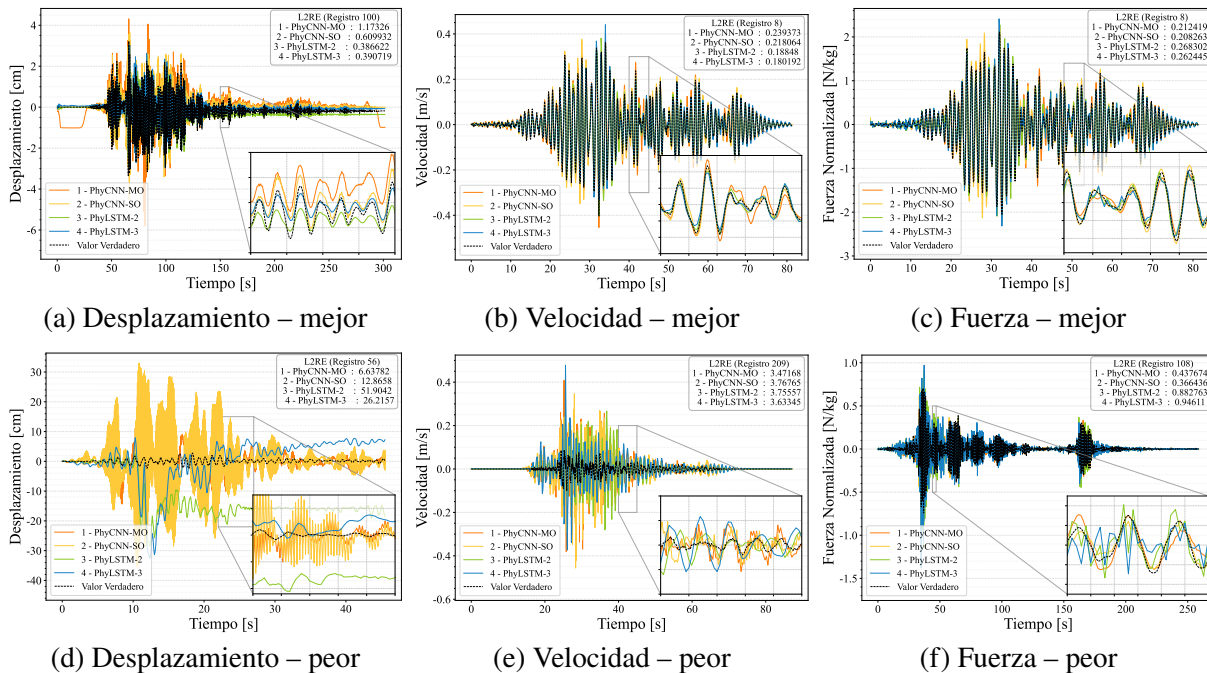


Figura 3.28: Predicciones temporales para el sistema Bouc–Wen aislado. La primera fila corresponde a los mejores registros y la segunda a los peores, según L2RE.

### Análisis espectral.

La Figura 3.29 muestra las PSD de desplazamiento, velocidad y fuerza interna normalizada para el registro de mejor desempeño en cada variable. Por restricciones de memoria, las PhyLSTM se entrenaron con submuestreo, esto es, con una frecuencia de muestreo menor que las PhyCNN; en consecuencia, su frecuencia de Nyquist también es menor. Todas las arquitecturas reproducen con fidelidad la banda de baja frecuencia donde se concentra la energía del sistema aislado. Cualquier incremento aparente en muy bajas frecuencias

debe interpretarse con cautela, pues puede originarse en efectos de estimación (ventaneo/solape de Welch y discretización) o en un detrending insuficiente, los cuales ensanchan picos y elevan el nivel cercano a 0 Hz sin implicar necesariamente un cambio dinámico real, sin embargo en la frecuencia más baja la PhyCNN–SO subestima la energía, consistente con su dificultad para reproducir desplazamientos residuales (cambios del eje de oscilación). Las diferencias se manifiestan hacia frecuencias medias y altas: en desplazamiento, las PhyCNN presentan un piso espectral más alto y un exceso persistente para  $f \gtrsim 6$  Hz, atribuible a pequeñas oscilaciones de alta frecuencia en las series temporales que la PSD magnifica. En velocidad, las discrepancias se atenúan: todas las curvas siguen la referencia en la banda dominante y el exceso en la cola es menor en las PhyLSTM. En fuerza interna normalizada, las PhyLSTM tienden a sobreestimar levemente la energía en la zona intermedia (2–5 Hz), mientras que las PhyCNN siguen mejor el decaimiento de alta frecuencia. Este piso espectral artificial observado en altas frecuencias para las arquitecturas que reconstruyen la fuerza interna a partir de las derivadas de  $\hat{u}(t)$  es consistente con la amplificación de ruido asociada al operador de segunda derivada. En efecto, al aproximar  $\ddot{u}$  mediante diferencias finitas centradas, el contenido de alta frecuencia de  $\hat{u}(t)$  se ve amplificado por un factor proporcional a  $1/\Delta t^2$ , lo que genera aceleraciones y fuerzas más ruidosas incluso cuando las predicciones de desplazamiento son visual y métricamente aceptables.

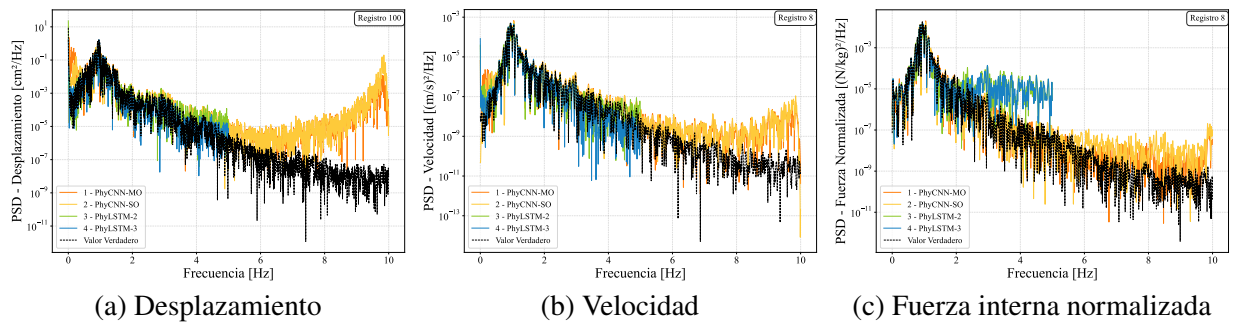


Figura 3.29: PSD de desplazamiento, velocidad y fuerza interna normalizada para el registro de mejor desempeño del sistema Bouc–Wen aislado bajo registros chilenos.

Dado que las señales no son estrictamente estacionarias, las PSD se presentan como referencia global y se complementan con espectrogramas (STFT) que revelan la evolución tiempo–frecuencia. En ambos casos se utilizó ventana de Hann; para las PSD se emplearon segmentos de 200 muestras con 75 % de solape, mientras que para los espectrogramas se usaron segmentos de 160 muestras para PhyCNN y de 80 muestras para PhyLSTM, con 80 % de solape. Adicionalmente, se incluye un mapa de *error PSD* definido como  $10 \log_{10} S_{\hat{x}} - 10 \log_{10} S_x$  (dB).

La Figura 3.30 compara el espectrograma de desplazamiento real con el de PhyCNN–SO y su error PSD. Se observa un exceso persistente de energía en la banda alta ( $\sim 6$ –10 Hz) durante prácticamente toda la duración, coherente con las colas de la PSD y lo observado en los graficos tiempo–historia; en las frecuencias más bajas, el error es pequeño salvo cerca de 0 Hz, donde puede aparecer sesgo asociado a derivas residuales espurias o no capturadas. Este patrón sugiere que la operación convolucional introduce componentes de alta frecuencia y presenta dificultades para representar cambios del eje de oscilación.

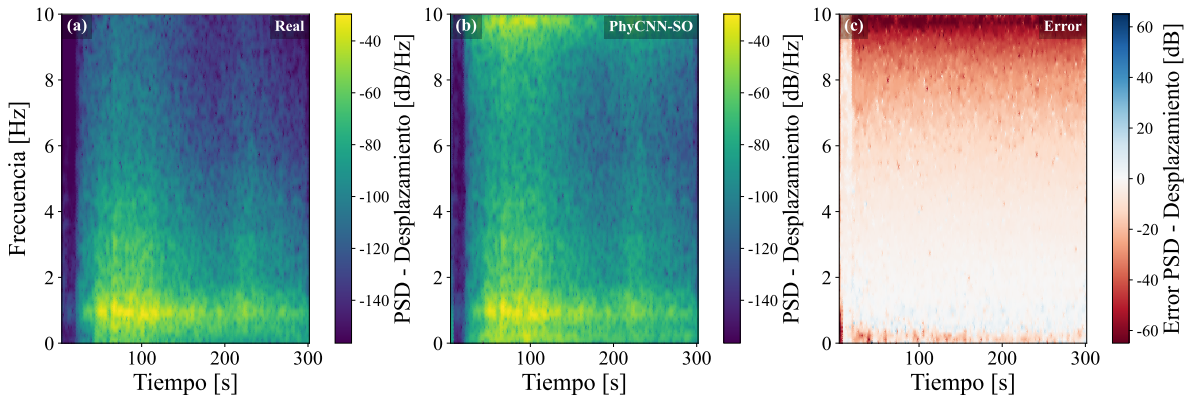


Figura 3.30: Espectrograma de desplazamiento para el registro con mejor desempeño (registro 100): (a) real, (b) PhyCNN–SO, (c) *error PSD* (dB).

La Figura 3.31 muestra el espectrograma de **fuerza interna normalizada** (reconstruida) para PhyLSTM–2. El error PSD se concentra en frecuencias medias–altas ( $f \gtrsim 3$  Hz) a lo largo de toda la secuencia, coherente con lo observado en las series tiempo–historia y con la derivación numérica utilizada para reconstruir la fuerza a partir de  $\hat{v}$ , que amplifica componentes de alta frecuencia.

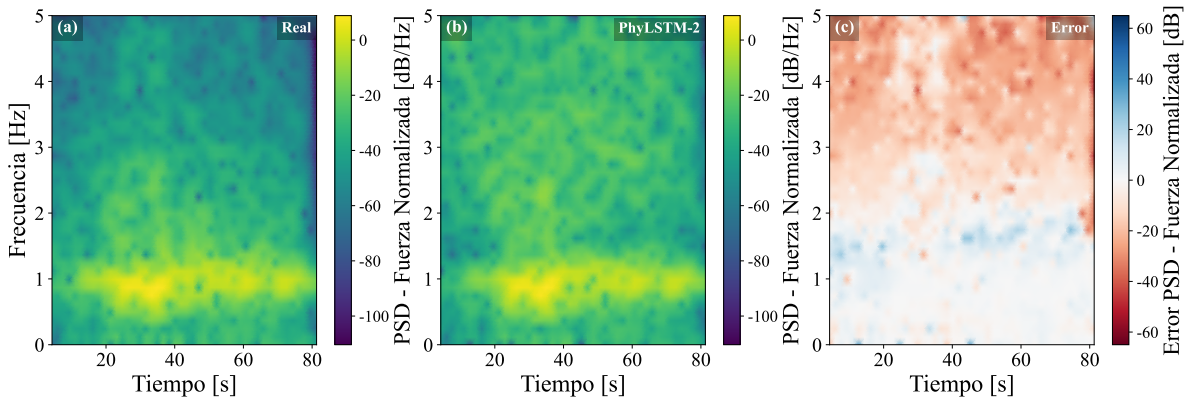


Figura 3.31: Espectrograma de fuerza interna normalizada para el registro con mejor desempeño (registro 8): (a) real, (b) PhyLSTM–2 (reconstruida), (c) *error PSD* (dB).

En síntesis, los espectrogramas confirman que el exceso de alta frecuencia es un fenómeno persistente: en PhyCNN aparece principalmente en el desplazamiento ( $f \gtrsim 6$  Hz), mientras que en PhyLSTM se manifiesta en la fuerza reconstruida ( $f \gtrsim 3$  Hz). Esto coincide con las colas de las PSD y con las diferencias observadas en las métricas y los graficos tiempo–historia: buen ajuste en la banda dominante y pequeñas discrepancias acumuladas en bandas altas que elevan los errores sin alterar la dinámica principal.

### Relación fuerza–desplazamiento.

La Figura 3.32 muestra los lazos fuerza–desplazamiento del registro de mejor desempeño (registro 8) junto con el error bidimensional  $L2RE_{2D}$ . Las variantes PhyLSTM son las que mejor reproducen la geometría

histerética ( $L2RE_{2D} \approx 0.24\text{--}0.26$ ): siguen de cerca la trayectoria de referencia, preservan la pendiente inicial, la curvatura de las ramas externas y el área disipada. Sus curvas se aprecian algo más angulosas, efecto atribuible al submuestreo usado en su entrenamiento, pero sin alterar de forma relevante la forma global del lazo.

Entre las convolucionales, las PhyCNN exhiben sesgos de forma y un ligero corrimiento del eje que estrecha/abre selectivamente las ramas de carga y descarga ( $L2RE_{2D} \approx 0.69$ ). La PhyCNN–MO presenta el mayor desajuste ( $L2RE_{2D} > 1$ ): lazos desplazados y más abiertos, pérdida de curvatura en los extremos y trayectorias que no cierran correctamente, compatibles con derivas espurias y desfases acumulados.

En síntesis, la memoria explícita de las LSTM permite integrar la historia de carga y la disipación con mayor coherencia, mientras que las CNN tienden a distorsionar la geometría del lazo cuando deben sostener dependencias de largo plazo.

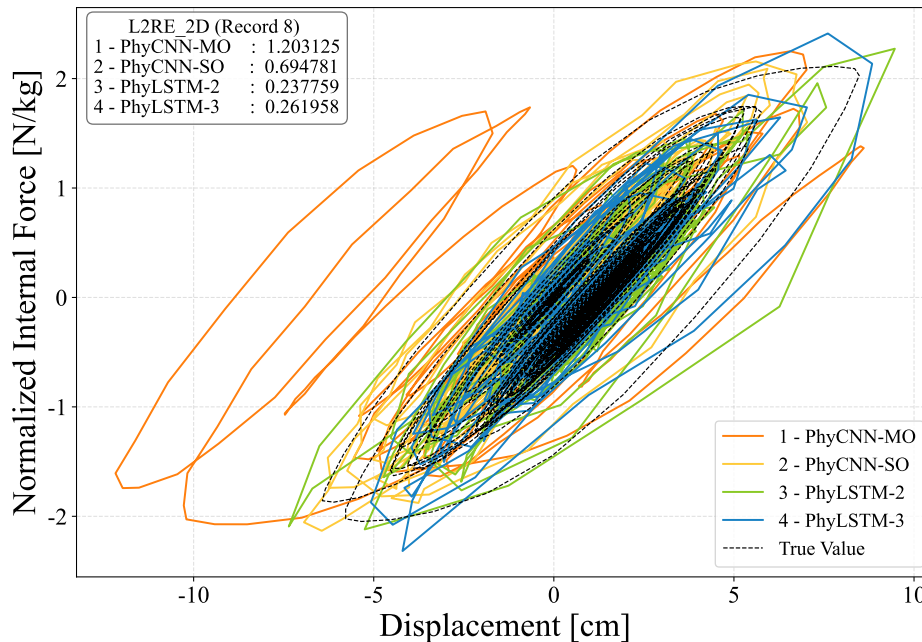


Figura 3.32: Curvas fuerza–desplazamiento correspondientes al registro de mejor desempeño (registro 8).

### Desplazamientos residuales.

Se evaluó explícitamente la capacidad de las arquitecturas para predecir el desplazamiento residual. La Figura 3.33 reúne, para el conjunto de prueba, los 100 registros con mayor residual verdadero (ordenados ascendientemente) y superpone el residual predicho por cada modelo. Los resultados muestran tres patrones robustos. Primero, las PhyCNN subestiman de forma sistemática la deriva en todo el rango: la mayoría de sus puntos se mantiene cercana a cero incluso cuando la referencia exhibe residuales altos, evidenciando una tendencia a recentrar la trayectoria alrededor del eje y a no capturar cambios de eje acumulados durante el movimiento. Segundo, las PhyLSTM presentan un comportamiento bimodal: en múltiples registros sin deriva real generan deriva espuria (sobre–predicción del residual) y, en registros con deriva marcada, alternan

aciertos con sobreestimaciones significativas. Esto sugiere que la red internaliza la existencia de derivas en la base de datos, pero no siempre logra condicionarlas a las combinaciones de historia de carga que efectivamente las producen. Tercero, ninguna arquitectura reproduce con estabilidad la transición gradual entre casos sin deriva y con deriva creciente: el escalón suave observado en la referencia se reemplaza, en las predicciones, por una nube dispersa sin monotonicidad clara.

La dificultad para capturar el residual se explica por la naturaleza integrativa del fenómeno. El desplazamiento residual puede escribirse como  $u_{\text{res}} = u(T) - u(0) = \int_0^T \dot{u}(t) dt$ ; por tanto, un sesgo pequeño pero persistente en la velocidad predicha (por ejemplo, una media ligeramente distinta de cero) se *acumula* a lo largo del tiempo y produce un corrimiento permanente proporcional a la duración del registro. Del mismo modo, desajustes sutiles de fase entre fuerza y velocidad rompen la simetría de los ciclos histeréticos, de modo que la energía neta disipada/inyectada en ida y vuelta no se compensa y la trayectoria queda desplazada. En términos prácticos, errores pequeños en la media de  $\dot{u}(t)$ , en la fase o en el nivel de la fuerza, imperceptibles a escala de ciclo, acaban manifestándose como error permanente cuando se consideran secuencias largas.

En síntesis, las PhyCNN muestran un comportamiento más estable pero con una clara propensión a *anular* la deriva, mientras que las PhyLSTM capturan mejor la posibilidad de residuales grandes a costa de introducir derivas espurias en varios casos. Estos resultados indican que, sin una penalización explícita sobre el residuo estático y sin términos de pérdida que den observabilidad cuasi-estática, las arquitecturas evaluadas tienen dificultades para decidir cuándo debe aparecer, o no, un desplazamiento residual.

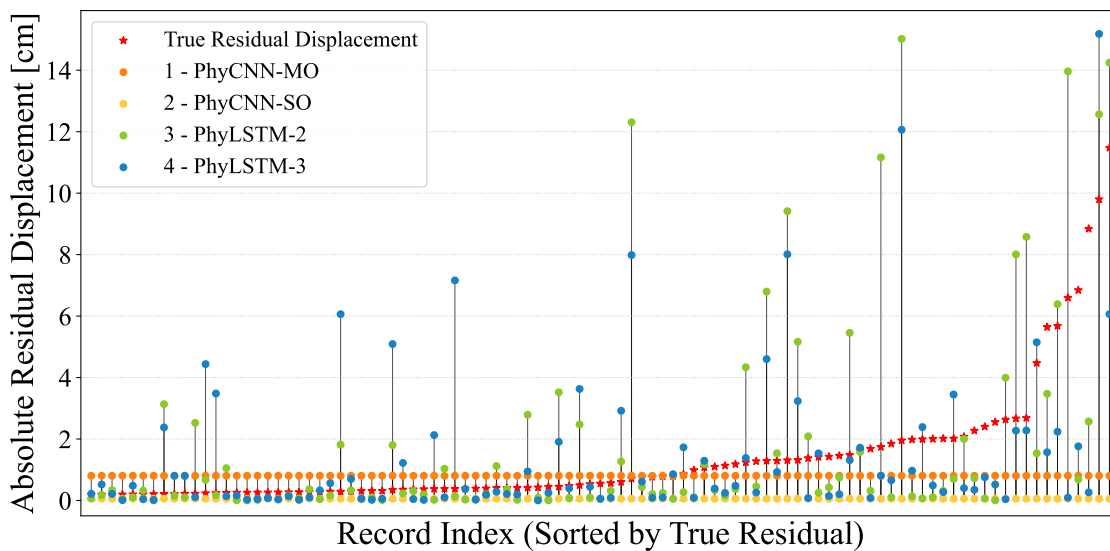


Figura 3.33: Desplazamiento residual absoluto por registro para los 100 registros con mayor desplazamiento residual (ordenado por el residual verdadero)

### Síntesis del Escenario III

Este escenario lleva a las arquitecturas a su límite operativo: los errores aumentan en magnitud y dispersión debido a la no-linealidad severa del sistema, la larga duración de los registros y la presencia de derivas residuales. Las PhyLSTM muestran, en sus mejores casos, una mayor fidelidad a la dinámica dominante

y a la geometría histerética; no obstante, presentan episodios de inestabilidad local reflejados en derivas espurias y sobreestimaciones puntuales. Las PhyCNN exhiben sesgos persistentes: piso espectral elevado, sobre/subestimación de amplitudes y recentrado del desplazamiento, que limitan la reproducción de derivas y de lazos amplios. En conjunto, la evidencia sugiere que el desempeño queda condicionado por la componente cuasi-estática de la respuesta (desplazamiento final y equilibrio de muy baja frecuencia) y por la acumulación de errores a lo largo de secuencias extensas. Estos efectos explican, de manera coherente con los análisis tiempo-historia y espectrales, tanto los valores medios de error como su gran variabilidad entre registros.

### 3.7. Conclusiones y Trabajo Futuro

En este trabajo se evaluó el desempeño de arquitecturas neuronales informadas por la física bajo el paradigma *guided-informed* (PhyCNN y PhyLSTM) para la predicción de respuestas sísmicas no-lineales. El estudio abarcó desde sistemas con no-linealidad moderada hasta escenarios de alta demanda inelástica bajo registros de larga duración, permitiendo identificar los límites de validez de estos enfoques.

Los resultados experimentales demuestran que, en sistemas con histéresis estable y excitaciones de banda ancha, las arquitecturas basadas en LSTM superan consistentemente a las variantes convolucionales. La capacidad de la celda recurrente para mantener un estado de memoria explícito permitió reducir el error de predicción en desplazamiento y velocidad hasta en un 70 % respecto a las redes convolucionales, confirmando su ventaja estructural para modelar la dependencia histórica inherente a la dinámica no-lineal. Sin embargo, esta ventaja se diluye al enfrentar registros sísmicos reales de larga duración y comportamiento histerético intenso. En este escenario, se observó una pérdida significativa de robustez en todas las arquitecturas evaluadas, evidenciada por un aumento drástico en la dispersión del error y la aparición de inestabilidades locales. Esto indica que la incorporación de restricciones físicas "blandas", mediante penalización en la función de pérdida, resulta insuficiente por sí sola para garantizar la estabilidad numérica en horizontes de predicción extensos cuando la supervisión de datos es limitada.

Un hallazgo crítico de esta investigación radica en los modos de falla diferenciados respecto a la predicción de desplazamientos permanentes. Las arquitecturas convolucionales mostraron una tendencia sistemática a recentrar la respuesta, subestimando la magnitud del *drift* residual. Por el contrario, las arquitecturas recurrentes, aunque capaces de capturar mejor la geometría de los lazos de histéresis, exhibieron inestabilidad temporal, generando frecuentemente derivas espurias en registros que no presentaban daño permanente. Asimismo, la estrategia de predecir la fuerza interna como una variable de salida directa sin supervisión demostró ser frágil, resultando en errores elevados y desacoplamiento físico. La reconstrucción de la fuerza a partir de la derivada de la velocidad predicha se perfila como una estrategia más robusta para garantizar la consistencia dinámica, aun a costa de introducir ruido de alta frecuencia en la estimación.

Estos hallazgos sugieren que, para avanzar hacia modelos surrogados confiables en ingeniería sísmica, es necesario superar la formulación *guided-informed* clásica. Las líneas de investigación futura deberían orientarse hacia la implementación de arquitecturas *solver-informed* que integren esquemas numéricos rígidos, como métodos de Runge-Kutta implícitos, directamente dentro del grafo computacional para garantizar la estabilidad. De igual manera, se requiere el desarrollo de funciones de pérdida especializadas que penalicen

explícitamente el error de deriva y el balance energético acumulado, así como la exploración de estrategias de entrenamiento curricular para mejorar la convergencia en superficies de optimización complejas. Estas direcciones son fundamentales para habilitar el uso efectivo de PINNs en la evaluación de riesgo sísmico y el diseño basado en desempeño.

## Capítulo 4

# RK4Net: Un Integrador Neuronal Autónomo de Alto Orden para Dinámica Estructural no-lineal

### 4.1. Introducción

En la ingeniería sísmica y estructural moderna, la capacidad de evaluar la integridad de los sistemas civiles de manera instantánea se ha convertido en un requisito crítico. Aplicaciones de vanguardia como la Simulación Híbrida en Tiempo Real (RTHS) y el Monitoreo de Salud Estructural (SHM) demandan modelos digitales capaces de predecir la respuesta dinámica del sistema con latencias mínimas para permitir la toma de decisiones automatizada (Carrion y Spencer, 2007; Bas y Moustafa, 2020). Sin embargo, esta necesidad de inmediatez entra en conflicto directo con la complejidad física inherente a las estructuras civiles, especialmente cuando son sometidas a excitaciones severas, exhiben comportamientos altamente no-lineales, tales como degradación de rigidez, fluencia y disipación de energía por histéresis, cuya modelación mediante métodos tradicionales, como los Elementos Finitos (FEM) de alta fidelidad, conlleva un costo computacional prohibitivo para aplicaciones en tiempo real o para procesos iterativos masivos (Palacio-Betancur y Gutierrez Soto, 2022).

Tradicionalmente, para sortear esta barrera computacional, la práctica ingenieril ha optado por modelos reducidos basados en simplificaciones lineales. Si bien estos modelos son eficientes en velocidad, a menudo sacrifican la precisión necesaria para capturar fenómenos críticos de daño. Este desafío se exagera en el contexto del SHM, donde frecuentemente se desconoce el modelo físico *a priori*. En estructuras existentes, parámetros fundamentales como la rigidez real, las condiciones de borde efectivas o el amortiguamiento son inciertos, y se cuenta únicamente con datos de respuesta provenientes de sensores (Farrar y Worden, 2007). Intentar ajustar un modelo físico simplificado a estos datos complejos constituye un problema inverso mal planteado, donde las aproximaciones tradicionales de actualización de modelos pueden fallar en representar la realidad física del sistema dañado (Mottershead y Friswell, 1993).

Como solución a este compromiso entre eficiencia y precisión, ha emergido el paradigma de los modelos *surrogados* o metamodelos. El uso de técnicas de aprendizaje profundo (*Deep Learning*) permite generar aproximaciones computacionalmente eficientes de sistemas complejos, siendo especialmente útiles en RTHS donde actúan como subestructuras computacionales rápidas (Bas y Moustafa, 2020). Su utilidad es evidente

también en análisis probabilísticos de desempeño sísmico, donde la necesidad de ejecutar miles de simulaciones de Monte Carlo ante diversas series de entrada hace inviable el uso de modelos FEM complejos. Una estrategia eficiente consiste en generar un conjunto limitado de simulaciones de alta fidelidad para entrenar al modelo surrogado, y posteriormente ejecutar el grueso de las simulaciones utilizando esta aproximación neuronal.

No obstante, la implementación de modelos surrogados basados en aprendizaje profundo presenta limitaciones significativas cuando se enfrentan a la dinámica estructural compleja. Estudios recientes, como los desarrollados por Zhang et al. (2020a) y Zhang et al. (2020b), han explorado el uso de redes tipo *sequence-to-sequence* (Seq2Seq) parcialmente informadas por la física. Si bien estas arquitecturas ofrecen un rendimiento aceptable en regímenes lineales o moderadamente no-lineales, su eficacia disminuye ante sistemas con histéresis severa o excitaciones sísmicas de larga duración. Una deficiencia crítica de estas redes es su tendencia a acumular errores de integración (fenómeno conocido como *drift*), lo que resulta en una incapacidad para predecir correctamente los desplazamientos residuales o deformaciones permanentes al finalizar el evento sísmico.

La capacidad de predecir desplazamientos residuales no es un detalle académico, sino una necesidad de seguridad pública. Un ejemplo paradigmático se observa en el comportamiento de puentes chilenos durante grandes terremotos (e.g., Maule 2010), donde el diseño permitía el deslizamiento entre la superestructura y los apoyos elastoméricos sin conexión mecánica fija (Wilches et al., 2021). Como documentan Buckle et al. (2012) y Elnashai et al. (2012), la falta de capacidad de recentrado provocó desplazamientos residuales excesivos y, en casos extremos, la pérdida de apoyo del tablero. Un modelo digital que suavice la respuesta y no capture esta deformación permanente es de utilidad limitada para la evaluación rápida post-desastre y la toma de decisiones sobre la habitabilidad de la estructura.

Ante las limitaciones de las arquitecturas puramente secuenciales, la literatura reciente ha comenzado a virar hacia enfoques denominados *solver-informed* o *Neural ODEs*. A diferencia de los modelos anteriores, estas arquitecturas incorporan explícitamente operadores de avance temporal para emular integradores numéricos clásicos, mejorando la interpretabilidad y la estabilidad en trayectorias largas. Investigaciones recientes han propuesto esquemas inspirados en integradores explícitos e implícitos, fórmulas de Runge–Kutta y esquemas recurrentes en espacio de estados (Sadeghi Eshkevari et al., 2021; Zhai et al., 2023; Su et al., 2024). Asimismo, el marco de las *Neural ODEs* ha ofrecido rutas prometedoras para la identificación estructural continua (Lai et al., 2021; Guo et al., 2023). Sin embargo, el entrenamiento directo de estos integradores de orden superior a menudo enfrenta desafíos de convergencia y estabilidad numérica cuando se desconocen los parámetros iniciales.

Alineado con esta tendencia, pero buscando abordar las dificultades de entrenamiento observadas, este trabajo propone una línea de investigación basada en Redes Neuronales Informadas por la Física (PINNs) con un cambio de paradigma incremental. Se plantea transformar la red de una “caja negra” a una “caja gris” que emule la estructura de un integrador numérico. La hipótesis fundamental es que, al obligar a la arquitectura a aprender la ecuación de estado del sistema ( $\dot{x} = f(x, t)$ ) y resolverla paso a paso, se garantiza consistencia temporal y causalidad física. Este enfoque basado en la integración numérica ofrece ventajas superiores más allá de la precisión: mejora la capacidad de generalización ante eventos sísmicos inéditos y

otorga interpretabilidad al modelo, permitiendo inspeccionar los estados internos para validar el aprendizaje de la rigidez o las fuerzas restauradoras.

Es imperativo establecer que las arquitecturas neuronales propuestas en este capítulo (EulerNet y RK4Net) constituyen modelos de carácter **experimental** y representan un primer paso incremental hacia el desarrollo de integradores neuronales robustos para ingeniería sísmica. En consecuencia, la evaluación presentada se restringe exclusivamente a sistemas de **Un Grado de Libertad (SDOF)** operando en régimen **autónomo** (vibración libre). El objetivo central de esta fase es aislar y evaluar la capacidad de la red para aprender la topología interna del sistema dinámico (rigidez, amortiguamiento e histéresis) y resolver la integral temporal de forma recursiva a partir de una condición inicial, sin la influencia de perturbaciones externas. Por tanto, la incorporación de excitaciones exógenas, tales como registros sísmicos o fuerzas externas variables en el tiempo, queda explícitamente **fuera del alcance de este capítulo**. La generalización de estos modelos hacia sistemas no autónomos (forzados) introduce complejidades adicionales relacionadas con el mapeo de entradas dependientes del tiempo, las cuales serán abordadas en etapas subsecuentes de esta investigación.

Para validar esta hipótesis bajo las condiciones descritas y superar los problemas de optimización de los esquemas complejos, este capítulo plantea un desarrollo metodológico incremental organizado de la siguiente manera:

- La **Sección 4.2 (Fase I)** presenta la formulación de la arquitectura base *EulerNet*, validando su capacidad para capturar la dinámica global en sistemas lineales y con no-linealidad geométrica suave.
- La **Sección 4.3 (Fase II)** diagnostica las limitaciones de los integradores de primer orden frente a sistemas rígidos y caóticos (Bouc–Wen, Duffing Bi-estable), utilizando un análisis de sensibilidad al paso temporal para discriminar entre errores de aprendizaje y errores numéricos de truncamiento.
- La **Sección 4.4 (Fase III)** introduce la arquitectura avanzada de alto orden *RK4Net*. Se aborda la paradoja de su optimización (*Cold Start*) y se demuestra la eficacia de la estrategia de estabilización propuesta (*Warm Start*), logrando una predicción precisa de la histéresis y estabilidad a largo plazo.

## 4.2. Marco Teórico: Modelado Dinámico y Espacio de Estados

### 4.2.1. Ecuación General de Movimiento

La respuesta dinámica de un sistema de  $n$  grados de libertad (MGDL) se rige por un sistema de ecuaciones diferenciales de segundo orden acopladas:

$$M\ddot{\mathbf{u}}(t) + C\dot{\mathbf{u}}(t) + \mathbf{f}_s(\mathbf{u}, \dot{\mathbf{u}}) = \mathbf{F}(t) \quad (4.1)$$

Donde  $M$  y  $C$  son las matrices de masa y amortiguamiento, respectivamente;  $\mathbf{u}(t)$  es el vector de desplazamientos y  $\mathbf{f}_s$  es el vector de fuerzas de restitución. En el contexto de este capítulo, centrado en la validación de integradores autónomos, se considera  $\mathbf{F}(t) = \mathbf{0}$  (vibración libre).

### 4.2.2. Modelos de Fuerzas de Restitución No-lineales

Para evaluar la capacidad de aprendizaje de las redes neuronales ante topologías complejas, se emplean dos modelos constitutivos:

#### Oscilador de Duffing (No-linealidad Geométrica)

Representa sistemas con rigidez variable, incluyendo un término cúbico que modela el endurecimiento (*hardening*) del material:

$$f_s(u) = k_1u + k_3u^3 \quad (4.2)$$

#### Modelo de Bouc-Wen (Histéresis)

Describe el comportamiento inelástico y disipativo con memoria. La fuerza se compone de una parte elástica y una histerética gobernada por una variable interna  $z$ :

$$f_s(u, z) = \alpha ku + (1 - \alpha)kz \quad (4.3)$$

La evolución de  $z$  sigue la ecuación diferencial constitutiva:

$$\dot{z} = A\dot{u} - \beta|\dot{u}||z|^{n-1}z - \gamma\dot{u}|z|^n \quad (4.4)$$

### 4.2.3. Formulación en el Espacio de Estados

Para habilitar el uso de aproximadores universales tipo red neuronal, se transforma el sistema de segundo orden a uno de primer orden. Definimos el vector de estado aumentado  $\mathbf{x}$ :

$$\mathbf{x} = \begin{bmatrix} \mathbf{u} \\ \dot{\mathbf{u}} \\ \boldsymbol{\mu} \end{bmatrix} \in \mathbb{R}^{2n+m} \quad (4.5)$$

Donde  $\boldsymbol{\mu}$  contiene las variables internas (como  $z$ ). Así, la dinámica se compacta en:

$$\dot{\mathbf{x}} = \mathbf{g}(\mathbf{x}) \quad (4.6)$$

Esta función vectorial  $\mathbf{g}(\mathbf{x})$  es el objeto latente que la red neuronal  $\mathcal{N}_\theta$  deberá aprender a aproximar.

## 4.3. Fase I: Arquitectura Base EulerNet y Validación en Sistemas Suaves

El objetivo de esta fase es validar la arquitectura neuronal más simple posible, demostrando que es capaz de aprender la topología de sistemas dinámicos estables y operar como un integrador numérico autónomo sin divergencia.

### 4.3.1. Formulación del Integrador Neuronal (EulerNet)

Se propone una red neuronal parametrizada por  $\theta$  para actuar como el aproximador del campo vectorial del sistema. La arquitectura recibe el estado actual  $\mathbf{x}_t$  y devuelve la predicción de su derivada temporal:

$$\dot{\mathbf{x}}_{pred,t} = \mathcal{N}_\theta(\mathbf{x}_t) \quad (4.7)$$

Con el objetivo de reproducir fielmente la historia temporal de los estados y no solo valores instantáneos, la predicción de la red se acopla a un esquema de integración numérica de **Euler hacia adelante** (Forward Euler). Bajo este esquema, el estado en el paso de tiempo siguiente se calcula como:

$$\mathbf{x}_{t+\Delta t} = \mathbf{x}_t + \mathcal{N}_\theta(\mathbf{x}_t) \cdot \Delta t \quad (4.8)$$

El objetivo es que  $\mathcal{N}_\theta(\mathbf{x}) \approx g(\mathbf{x})$  para cualquier punto dentro del dominio de interés, logrando que la red represente las trayectorias en el plano de fase como combinaciones de los modos propios de vibración del sistema. Esta formulación residual obliga a la red a aprender la tasa de cambio instantánea escalada por el paso de tiempo físico  $\Delta t$ . El entrenamiento se realiza minimizando el Error Cuadrático Medio (MSE) entre el estado integrado predicho  $\hat{\mathbf{x}}_{t+\Delta t}$  y el estado real objetivo.

### 4.3.2. Arquitectura de la Red Neuronal

Para aproximar la función de transición de estados, se implementa una arquitectura de tipo Perceptrón Multicapa (MLP). Dado que el sistema busca modelar dinámicas físicas continuas y diferenciables (vibraciones mecánicas), la elección de las funciones de activación es crítica.

La red  $\mathcal{N}_\theta$  se define como una función de mapeo  $\mathbb{R}^{2n} \rightarrow \mathbb{R}^{2n}$ , estructurada de la siguiente manera:

- **Capa de Entrada:** Recibe el vector de estado actual  $\mathbf{x}_t$  de dimensión  $2n$  (desplazamientos y velocidades concatenados).
- **Capas Ocultas:** Se dispone de una arquitectura fija para todos los experimentos, consistente en **dos capas densas** (*fully connected*) consecutivas con **64 neuronas cada una**. Matemáticamente, la operación en la capa  $l$ -ésima se describe como:

$$\mathbf{h}_l = \sigma(\mathbf{W}_l \mathbf{h}_{l-1} + \mathbf{b}_l) \quad (4.9)$$

Donde  $\mathbf{W}_l \in \mathbb{R}^{64 \times 64}$  y  $\mathbf{b}_l \in \mathbb{R}^{64}$  son los pesos y sesgos aprendibles, y  $\sigma(\cdot)$  es la función de activación no-lineal.

- **Función de Activación:** Se selecciona la tangente hiperbólica ( $\tanh$ ) para todas las capas ocultas. A diferencia de funciones lineales a trozos (como ReLU), la  $\tanh$  es suave, acotada en  $[-1, 1]$  y centrada en cero, lo cual facilita la representación de comportamientos oscilatorios suaves típicos de sistemas estructurales y evita discontinuidades en el cálculo de gradientes de orden superior.

- **Capa de Salida:** Una transformación lineal final que proyecta las **64 características latentes** de la última capa oculta de vuelta al espacio  $\mathbb{R}^{2n}$ , representando la estimación del vector derivada  $\dot{\mathbf{x}}_t$ .

Es importante destacar que esta configuración (profundidad de 2 capas y ancho de 64 unidades) se mantuvo **fija e inmutable** a lo largo de todos los experimentos de este capítulo. Se optó por mantener una arquitectura fija para todos los experimentos, **prescindiendo de optimización de hiperparámetros** (como *Grid Search* o búsqueda Bayesiana). Esta decisión metodológica busca aislar el desempeño del esquema de integración numérico (Euler vs. RK4) como la única variable independiente del estudio, asegurando que las mejoras en la precisión provengan de la formulación matemática del integrador y no de un aumento arbitrario en la capacidad de representación de la red neuronal.

### 4.3.3. Estrategia de Entrenamiento y Función de Pérdida

El entrenamiento de la red no se realiza supervisando directamente la derivada  $\dot{\mathbf{x}}$ , sino supervisando el *estado en el siguiente paso de integración*. Esto fuerza a la red a aprender un campo vectorial que sea consistente con el esquema numérico de Euler definido en la Ecuación 4.8.

Para un conjunto de datos de entrenamiento  $\mathcal{D} = \{(\mathbf{x}_t^{(i)}, \mathbf{x}_{t+\Delta t}^{(i)})\}_{i=1}^N$ , donde  $\mathbf{x}_{t+\Delta t}^{(i)}$  representa el estado futuro real (*ground truth*) obtenido de la simulación o medición, definimos la predicción del modelo como  $\hat{\mathbf{x}}_{t+\Delta t}^{(i)}$ .

La función de pérdida ( $\mathcal{L}$ ) se define como el Error Cuadrático Medio (MSE) entre el estado integrado por la red y el estado real objetivo:

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^N \left\| \mathbf{x}_{t+\Delta t}^{(i)} - \underbrace{(\mathbf{x}_t^{(i)} + \mathcal{N}_\theta(\mathbf{x}_t^{(i)}) \cdot \Delta t)}_{\hat{\mathbf{x}}_{t+\Delta t}^{(i)}} \right\|_2^2 \quad (4.10)$$

Esta formulación tiene una doble ventaja física:

1. **Regularización Implícita:** Al incluir  $\Delta t$  en el grafo computacional, la red aprende a escalar sus salidas para corresponder a tasas de cambio físicas reales ( $\dot{\mathbf{u}}$ ,  $\ddot{\mathbf{u}}$ ).
2. **Consistencia Dinámica:** Se penaliza el error en la trayectoria, asegurando que la aproximación local del campo vectorial conduzca al estado futuro correcto tras la integración.

La optimización de los parámetros  $\theta$  se realiza mediante el algoritmo Adam, utilizando propagación hacia atrás (*Backpropagation*) a través del solucionador Euler para calcular los gradientes  $\nabla_\theta \mathcal{L}$ .

### 4.3.4. Experimento 1: Validación en Sistema Lineal

En esta etapa inicial, se busca que la red neuronal aproxime el campo vectorial de un sistema lineal sin excitación externa, validando su capacidad de aprendizaje sobre el espacio de estados mediante la variación de condiciones iniciales.

Para esto se simuló un marco de acero de un grado de libertad ( $m = 15$  ton,  $k = 3500$  kN/m,  $\xi \approx 2\%$ ) sometido a vibración libre. El modelo se evaluó en régimen 100 % recursivo (autónomo) sobre trayectorias no vistas.

### Definición del Vector de Estado y Variables

Para un sistema de  $n$  grados de libertad, definimos el vector de desplazamientos como  $\mathbf{u} \in \mathbb{R}^n$ . El vector de estado  $\mathbf{x}$  se construye para reducir el sistema a primer orden:

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{u}(t) \\ \dot{\mathbf{u}}(t) \end{bmatrix} \in \mathbb{R}^{2n} \quad (4.11)$$

La dinámica del sistema lineal en vibración libre ( $a_g = 0$ ) está gobernada por:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) \quad (4.12)$$

Donde la matriz de estado  $\mathbf{A}$  contiene la información de masa, rigidez y amortiguamiento:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C} \end{bmatrix} \quad (4.13)$$

### Generación de Datos Sintéticos y Definición del Sistema

Para el entrenamiento y validación de la red neuronal, se generó un conjunto de datos sintéticos (Ground Truth) mediante integración numérica. Con el propósito de trascender el ámbito de los problemas matemáticos puramente abstractos y establecer un primer acercamiento hacia la práctica de la ingeniería sísmica, se define un caso de estudio que incorpora parámetros físicos realistas. Si bien este modelo constituye una idealización numérica y su aplicación directa a estructuras complejas construidas es aún distante, emula las propiedades dinámicas fundamentales (masa, rigidez y amortiguamiento) de un sistema de un grado de libertad (SDOF) representativo de un marco de acero sometido a vibración libre.

La dinámica del sistema está gobernada por la ecuación de movimiento lineal:

$$m\ddot{u}(t) + c\dot{u}(t) + ku(t) = 0 \quad (4.14)$$

**Método de Integración Numérica** La solución paso a paso se obtuvo utilizando el método de Newmark- $\beta$ , un esquema implícito incondicionalmente estable para sistemas lineales. Se seleccionaron los parámetros  $\gamma = 0.5$  y  $\beta = 0.25$  (método de la aceleración promedio constante), lo que garantiza la conservación de la energía en ausencia de amortiguamiento físico y minimiza el error numérico algorítmico.

Las ecuaciones de actualización de estado para un paso de tiempo  $\Delta t$  son:

$$\dot{u}_{t+\Delta t} = \dot{u}_t + [(1 - \gamma)\ddot{u}_t + \gamma\ddot{u}_{t+\Delta t}] \Delta t \quad (4.15)$$

$$u_{t+\Delta t} = u_t + \dot{u}_t \Delta t + \left[ \left( \frac{1}{2} - \beta \right) \ddot{u}_t + \beta \ddot{u}_{t+\Delta t} \right] \Delta t^2 \quad (4.16)$$

**Parámetros del Sistema Estructural** Para asegurar que los resultados sean representativos de una aplicación de ingeniería civil, se seleccionaron parámetros correspondientes a una estructura de acero típica de baja altura (e.g., una nave industrial o un marco resistente a momento de un piso).

Los valores adoptados se detallan en la Tabla 4.1. Se asumió una masa sísmica de 15 toneladas y una rigidez lateral que resulta en un periodo fundamental  $T_n \approx 0.41$  s, característico de estructuras rígidas. El amortiguamiento se fijó en un  $\xi \approx 2\%$ , valor estándar para estructuras de acero en rango elástico.

Tabla 4.1: Definición de parámetros físicos para el marco de acero simulado.

Parámetro	Símbolo	Valor
<i>Propiedades Físicas</i>		
Masa Sísmica	$m$	15,000 kg
Rigidez Lateral	$k$	3,500 kN/m
Amortiguamiento	$c$	9.0 kNs/m
<i>Propiedades Dinámicas Resultantes</i>		
Frecuencia Natural	$\omega_n$	15.275 rad/s
Periodo Natural	$T_n$	0.411 s
Razón de Amortiguamiento	$\xi$	1.96 %
<i>Configuración de Simulación</i>		
Paso de tiempo	$\Delta t$	0.01 s
Duración total	$T_{total}$	20.0 s

**Espacio de Muestreo y Condiciones Iniciales** Para que la red neuronal aprenda la topología global del espacio de fases, se generaron  $N = 100$  trayectorias variando estocásticamente las condiciones iniciales  $(u_0, v_0)$ .

Es crucial que estas condiciones iniciales sean físicamente consistentes con la hipótesis de linealidad. Desplazamientos excesivos implicarían daño estructural (fluencia), invalidando el modelo lineal utilizado ( $ku$ ). Por ello, se restringió el muestreo a los siguientes rangos operativos:

$$u_0 \sim \mathcal{U}(-0.05, 0.05) \text{ m}, \quad v_0 \sim \mathcal{U}(-0.7, 0.7) \text{ m/s} \quad (4.17)$$

#### Justificación de los rangos:

- **Desplazamiento ( $\pm 5$  cm):** Considerando una altura de entrepiso típica de  $h = 4.0$  m, un desplazamiento de 5 cm corresponde a una distorsión de entrepiso (*drift*) del 1.25 %. Este valor se encuentra en el límite del comportamiento elástico aceptable para marcos de acero antes de incursionar en deformaciones plásticas significativas.

- **Velocidad ( $\pm 0.7$  m/s):** Este rango es consistente energéticamente con los desplazamientos máximos. Dado que la velocidad máxima en vibración libre se aproxima como  $v_{max} \approx \omega_n u_{max}$ , para nuestro sistema ( $15.27 \times 0.05 \approx 0.76$ ), limitar la velocidad a 0.7 m/s asegura que la energía cinética inicial no exceda la capacidad de almacenamiento de energía potencial elástica del rango definido.

**Partición del Conjunto de Datos y Estrategia de Validación** De las  $N = 100$  trayectorias generadas, se realizó una partición aleatoria para asegurar que la distribución estadística de las condiciones iniciales fuera homogénea entre los subconjuntos. El corpus de datos se dividió en: Entrenamiento (65 trayectorias), Validación (17 trayectorias) y Prueba (18 trayectorias).

Esta decisión de limitar el conjunto de datos a un número reducido de muestras responde a un criterio de **escalabilidad metodológica**. Si bien el sistema SDOF lineal permite generar miles de simulaciones con bajo costo computacional, el objetivo de esta tesis es validar una arquitectura eficiente en el uso de datos (*data-efficient*). Esto anticipa escenarios de aplicación futura donde los datos de entrenamiento provengan de modelos de Elementos Finitos de alta fidelidad o de ensayos experimentales, contextos donde el costo computacional o logístico hace inviable la generación masiva de datos (Big Data). Por tanto, se busca demostrar que el modelo puede generalizar correctamente en un régimen de datos escasos.

El rol de cada subconjunto se definió estrictamente para evitar sesgos en la evaluación:

- **Conjunto de Validación (17 %):** Se utilizó exclusivamente para el monitoreo de la función de pérdida durante el entrenamiento y para la selección del mejor modelo (*Model Checkpointing*). Es importante destacar que **no se realizó optimización de hiperparámetros** (como búsqueda de rejilla para tasa de aprendizaje o tamaño de red) sobre este conjunto. Se optó por mantener una arquitectura fija para evaluar la capacidad de aprendizaje intrínseca del integrador propuesto sin sobreajustar la configuración al caso lineal específico, garantizando así que la arquitectura sea robusta y generalizable a fases posteriores más complejas.
- **Conjunto de Prueba (18 %):** Estas trayectorias se mantuvieron completamente aisladas (ciegas) durante todo el proceso de optimización. Se utilizan única y exclusivamente en la etapa final para evaluar el desempeño del modelo en datos no vistos, proporcionando una métrica de error objetiva sobre la capacidad de generalización del integrador neuronal.

La distribución resultante de los 100 puntos de inicio utilizados para el entrenamiento, validación y prueba se visualiza en la Figura 4.1, cubriendo de manera uniforme los cuatro cuadrantes del plano de fase de interés.

**Pre-procesamiento y Normalización de Datos** El uso de variables físicas reales introduce un desafío numérico conocido como *desequilibrio de escalas* (*scale imbalance*). Mientras que los desplazamientos son del orden de  $10^{-2}$  m, las fuerzas restauradoras (producto de la rigidez  $k \approx 3.5 \times 10^6$ ) alcanzan magnitudes de  $10^5$  N. Esta disparidad de siete órdenes de magnitud dificulta la convergencia del descenso de gradiente y puede llevar a la saturación de las funciones de activación no-lineales.

Para mitigar esto, se aplicó una estandarización (*Z-score normalization*) a los vectores de estado antes de su ingreso a la red neuronal. A diferencia del escalamiento Min-Max, la estandarización preserva la naturaleza

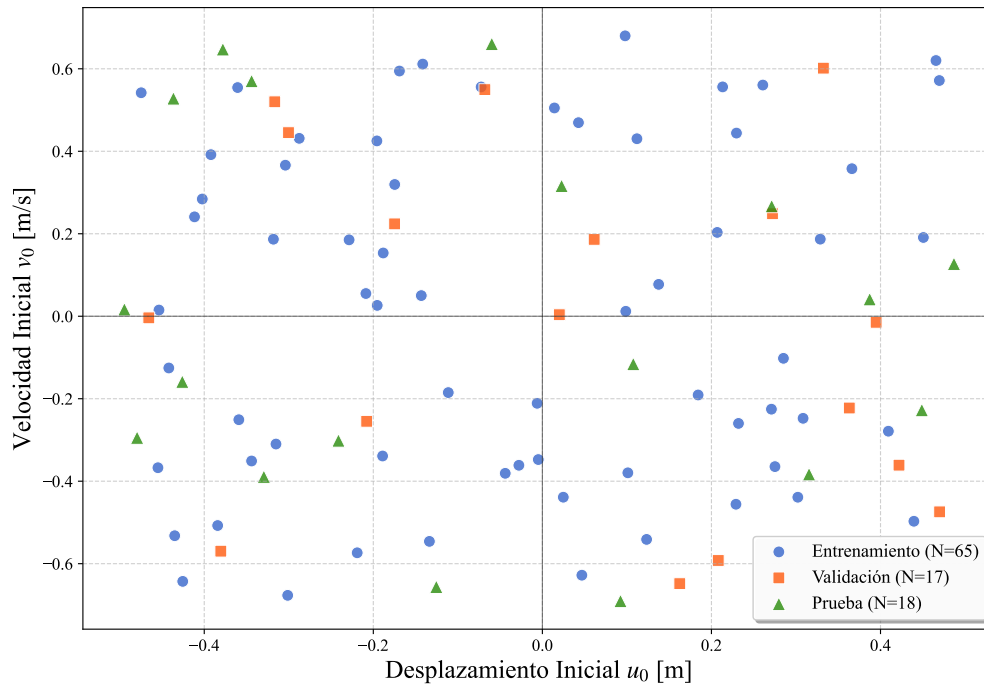


Figura 4.1: Distribución de las 100 condiciones iniciales aleatorias. Los rangos se han acotado a valores realistas de operación elástica ( $\pm 5$  cm de desplazamiento).

Gaussiana de las vibraciones alrededor del punto de equilibrio y es más robusta ante valores atípicos en la fase de prueba.

Cada componente del estado  $x_i$  (desplazamiento y velocidad) se transforma en una variable latente adimensional  $\hat{x}_i$ :

$$\hat{x}_i = \frac{x_i - \mu_i}{\sigma_i} \quad (4.18)$$

Donde  $\mu_i$  y  $\sigma_i$  son la media y la desviación estándar calculadas exclusivamente sobre el conjunto de entrenamiento, evitando así la fuga de información (*data leakage*) desde los conjuntos de validación o prueba.

**Recuperación de Variables Físicas (Des-normalización)** Dado que el entrenamiento se realiza en el espacio latente estandarizado  $\hat{x}$ , las predicciones deben transformarse nuevamente al espacio físico para su interpretación y evaluación de errores.

Para el vector de estado  $x = [u, v]^T$ , se aplica la transformación inversa estándar:

$$x_{pred} = \hat{x}_{pred} \odot \sigma + \mu \quad (4.19)$$

Donde  $\odot$  denota multiplicación elemento a elemento,  $\sigma$  es el vector de desviaciones estándar y  $\mu$  el vector de medias obtenidos en la etapa de pre-procesamiento.

Es fundamental notar que para la **aceleración**  $a_{pred}$ , que corresponde a la derivada temporal de la velocidad  $\dot{v}$ , la transformación difiere. Al derivar la expresión de normalización respecto al tiempo, el término

constante de la media ( $\mu_v$ ) se anula:

$$a_{pred}(t) = \frac{d}{dt}(\sigma_v \hat{v}(t) + \mu_v) = \sigma_v \cdot \dot{\hat{v}}_{pred}(t) \quad (4.20)$$

Por consiguiente, la aceleración física se recupera escalando la salida de la red únicamente por la desviación estándar de la velocidad ( $\sigma_v$ ), sin adición de la media. Esta distinción es crítica para verificar el cumplimiento de la ley de movimiento ( $F = ma$ ) en unidades físicas correctas.

### Metodología de Evaluación y Métricas de Desempeño

Para validar la capacidad del modelo no solo como un aproximador de funciones local, sino como un integrador numérico estable capaz de operar como un Gemelo Digital, es fundamental definir el régimen de simulación bajo el cual se calcularán los errores.

**Régimen de Simulación: Enfoque Autónomo** En la literatura de modelado dinámico con redes neuronales, existen dos modalidades principales de operación, las cuales cumplen roles distintos en las etapas de optimización y validación:

1. **Predicción a un paso (*Teacher Forcing*):** En cada instante  $t$ , la red recibe el estado *real* (medido)  $x_t$  para predecir  $x_{t+1}$ . **Este es el esquema adoptado durante la fase de entrenamiento**, ya que permite optimizar los parámetros de la red minimizando el error local de integración paso a paso, asegurando una convergencia estable del descenso de gradiente. Sin embargo, si se utiliza como métrica de desempeño final, este método enmascara problemas de estabilidad dinámica, ya que corrige artificialmente la trayectoria en cada paso con datos externos.
2. **Simulación Recursiva (Autónoma):** La red opera de manera independiente, desacoplada de los datos reales. Recibe únicamente la condición inicial  $x_0^{real}$ . Para todo  $t > 0$ , la entrada de la red es su propia predicción del paso anterior:

$$\hat{x}_{t+1} = \hat{x}_t + \mathcal{N}_\theta(\hat{x}_t)\Delta t \quad (4.21)$$

Dado que el objetivo central de este trabajo es desarrollar un integrador numérico capaz de extrapolar la respuesta estructural sin depender de valores conocidos continuos o sensores continuos para una aplicación en SHM o RTHS, se desestima la evaluación a un paso. El análisis de desempeño se restringirá **exclusivamente al régimen recursivo**, ya que es el único que permite evaluar la estabilidad a largo plazo, la conservación de energía y la resistencia al *drift* (acumulación de error numérico) del modelo propuesto.

**Validación de la Física Latente (Aceleración)** Dado que el vector de estado se define como  $x = [u, v]^T$ , la salida de la red neuronal  $\mathcal{N}_\theta(x)$  representa la derivada temporal  $[\dot{u}, \dot{v}]^T$ . Por definición física,  $\dot{v}$  corresponde a la aceleración  $a$ . Por lo tanto, es posible evaluar explícitamente si la red satisface la ecuación de movimiento ( $F = ma$ ) comparando la segunda mitad del vector de salida de la red ( $a_{pred}$ ) contra la aceleración real del sistema, sin necesidad de diferenciación numérica adicional.

**Métricas de Error** Para cuantificar la discrepancia entre las trayectorias reales autónomas ( $\mathbf{y}$ ) y las predichas recursivamente ( $\hat{\mathbf{y}}$ ), se utilizan dos indicadores:

- **Raíz del Error Cuadrático Medio (RMSE):** Medida absoluta de la desviación estándar de los residuales, útil para entender la magnitud física del error.
- **Error Relativo L2 (L2RE):** Métrica adimensional que normaliza el error respecto a la energía de la señal, facilitando la comparación entre variables con distintas magnitudes (desplazamiento vs. aceleración).

$$L2RE = \frac{\|\hat{\mathbf{y}} - \mathbf{y}\|_2}{\|\mathbf{y}\|_2} \quad (4.22)$$

## Resultados y Discusión del Experimento 1

Tras el entrenamiento de la red neuronal utilizando el conjunto de datos normalizados, se procedió a evaluar su capacidad de generalización en el conjunto de prueba de 18 trayectorias no vistas. A continuación, se analiza el desempeño del modelo EulerNet operando en régimen 100 % autónomo.

**Evaluación Cualitativa de la Dinámica** El primer criterio de validación es la capacidad del modelo para reproducir la morfología de la respuesta sísmica. La Figura 4.2 presenta la comparación de las historias de tiempo para una condición inicial aleatoria.

Se observa una coincidencia notable entre la solución numérica de referencia (línea gris) y la predicción recursiva de la red neuronal (línea discontinua). El modelo reproduce fielmente los dos parámetros fundamentales del sistema SDOF lineal:

1. **Frecuencia Natural:** La red captura el periodo  $T_n \approx 0.41$  s sin desfase temporal acumulado, indicando que la rigidez  $k$  ha sido aprendida correctamente.
2. **Amortiguamiento:** La tasa de decaimiento de la amplitud sigue la envolvente viscosa teórica ( $\xi \approx 2\%$ ), lo que confirma que el modelo es disipativo y numéricamente estable.

La red no solo predice el desplazamiento ( $u$ ), sino que infiere correctamente la velocidad ( $v$ ) y la aceleración ( $a$ ), esta última recuperada derivando la salida de la red en el espacio latente.

Para verificar la estabilidad asintótica global, se analiza el retrato de fase en la Figura 4.3. La trayectoria predicha describe una espiral que converge suavemente al origen  $(0, 0)$ . Esto demuestra geoméricamente que la red ha interiorizado las propiedades de estabilidad de Lyapunov del sistema físico, sin introducir energía espuria que pudiera inestabilizar la integración a largo plazo.

**Desempeño Cuantitativo Global** Para objetivar la precisión del integrador, la Tabla 4.2 resume las métricas de error promedio calculadas sobre todo el conjunto de prueba.

El modelo alcanza un Error Relativo (L2RE) inferior al 2 % tanto para desplazamiento como para velocidad. Es relevante destacar el comportamiento de la aceleración, cuyo error (7.51 %) es significativamente mayor que el error de estado. Este fenómeno es esperado en modelos basados en física: dado que la aceleración

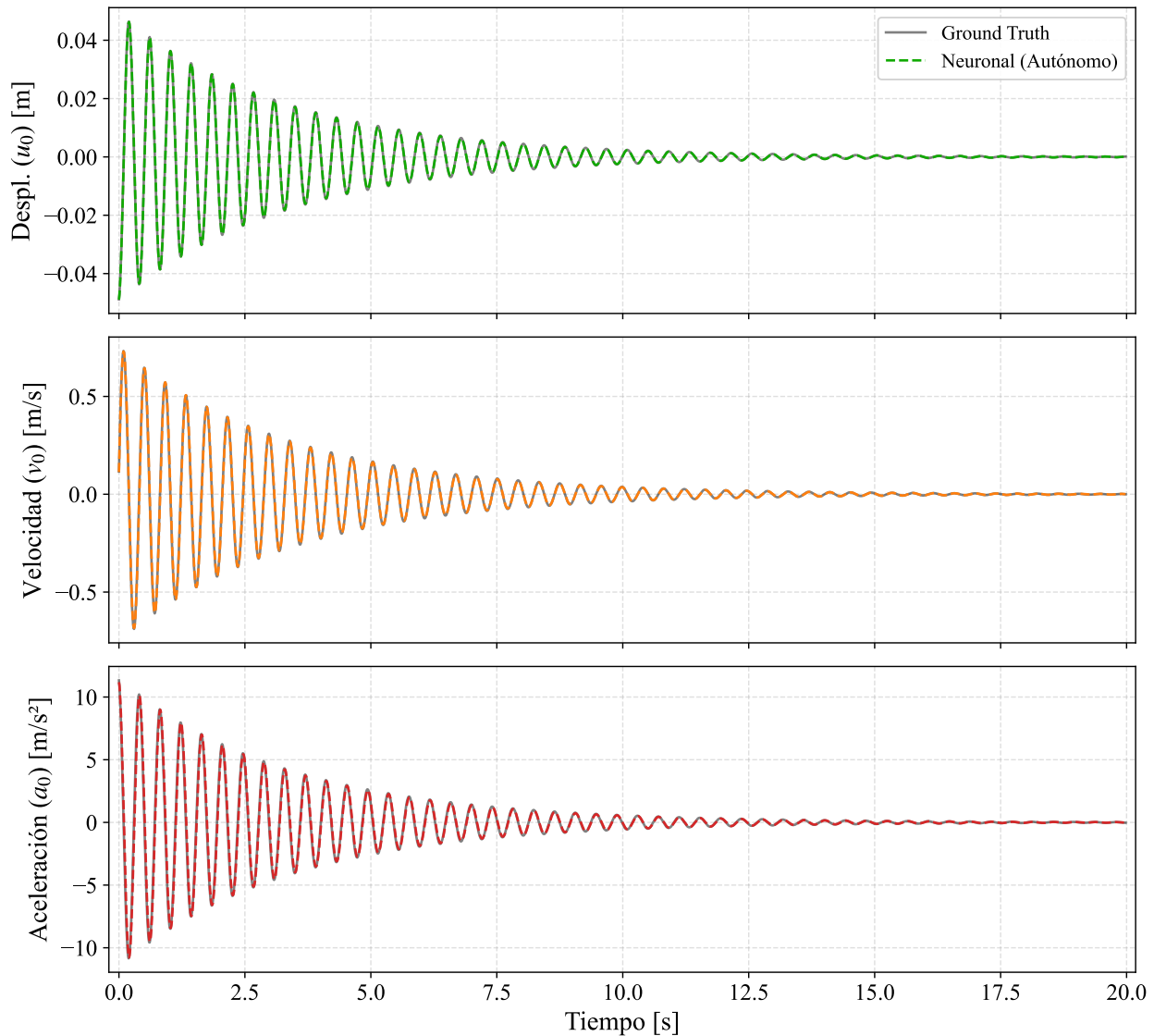


Figura 4.2: Evaluación de trayectoria temporal en simulación autónoma. La predicción neuronal (rojo/azul/magenta) se superpone casi perfectamente a la realidad (gris), validando la captura de la frecuencia y el amortiguamiento sin divergencia.

es proporcional a la segunda derivada del desplazamiento ( $a \propto -\omega^2 u$ ), cualquier ruido de alta frecuencia en la predicción del campo vectorial se amplifica. No obstante, un error inferior al 8% en la aceleración es altamente satisfactorio para una integración autónoma a 2000 pasos de tiempo.

**Análisis de Robustez y Dispersión** Finalmente, se evalúa la consistencia del modelo ante distintas condiciones iniciales mediante diagramas de caja (Figura 4.4).

La distribución de errores es compacta para el desplazamiento y la velocidad, lo que indica que el desempeño del modelo es robusto y no depende de la amplitud inicial de la vibración dentro del rango elástico estudiado. La aceleración presenta una dispersión levemente mayor, con algunos valores atípicos acercándose al

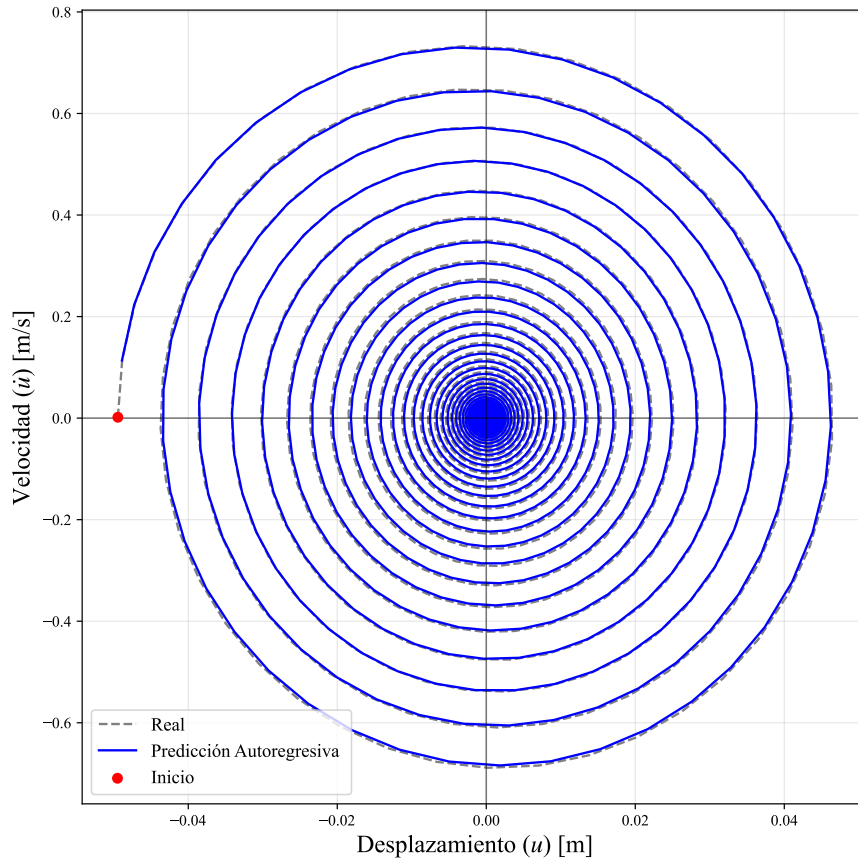


Figura 4.3: Retrato de fase ( $u$  vs.  $\dot{u}$ ) Fase I, Lineal. La coincidencia geométrica de las espirales confirma que el campo vectorial aprendido preserva la estructura disipativa del sistema.

Tabla 4.2: Resumen de Métricas de Desempeño

Métrica	Desplazamiento ( $u$ )	Velocidad ( $v$ )	Aceleración ( $a$ )
RMSE (Absoluto)	$1.06 \times 10^{-4}$ m	$1.64 \times 10^{-3}$ m/s	$1.08 \times 10^{-1}$ m/s <sup>2</sup>
L2RE (Relativo)	1.77 %	1.79 %	7.51 %
Max L2RE	3.64 %	3.75 %	8.48 %

10 % de error, comportamiento atribuible a la sensibilidad numérica en los puntos de inversión de movimiento donde la velocidad es nula y la aceleración máxima.

**Conclusión del Experimento 1:** Los resultados validan que la arquitectura EulerNet, combinada con la normalización de datos propuesta, es capaz de aprender la física de un sistema estructural lineal realista. El modelo actúa como un *gemelo digital* estable, capaz de predecir la respuesta completa a partir de una única condición inicial. Habiendo establecido esta línea base de desempeño en régimen lineal, la siguiente sección desafía al modelo introduciendo no-linealidades geométricas mediante el oscilador de Duffing.

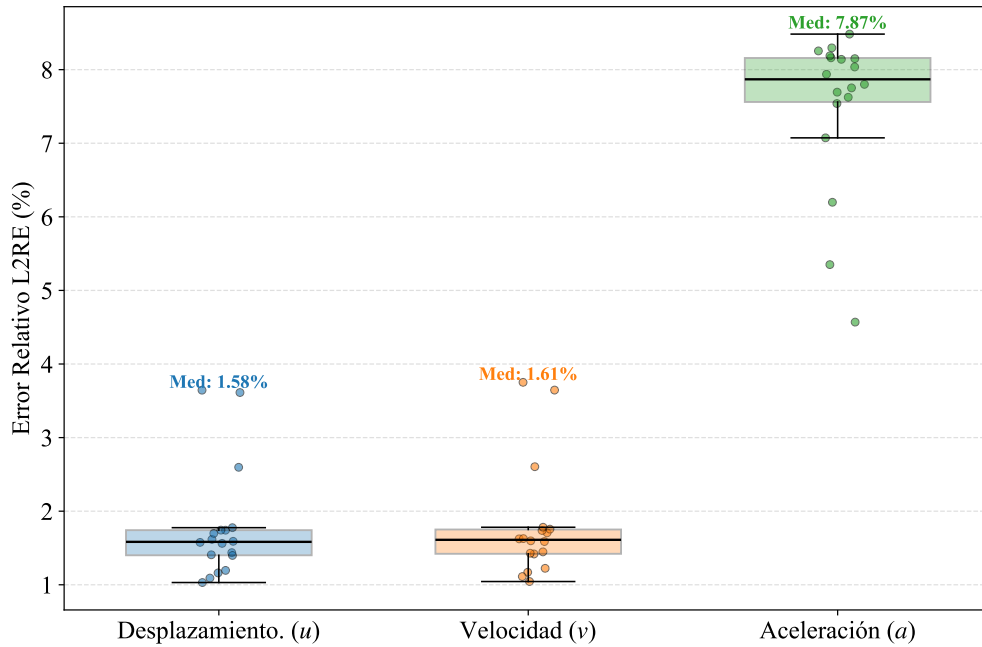


Figura 4.4: Dispersión del error L2RE para las 18 trayectorias de prueba. La consistencia en  $u$  y  $v$  confirma la fiabilidad del modelo ante diversas condiciones iniciales.

#### 4.3.5. Experimento 2: Generalización a No-linealidad Geométrica (Duffing)

Una vez validada la arquitectura en el régimen lineal, se procede a evaluar la capacidad del modelo para capturar dinámicas no-lineales. Para evaluar la capacidad del modelo ante estas topologías variables, se introdujo una rigidez cúbica  $k_3 = 2.0 \times 10^9 \text{ N/m}^3$  (Oscilador de Duffing), calibrada para generar un endurecimiento significativo ( $F_{nl} \approx 140\% F_{lin}$  a 5 cm).

#### Formulación Matemática y Espacio de Estados

La ecuación de movimiento que gobierna el sistema se caracteriza por la adición de un término de restauración cúbico a la rigidez elástica. La dinámica viene dada por:

$$m\ddot{u}(t) + c\dot{u}(t) + k_1u(t) + k_3u(t)^3 = 0 \quad (4.23)$$

Donde  $k_1$  representa la rigidez lineal elástica y  $k_3$  el coeficiente de rigidez no-lineal. Para formular el problema como una entrada apta para la red neuronal, se reduce el sistema a su representación en el espacio de estados  $\mathbf{x} = [u, v]^T$ :

$$\frac{d}{dt} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} v \\ -\frac{1}{m}(cv + k_1u + k_3u^3) \end{bmatrix} \quad (4.24)$$

Esta formulación permite definir el problema de aprendizaje supervisado, donde la red neuronal parametrizada por  $\theta$ , denotada como  $\mathcal{N}_\theta$ , se entrena para aproximar la función derivada temporal del estado. Matemáticamente, se busca que la red aprenda el mapeo:

$$\mathcal{N}_\theta(\mathbf{x}_t) \approx \frac{d\mathbf{x}}{dt} = \mathbf{F}(\mathbf{x}_t) \quad (4.25)$$

Sustituyendo la dinámica del oscilador de Duffing, la red neuronal debe ser capaz de inferir implícitamente la siguiente estructura vectorial a partir de los datos, sin conocer los parámetros  $m, c, k_1, k_3$  *a priori*:

$$\mathcal{N}_\theta \left( \begin{bmatrix} u \\ v \end{bmatrix} \right) \rightarrow \begin{bmatrix} v \\ -\frac{c}{m}v - \frac{k_1}{m}u - \frac{k_3}{m}u^3 \end{bmatrix} \quad (4.26)$$

De esta forma, la primera salida de la red ( $\dot{u}_{pred}$ ) debe coincidir con la entrada de velocidad ( $v$ ), imponiendo una restricción cinemática, mientras que la segunda salida ( $\dot{v}_{pred}$ ) debe capturar la aceleración resultante del equilibrio de fuerzas lineales y no-lineales.

A diferencia del experimento anterior, la matriz del sistema (Jacobiano) ya no es constante, sino que depende del cuadrado del desplazamiento ( $u^2$ ). Esto obliga a la red neuronal a aprender una función de transición de estados que varía no-linealmente a través del dominio, adaptando la rigidez tangente instantánea en cada paso de tiempo.

### Configuración del Sistema y Datos

Se mantiene la configuración base del marco de acero del experimento 1 ( $m = 15$  ton,  $k_1 = 3500$  kN/m) para asegurar la continuidad del estudio. Sin embargo, para evaluar el desempeño del modelo en régimen no-lineal, se ha introducido una rigidez cúbica  $k_3$  calibrada agresivamente para inducir un comportamiento de endurecimiento severo.

El valor de  $k_3$  se seleccionó bajo un criterio de estrés medio: se buscó que la fuerza no-lineal representara aproximadamente el 50 % de la fuerza lineal a una amplitud de desplazamiento típica de 3 cm (0.03 m), y no solo en los extremos máximos. Esto asegura que la red neuronal enfrente una dinámica no-lineal significativa durante la mayor parte de la fase de decaimiento libre.

### Continuidad Metodológica

Es importante destacar que *la arquitectura de la red neuronal y la estrategia de pre-procesamiento se mantienen idénticas* al experimento anterior:

- **Arquitectura:** Perceptrón Multicapa (MLP) con funciones de activación tanh.
- **Normalización:** Se utiliza ‘StandardScaler’ ajustado a los nuevos datos de entrenamiento para mitigar el desequilibrio de escalas, manteniendo la lógica de recuperación de variables físicas y cálculo de aceleración descrita anteriormente.

Esta decisión metodológica busca demostrar que la arquitectura propuesta es agnóstica a la física subyacente y capaz de generalizar a sistemas no-lineales sin requerir modificaciones en sus hiperparámetros (fine-tuning).

Tabla 4.3: Parámetros del Oscilador de Duffing (Escala Estructural Real).

Parámetro	Símbolo	Valor
<i>Propiedades Base (Fase I)</i>		
Masa	$m$	15,000 kg
Amortiguamiento	$c$	9,000 Ns/m
Rigidez Lineal	$k_1$	$3.5 \times 10^6$ N/m
<i>Parámetro no-lineal</i>		
Rigidez Cúbica	$k_3$	$2.0 \times 10^9$ N/m <sup>3</sup>
<i>Grado de No-Linealidad</i>		
Ratio de Fuerza ( $u = 3$ cm)	$F_{nl}/F_{lin}$	$\approx 51$ %
Ratio de Fuerza ( $u = 5$ cm)	$F_{nl}/F_{lin}$	$\approx 142$ %

**Condiciones Iniciales y Rango de Operación** Dado que el efecto del término cúbico  $k_3u^3$  se magnifica con el desplazamiento, se han seleccionado los mismos rangos de muestreo que en el experimento anterior los cuales aseguran la exploración de regiones con fuerte interacción no-lineal:

$$u_0 \sim \mathcal{U}(-0.05, 0.05) \text{ m}, \quad v_0 \sim \mathcal{U}(-0.7, 0.7) \text{ m/s} \quad (4.27)$$

**Nota sobre la Magnitud de Fuerzas:** Para el desplazamiento máximo considerado ( $u = 0.05$  m), la fuerza restauradora lineal es  $k_1u \approx 175$  kN, mientras que la componente no-lineal alcanza  $k_3u^3 \approx 250$  kN. Al ser la fuerza no-lineal un 142 % de la lineal, el sistema se aleja drásticamente del oscilador armónico simple, presentando un desafío de estabilidad significativo para el integrador neuronal.

## Resultados y Discusión del Experimento 2

Tras el entrenamiento con datos de naturaleza no lineal ( $k_3 = 2.0 \times 10^9$  N/m<sup>3</sup>), se evaluó el desempeño del modelo sobre un conjunto de prueba compuesto por 18 trayectorias inéditas, operando bajo un régimen de simulación 100 % autónomo (autorregresivo).

**Evaluación Cualitativa de la Dinámica No Lineal** La validación crítica de este experimento radica en determinar si la arquitectura EulerNet es capaz de capturar el fenómeno de *endurecimiento por deformación* (*strain hardening*). Como se observa en la Figura 4.5, la red reproduce con alta fidelidad la respuesta temporal del oscilador de Duffing, capturando la modulación de frecuencia intrínseca a este régimen. A diferencia de un sistema lineal, donde la frecuencia natural permanece constante, en este caso la red debe adaptarse a una rigidez tangente dependiente del estado, lo que incrementa significativamente la complejidad de la predicción autónoma a largo plazo.

Esta precisión se manifiesta también en el espacio de fase (Figura 4.6). La trayectoria predicha converge de forma consistente con la geometría no lineal de la órbita real, preservando la estabilidad hacia el equilibrio sin introducir artefactos numéricos o inestabilidades visibles. Mientras que en el caso lineal el retrato de

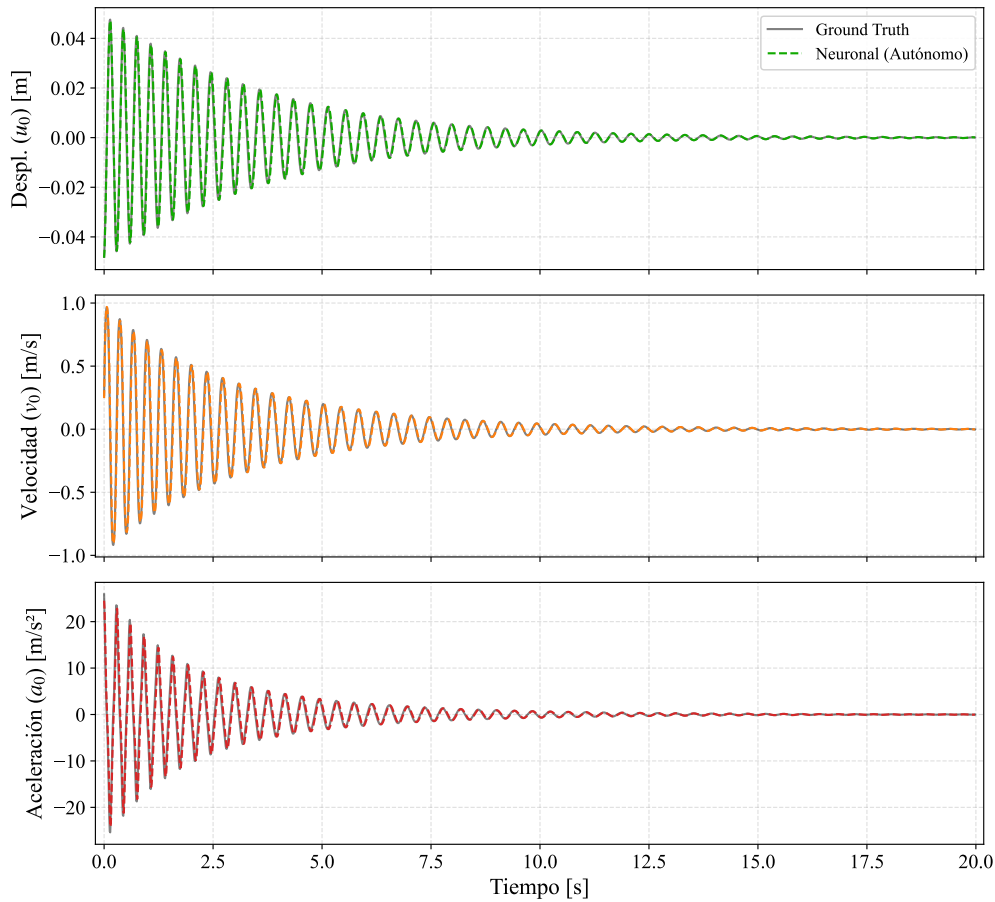


Figura 4.5: Comparación de trayectorias en el sistema Duffing. La red sigue con precisión los cambios de frecuencia y fase, capturando la dependencia entre rigidez y amplitud instantánea.

fase sigue una topología de espiral invariante, en el sistema de Duffing la curvatura de la trayectoria varía dinámicamente con la amplitud, una característica que la red logra generalizar con éxito.

El análisis detallado revela que el modelo predice correctamente la evolución de la frecuencia en función de la energía del sistema:

- **Fase de alta energía ( $t < 5$  s):** En los ciclos iniciales, donde los desplazamientos son máximos, la red reproduce las oscilaciones de alta frecuencia inducidas por el incremento en la rigidez tangente, definida por  $k_t \approx k_1 + 3k_3u^2$ .
- **Transición al régimen lineal:** A medida que el amortiguamiento disipa la energía, el modelo ajusta suavemente la respuesta hacia la frecuencia natural del sistema base, demostrando una transición fluida entre regímenes dinámicos.

La coincidencia de fase entre la predicción (línea discontinua) y la referencia confirma que EulerNet ha interiorizado la topología variable del campo vectorial, manteniendo la sincronía temporal incluso ante variaciones de amplitud significativas.

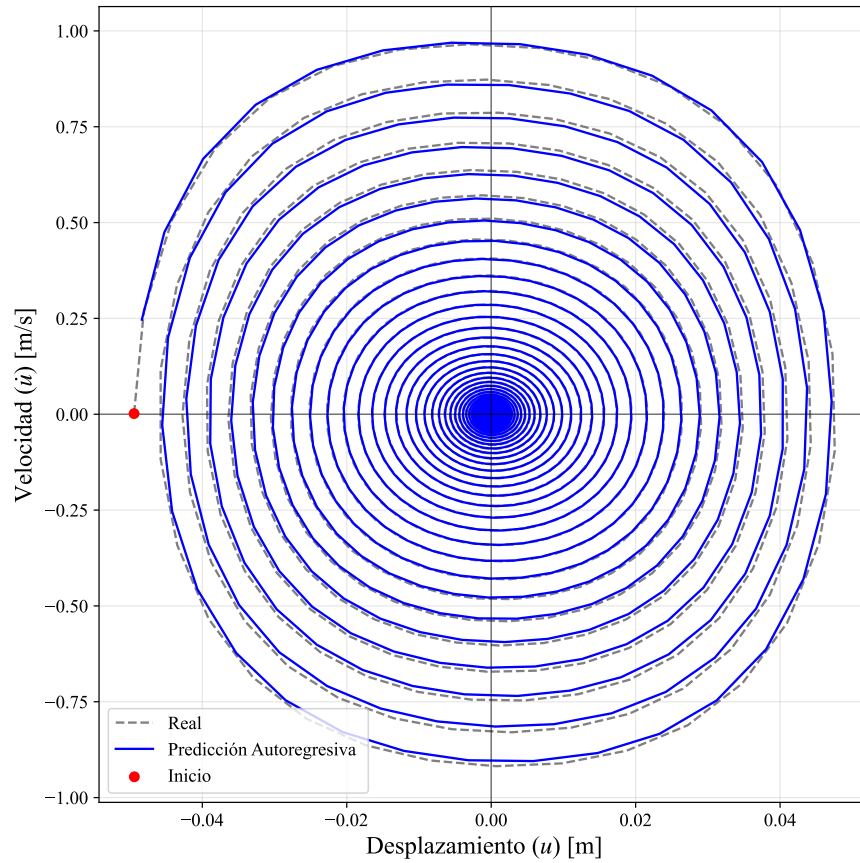


Figura 4.6: Retrato de fase ( $u$  vs.  $\dot{u}$ ) del sistema Duffing. La predicción autorregresiva reproduce la geometría no lineal de la trayectoria y su convergencia al equilibrio.

**Desempeño Cuantitativo** La Tabla 4.4 resume las métricas de error promedio. Se reporta un Error Relativo (L2RE) del 5.74 % para el desplazamiento.

Al comparar estos valores con el Experimento 1 (Lineal: 1.77 %), se evidencia un incremento moderado en el error. Este comportamiento es esperable y físico: la red debe aproximar gradientes mucho más pronunciados introducidos por el término cúbico. Sin embargo, el hecho de que el error se mantenga en el orden del 5 % confirma que la arquitectura EulerNet posee la capacidad de representación suficiente para modelar no-linealidades geométricas suaves sin colapsar.

Tabla 4.4: Resumen de Métricas de Desempeño (Promedio en Test - Fase II)

Métrica	Desplazamiento ( $u$ )	Velocidad ( $v$ )	Aceleración ( $a$ )
RMSE (Absoluto)	$4.48 \times 10^{-4}$ m	$7.60 \times 10^{-3}$ m/s	$2.27 \times 10^{-1}$ m/s <sup>2</sup>
L2RE (Relativo)	5.74 %	5.43 %	10.82 %
Max L2RE	19.06 %	18.39 %	15.15 %

**Análisis de Robustez en el Régimen No Lineal** Para verificar la consistencia del modelo en el escenario no lineal, se analizó la dispersión del error L2RE mediante diagramas de caja (Figura 4.7). Como era de esperar, la complejidad intrínseca del sistema Duffing conlleva un incremento en las métricas de error respecto al caso lineal; sin embargo, los valores medianos para el desplazamiento (4.43 %) y la velocidad (4.40 %) se mantienen notablemente bajos para una predicción autorregresiva de larga duración.

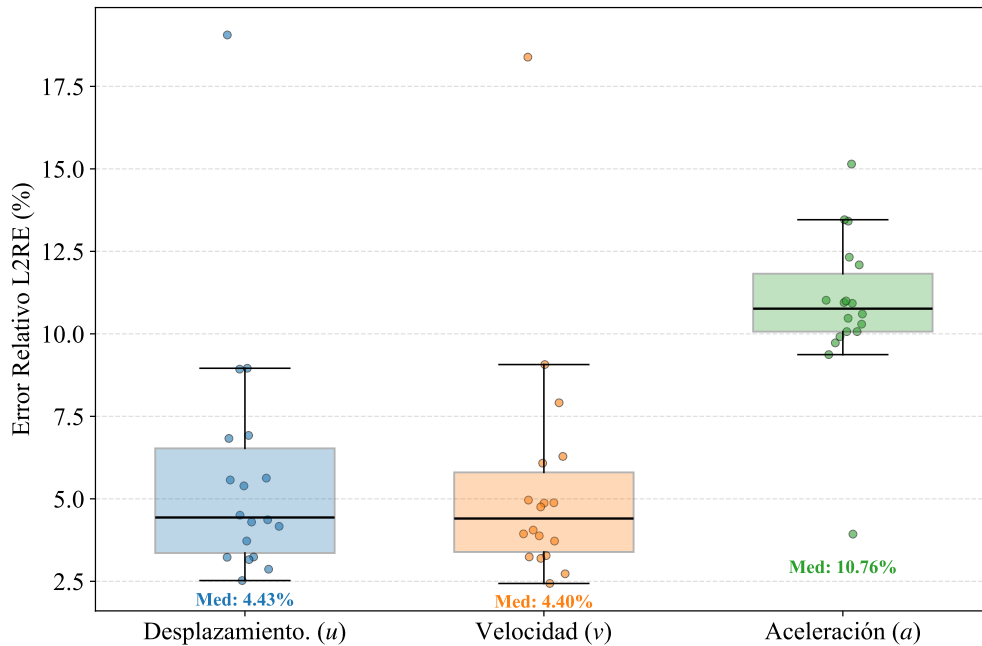


Figura 4.7: Distribución del error L2RE para el sistema Duffing. A pesar del aumento en la complejidad dinámica, la red mantiene una dispersión controlada y errores medianos inferiores al 5 % en variables de estado

Al comparar con el sistema lineal (donde el error en  $u$  fue de 1.77 %), se observa que la red conserva una robustez significativa ante diversas condiciones iniciales, sin presentar explosiones de error incluso en presencia de fuerzas restauradoras cúbicas. La aceleración continúa siendo la variable con mayor dispersión (mediana de 10.76 %), lo cual es consistente con la mayor demanda computacional que implica capturar las variaciones bruscas de la rigidez tangente en los puntos de máxima amplitud.

**Validación Mecánica: Ciclos de Fuerza-Desplazamiento** Para verificar si la predicción cinemática ( $\hat{u}$ ) es consistente con la mecánica subyacente, se reconstruyó la fuerza restauradora del sistema. Es importante aclarar que la red neuronal predice estados cinemáticos, no fuerzas directamente. Por tanto, la fuerza visualizada en la Figura 4.8 es una **reconstrucción post-proceso** calculada introduciendo el desplazamiento predicho recursivamente ( $\hat{u}$ ) en la ecuación constitutiva real:

$$F_{rec} = k_1 \hat{u} + k_3 \hat{u}^3 \tag{4.28}$$

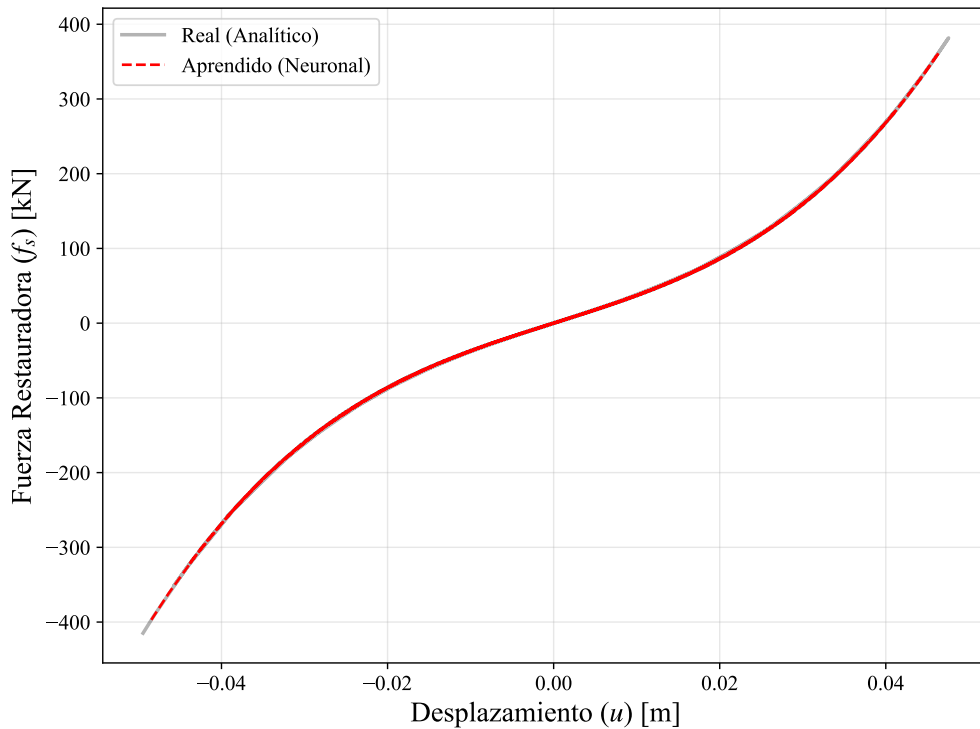


Figura 4.8: Comparación de la relación fuerza-desplazamiento del oscilador de Duffing entre la solución analítica y la reconstrucción basada en la predicción neuronal.

El gráfico revela la característica curva en forma de S del oscilador de Duffing. La superposición de la curva reconstruida (rojo) sobre la real (gris) demuestra que los desplazamientos predichos por la red recorren correctamente el camino de equilibrio no-lineal. No se observan oscilaciones espurias ni violaciones significativas de la rigidez tangente, confirmando que la red no solo memorizó una forma de onda, sino que aprendió la ley de fuerza implícita.

**Influencia de la Amplitud en la Precisión** Para profundizar en el origen de las desviaciones, se analizó la correlación entre el desplazamiento máximo real ( $|u|_{max}$ ) y el error relativo L2RE (Figura 4.9). Los resultados revelan una tendencia cuadrática clara ( $R^2 = 0.618$ ), donde el error se incrementa significativamente a medida que aumenta la amplitud de la oscilación.

Este comportamiento es físicamente consistente con la naturaleza del oscilador de Duffing: a mayores desplazamientos, el término de rigidez cúbica cobra mayor relevancia, introduciendo variaciones de frecuencia más abruptas que exigen una mayor precisión en el paso de integración aprendido por la red. Mientras que en amplitudes moderadas ( $\approx 0.02$  m) el modelo es extremadamente preciso, el aumento del error hacia los 0.05 m marca el límite de generalización del modelo ante condiciones de alta no linealidad.

**Conclusión del Experimento 2:** Los resultados demuestran que EulerNet es capaz de generalizar a sistemas con rigidez variable, capturando correctamente la física del endurecimiento. Sin embargo, la correlación entre

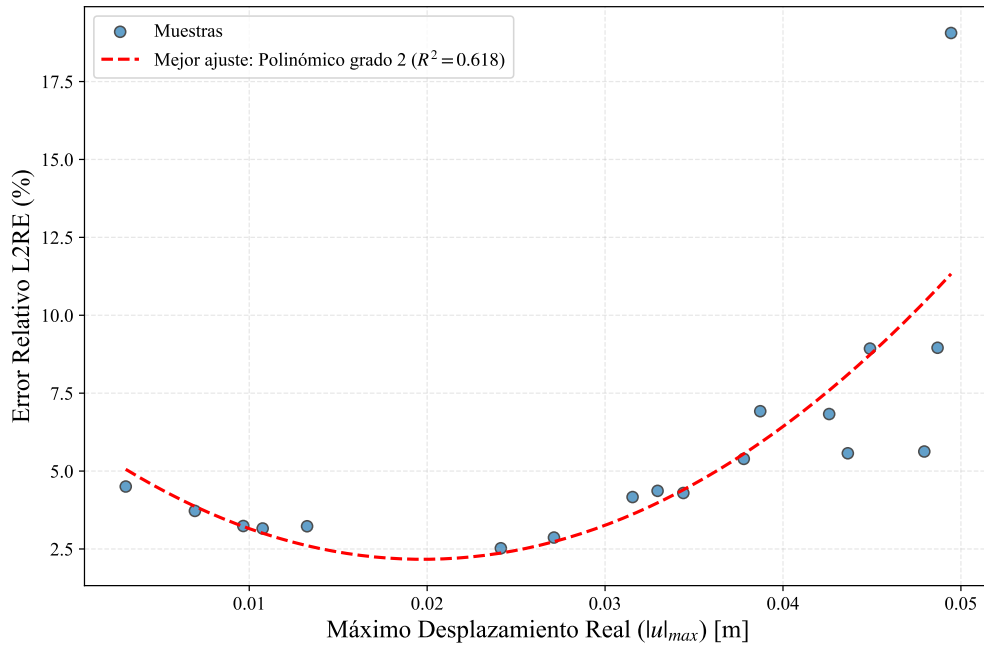


Figura 4.9: Correlación Error vs. Amplitud Máxima. El aumento del error proporcional a la deformación identifica a las zonas de alta rigidez no-lineal como la fuente principal de inexactitud del integrador.

error y amplitud advierte una limitación: a medida que la no-linealidad se vuelve más severa, la precisión del integrador de primer orden se degrada. Esto motiva la necesidad de investigar estrategias más robustas (Fase II y III) para sistemas donde la no-linealidad no es suave, como en el caso de la histéresis severa.

#### 4.3.6. Discusión General y Conclusiones de la Fase I

La ejecución de los Experimentos 1 (Lineal) y 2 (Duffing) permite establecer que la arquitectura propuesta, EulerNet, constituye un integrador numérico autónomo válido para sistemas dinámicos con no-linealidades suaves y conservativas. Los resultados obtenidos validan la hipótesis de que es posible aprender la topología de un sistema físico a partir de datos y utilizarla para realizar predicciones recursivas estables.

A partir de la evidencia presentada, se derivan las siguientes conclusiones:

1. Más allá de memorizar trayectorias, el modelo demostró capacidad para aprender leyes constitutivas complejas. En el caso del oscilador de Duffing, la red ajustó correctamente la frecuencia instantánea de oscilación en respuesta a la amplitud, lo que implica que la red neuronal ha aprendido con éxito una función de rigidez tangente dependiente del estado ( $k_t(\mathbf{x})$ ), y no solo una rigidez promedio constante.
2. Si bien se observó un incremento moderado en el error del sistema Duffing (5.74 %) respecto al lineal (1.77 %), este comportamiento es consistente con la naturaleza espectral del problema. Debido al endurecimiento por deformación, el sistema Duffing oscila a frecuencias significativamente más altas durante la fase inicial de alta energía. Para un esquema de integración con paso de tiempo fijo ( $\Delta t$ ), el aumento de la frecuencia de la señal reduce la resolución temporal efectiva (menos puntos por ciclo), lo

que explica naturalmente la leve degradación en la precisión sin invalidar la capacidad de aprendizaje del modelo.

3. En ambos casos experimentales, las simulaciones autónomas mostraron una estabilidad asintótica robusta, convergiendo al equilibrio sin presentar divergencias ni oscilaciones espurias. Esto sugiere que el operador de evolución temporal aprendido por la red respeta implícitamente las condiciones de estabilidad de Lyapunov y la disipación de energía del sistema físico.

Habiendo confirmado que EulerNet es una herramienta eficaz y estable para dinámicas suaves donde la rigidez varía gradualmente, el siguiente paso lógico es evaluar los límites de esta arquitectura. La siguiente fase de la investigación someterá al modelo a escenarios de **alta complejidad estructural**, caracterizados por cambios abruptos de rigidez (histéresis severa) y comportamiento caótico, para determinar si la simplicidad del esquema de Euler sigue siendo suficiente o si se requiere transicionar hacia integradores de orden superior.

#### 4.4. Fase II: Diagnóstico de Limitaciones en Sistemas Rígidos y Caóticos

Habiendo demostrado que la arquitectura EulerNet es competente para modelar dinámicas suaves y conservativas, esta fase tiene como objetivo someter al modelo a condiciones de demanda extrema. En la ingeniería sísmica real, las estructuras rara vez operan en regímenes suaves; por el contrario, exhiben comportamientos de histéresis severa, degradación de rigidez abrupta y, en ciertos casos, dinámicas caóticas (bi-estabilidad).

El objetivo de esta fase trasciende la obtención de un modelo perfecto; busca sistemáticamente **identificar el punto de falla** de la integración de primer orden. Se busca responder una pregunta fundamental: *¿Cuándo el modelo falla en sistemas complejos, ¿es porque la red neuronal no pudo aprender la función, o porque el integrador numérico es insuficiente?*

##### 4.4.1. Sistemas de Prueba de Alta Complejidad

Para este diagnóstico, se seleccionaron dos sistemas que desafían las suposiciones de suavidad de la Fase I.

##### 4.4.2. Caso A: Oscilador de Duffing de Doble Pozo (Bi-estable)

El primer desafío de la Fase II consiste en un sistema con inestabilidad estática en el origen, configurado mediante un potencial de doble pozo (*double-well potential*).

Es fundamental establecer que, a diferencia de los experimentos de la Fase I, esta configuración **no busca representar un sistema estructural convencional** ni una patología sísmica típica. Se trata de un **banco de pruebas algorítmico** diseñado para llevar al modelo a su límite operacional. El objetivo es forzar al integrador neuronal a navegar una topología con puntos de silla inestables y atractores múltiples, una tarea que desafía la estabilidad de cualquier esquema numérico explícito.

### Formulación Matemática y Espacio de Estados

La ecuación de movimiento se modifica invirtiendo el signo de la rigidez lineal ( $k_1 < 0$ ). La dinámica está gobernada por:

$$m\ddot{u}(t) + c\dot{u}(t) - \underbrace{|k_1|u(t) + k_3u(t)^3}_{f_s(u)} = 0 \tag{4.29}$$

Esto genera tres puntos de equilibrio: uno inestable en el origen ( $u = 0$ ) y dos estables en  $u = \pm\sqrt{|k_1|/k_3}$ . En el espacio de estados  $x = [u, v]^T$ , el campo vectorial presenta una topología compleja con un punto de silla (*saddle point*) en el centro y dos atractores espirales.

El desafío de aprendizaje supervisado para la red  $\mathcal{N}_\theta$  se define como la aproximación del mapeo:

$$\mathcal{N}_\theta \left( \begin{bmatrix} u \\ v \end{bmatrix} \right) \rightarrow \begin{bmatrix} v \\ -\frac{c}{m}v + \frac{|k_1|}{m}u - \frac{k_3}{m}u^3 \end{bmatrix} \tag{4.30}$$

La red debe aprender a modelar la bifurcación de trayectorias: dependiendo de la energía cinética instantánea, el sistema puede elegir caer en el pozo positivo o negativo, o cruzar entre ellos (caos transitorio).

### Configuración del Sistema

Para mantener la comparabilidad dimensional con los estudios previos, se utilizaron las propiedades de masa y amortiguamiento del marco de acero original ( $m = 15$  ton,  $c = 9000$  Ns/m). Se invirtió el signo de la rigidez lineal ( $k_1 = -3500$  kN/m) y se calculó la rigidez cúbica  $k_3$  necesaria para situar los puntos de equilibrio estable en  $\pm 3$  cm de desplazamiento ( $u_{eq} = 0.03$  m).

Tabla 4.5: Parámetros del Sistema de Doble Pozo (2EP).

Parámetro	Símbolo	Valor
Masa	$m$	15,000 kg
Amortiguamiento	$c$	9,000 Ns/m
Rigidez Lineal (Inestable)	$k_1$	$-3.5 \times 10^6$ N/m
Rigidez Cúbica	$k_3$	$3.89 \times 10^9$ N/m <sup>3</sup>
<i>Puntos de Equilibrio</i>		
Inestable (Silla)	$u_{eq}^0$	0.0 m
Estables (Focos)	$u_{eq}^\pm$	$\pm 0.03$ m

### Continuidad Metodológica

Para aislar el efecto de la complejidad dinámica, se mantuvieron rigurosamente las mismas condiciones experimentales utilizadas en la Fase I:

- **Protocolo de Datos:** Se generaron  $N = 100$  trayectorias con las mismas condiciones iniciales estocásticas ( $u_0 \in \pm 5$  cm,  $v_0 \in \pm 0.7$  m/s) y la misma partición (65/17/18 para entrenamiento, validación y prueba).
- **Arquitectura:** Se utilizó la misma red MLP con normalización Z-score.

Esta consistencia asegura que cualquier degradación en el desempeño sea atribuible exclusivamente a la incapacidad del esquema de integración EulerNet para manejar la dinámica caótica del sistema, y no a variaciones en los hiperparámetros o en la distribución de los datos.

### Resultados y Discusión del Caso A (Duffing Doble Pozo)

Tras el entrenamiento del modelo con los parámetros de rigidez negativa ( $k_1 = -3.5 \times 10^6$  N/m), se evaluó su desempeño en el conjunto de prueba bajo régimen 100 % autónomo. A diferencia de los casos anteriores, este sistema presenta una dinámica no trivial donde la precisión numérica es crítica para determinar la estabilidad global.

**Desempeño Cuantitativo: El Costo de la Inestabilidad** La Tabla 4.6 presenta el resumen de las métricas de error. Se observa una degradación drástica en el rendimiento en comparación con la Fase I. El Error Relativo (L2RE) promedio en desplazamiento asciende al **25.6 %**, con un máximo alarmante del **195.6 %**.

Tabla 4.6: Resumen de Métricas de Desempeño (Promedio en Test - Duffing Doble Pozo)

Métrica	Desplazamiento ( $u$ )	Velocidad ( $v$ )	Aceleración ( $a$ )
RMSE (Absoluto)	$7.51 \times 10^{-3}$ m	$3.39 \times 10^{-2}$ m/s	$7.57 \times 10^{-1}$ m/s <sup>2</sup>
L2RE (Relativo)	25.61 %	40.68 %	49.48 %
Max L2RE	195.68 %	122.90 %	130.02 %

Estos valores indican que el modelo ya no actúa como un predictor confiable para la totalidad del dominio de prueba. Un error cercano al 200 % en desplazamiento implica físicamente que la predicción está en antifase o, en este contexto bi-estable, que el modelo numérico se encuentra en el pozo de potencial opuesto al sistema real.

**Análisis de Robustez y Bifurcación** Como se observa en la Figura 4.10, este sistema representa el mayor desafío predictivo para la EulerNet: si bien la mediana del error en el desplazamiento se mantiene en un notable 2.72 %, lo que sugiere que el modelo identifica correctamente la permanencia de la trayectoria dentro de los pozos de potencial, las variables de velocidad y aceleración experimentan un incremento drástico en su error relativo, con medianas de 26.80 % y 31.31 % respectivamente.

Este fenómeno es intrínseco a la extrema sensibilidad del sistema ante las condiciones iniciales; pequeños desajustes temporales en la predicción del instante exacto del salto entre pozos resultan en errores de velocidad muy elevados, incluso cuando la morfología de la trayectoria en el espacio de configuración es cualitativamente correcta. La aparición de valores atípicos que alcanzan casi el 200 % de error subraya que, aunque la red

captura la topología del campo vectorial, la acumulación de error en simulaciones autorregresivas de larga duración encuentra su límite en regímenes que rozan el comportamiento caótico.

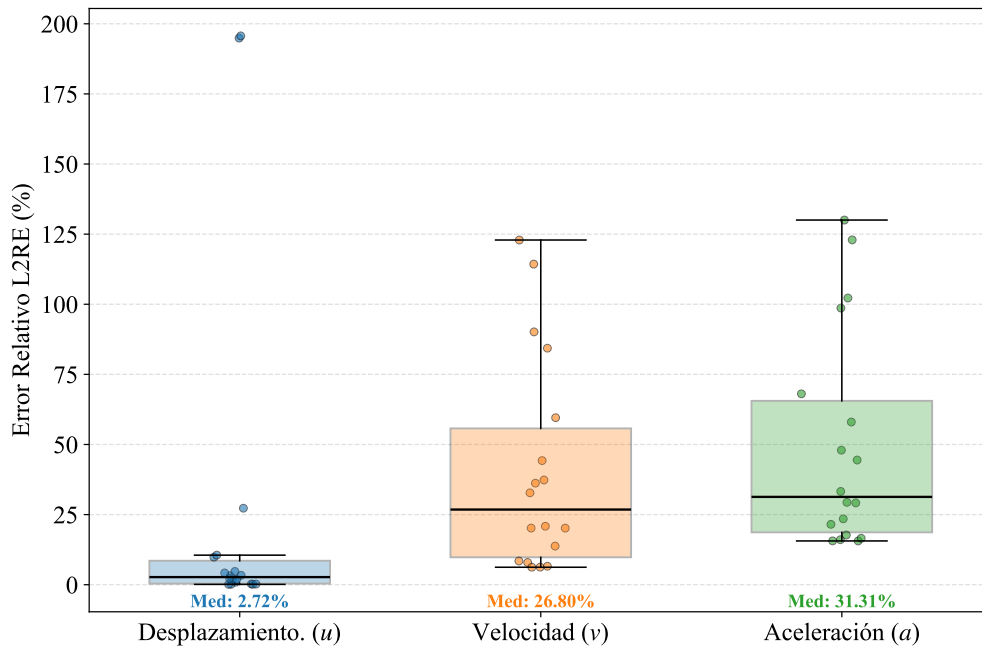


Figura 4.10: Dispersión del Error Relativo. Nótese la presencia de valores atípicos extremos (cerca al 200 %), indicativos de una falla cualitativa en la predicción de la trayectoria (divergencia de pozo).

**Evaluación Cualitativa de la Dinámica** Para comprender el origen de los errores elevados en este régimen, es imperativo analizar la evolución temporal de una trayectoria con comportamiento divergente. La Figura 4.11 ilustra la respuesta de la red frente al sistema de referencia bajo una condición inicial crítica.

Se observa que, tras el transitorio inicial ( $t < 2.5$  s), el sistema real (gris) pierde la energía suficiente para saltar entre pozos y queda confinado oscilando en torno al equilibrio positivo ( $u \approx 0.03$  m). En contraste, la red neuronal (verde) no logra replicar esta captura con precisión temporal: aunque eventualmente también queda confinada en el mismo pozo, presenta un error de fase acumulado significativo y amplitudes de oscilación ligeramente desajustadas. Este fenómeno es una manifestación de la extrema sensibilidad a las condiciones iniciales (o error de truncamiento del integrador aprendido) cerca del punto de silla ( $u = 0$ ).

Complementariamente, la Figura 4.12 muestra el retrato de fase para esta misma trayectoria problemática. Se observa cómo el integrador neuronal (línea azul) logra capturar la forma general de la órbita dentro del pozo derecho ( $u > 0$ ). Sin embargo, existe una discrepancia notable en la disipación de energía: la predicción tiende a estabilizarse o decaer mucho más rápido que el sistema real (Ground Truth, línea gris discontinua).

Esta falta de precisión en el amortiguamiento efectivo cerca del punto de silla ( $u \approx 0$ ) es crítica. En sistemas caóticos, una pequeña diferencia energética determina si la partícula tiene suficiente impulso para escapar del pozo.

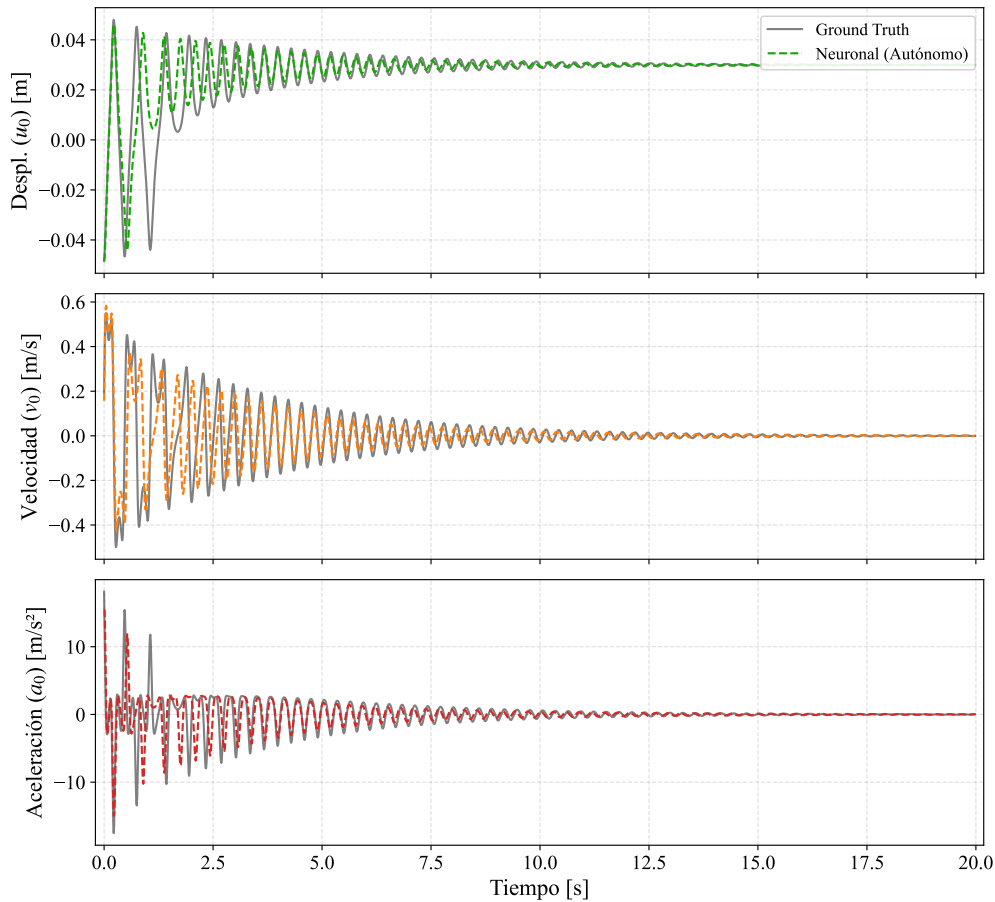


Figura 4.11: Historia de tiempo en el sistema de doble pozo. Se aprecia la divergencia en fase tras los primeros ciclos de alta energía, donde la red lucha por seguir la tasa de disipación y la captura final en el pozo de potencial.

**Diagnóstico de la Causa Raíz** Finalmente, la Figura 4.13 correlaciona el error con la amplitud máxima del desplazamiento. El gráfico confirma la hipótesis de la separatriz:

- Para amplitudes bajas ( $|u|_{max} < 0.045$  m), el error es mínimo. El sistema se comporta casi linealmente dentro del pozo.
- Para amplitudes altas ( $|u|_{max} > 0.048$  m), el error salta abruptamente a casi 200 %.

Esto demuestra que **EulerNet falla específicamente en la región de alta no-linealidad geométrica** (cerca del punto inestable  $u = 0$ ). El error de truncamiento del integrador ( $O(\Delta t)$ ) introduce una incertidumbre numérica que, en este sistema sensible, altera la topología de la solución, haciendo que el modelo numérico diverja cualitativamente de la realidad física.

**Conclusión del Caso A:** El sistema de Doble Pozo ha expuesto la primera gran limitación de la arquitectura base: la inestabilidad frente al caos transitorio. Si bien la red aprende la dinámica local, el integrador de primer orden carece de la precisión necesaria para resolver correctamente las bifurcaciones cerca de puntos de

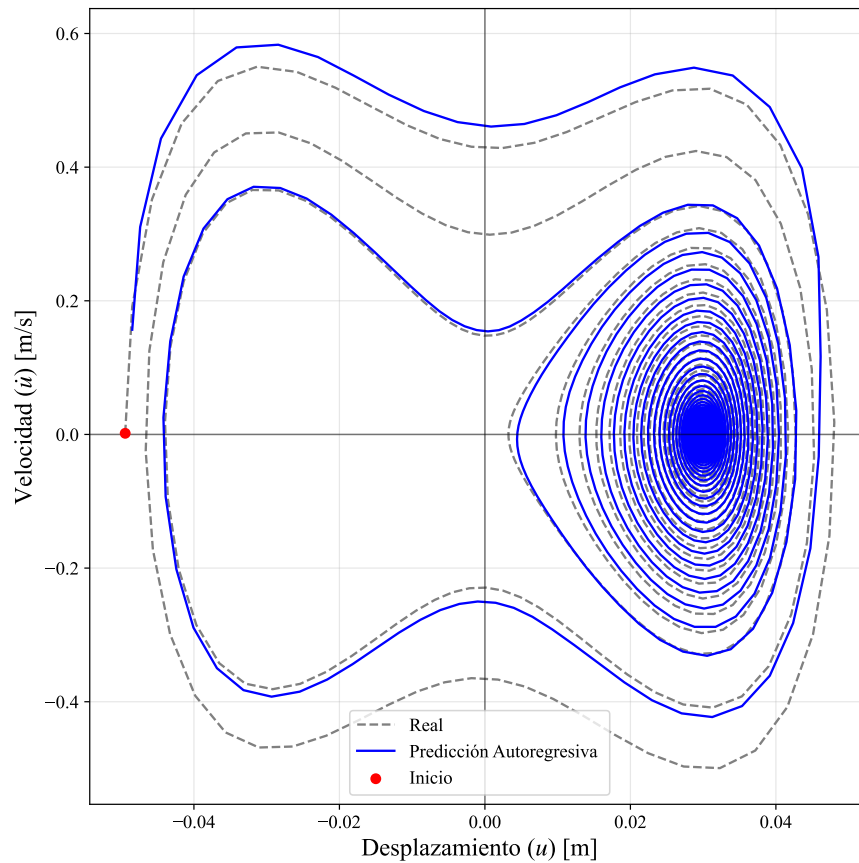


Figura 4.12: Retrato de Fase Duffing Doble Pozo. Aunque la red captura la topología del atractor derecho, se evidencia una dificultad para replicar la tasa exacta de convergencia inicial al equilibrio.

equilibrio inestable, resultando en predicciones que, aunque físicamente posibles, son incorrectas respecto a la condición inicial dada.

#### 4.4.3. Caso B: Sistema con Histéresis Severa (Bouc-Wen)

El último caso de estudio de esta fase busca evaluar la capacidad del integrador neuronal frente a no-linealidades constitutivas severas. A diferencia de la no-linealidad geométrica suave de Duffing, este sistema modela el comportamiento inelástico con degradación de rigidez y disipación de energía histerética, utilizando el modelo fenomenológico de Bouc-Wen.

Para dotar de realismo ingenieril al experimento, se ha diseñado el sistema representando un edificio con **Aislamiento Sísmico de Base**. Estos sistemas se caracterizan por una transición elasto-plástica abrupta y una baja rigidez post-fluencia, lo que genera ecuaciones diferenciales rígidas (*stiff equations*) que desafían la estabilidad de los integradores explícitos clásicos.

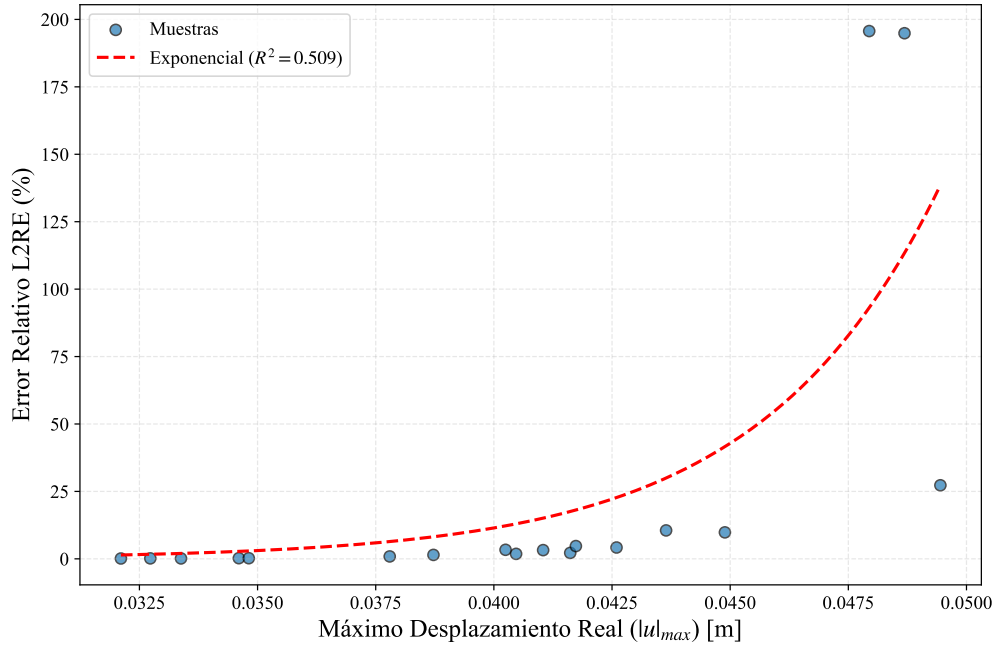


Figura 4.13: Diagnóstico de Error vs. Amplitud. Se observa una bifurcación clara: las trayectorias con suficiente energía para acercarse al punto inestable (amplitudes altas) sufren errores catastróficos, confirmando la limitación del integrador ante dinámicas complejas.

### Formulación Matemática y Espacio de Estados Aumentado

La dinámica del sistema está gobernada por la ecuación de movimiento para un grado de libertad, donde la fuerza restauradora se descompone en una componente elástica lineal y una componente histerética auxiliar. Basado en la implementación numérica utilizada, la ecuación de movimiento es:

$$m\ddot{u}(t) + c\dot{u}(t) + \underbrace{\alpha ku(t) + (1 - \alpha)ku_y z(t)}_{f_s(u,z)} = 0 \quad (4.31)$$

Donde  $\alpha$  es la relación de rigidez post-fluencia y  $u_y$  es el desplazamiento de fluencia. La variable interna adimensional  $z(t)$  captura la memoria del material y evoluciona según la ecuación diferencial constitutiva de Bouc-Wen:

$$\dot{z}(t) = \frac{1}{u_y} \left( \dot{u} - \beta|\dot{u}||z|^{n-1}z - \gamma\dot{u}|z|^n \right) \quad (4.32)$$

Para que la red neuronal pueda operar como un integrador autónomo, debe ser capaz de predecir no solo la cinemática ( $u, \dot{u}$ ), sino también la evolución de la variable oculta  $z$ . Esto implica aumentar la dimensión del vector de estado a  $\mathbf{x} = [u, v, z]^T \in \mathbb{R}^3$ .

El problema de aprendizaje supervisado se define como la aproximación del campo vectorial  $\mathcal{N}_\theta(\mathbf{x}_t) \approx \dot{\mathbf{x}}_t$ :

$$\mathcal{N}_\theta \left( \begin{bmatrix} u \\ v \\ z \end{bmatrix} \right) \longrightarrow \begin{bmatrix} v \\ -\frac{1}{m}(cv + \alpha ku + (1 - \alpha)ku_y z) \\ \frac{1}{u_y}(v - \beta|v||z|^{n-1}z - \gamma v|z|^n) \end{bmatrix} \quad (4.33)$$

### Diseño y Configuración del Sistema Estructural

Los parámetros físicos del sistema no fueron elegidos arbitrariamente, sino calculados para representar un edificio aislado típico. El diseño se basó en un Periodo Efectivo objetivo ( $T_{eff}$ ) y una demanda de desplazamiento máxima esperada, derivando las propiedades de rigidez y amortiguamiento resultantes.

La Tabla 4.7 resume los parámetros estructurales derivados del siguiente proceso de diseño:

- **Masa ( $M$ ):** Se consideró una superestructura de 15,000 kN y una nivel base de aislación de 4,500 kN, totalizando un peso sísmico de 19,500 kN.
- **Rigidez ( $k$ ):** Se definió un periodo objetivo aislado de  $T_{eff} = 2.5$  s, asumiendo una ductilidad de diseño  $\mu = 6$ .
- **Amortiguamiento ( $c$ ):** Se calculó el amortiguamiento viscoso equivalente ( $c_{eff}$ ) basado en la energía disipada por ciclo histérico bilineal perfecto, resultando en un alto nivel de amortiguamiento efectivo ( $\xi_{eff} \approx 50\%$ ).

Tabla 4.7: Parámetros Físicos y Constitutivos del Sistema de Aislamiento (Bouc-Wen).

Parámetro	Símbolo	Valor
<i>Propiedades Físicas Derivadas</i>		
Masa Total	$m$	1,987.77 ton
Rigidez Elástica (Inicial)	$k$	71,750 kN/m
Rigidez Post-Fluencia	$k_{pos}$	717.5 kN/m
Desplazamiento de Fluencia	$u_y$	0.06 m (6 cm)
Amortiguamiento Viscoso	$c$	4,997.9 kNs/m
Periodo Efectivo	$T_{eff}$	2.50 s
<i>Parámetros Constitutivos (Bouc-Wen)</i>		
Ratio de Rigidez	$\alpha$	0.01
Parámetros de Forma	$\beta, \gamma$	0.9, 0.1
Exponente de Transición	$n$	2.0

**Generación de Datos: Simulación de Sismo Impulsivo** A diferencia de los experimentos previos donde se varió el desplazamiento inicial, en este caso se busca simular la respuesta del aislador ante una carga extrema tipo **sismo impulsivo** (Near-Fault Pulse) o un impacto.

Por esta razón, la generación de las  $N = 100$  trayectorias de entrenamiento se realizó imponiendo un desplazamiento inicial nulo y variando exclusivamente la velocidad inicial, simulando una transferencia súbita de energía cinética a la base.

$$u_0 = 0, \quad v_0 \sim \mathcal{U}(-1.4, 1.4) \text{ m/s} \quad (4.34)$$

Una velocidad inicial de 1.4 m/s induce desplazamientos que superan ampliamente el punto de fluencia ( $u_y = 6$  cm), forzando al sistema a incursionar profundamente en el rango inelástico donde la rigidez cae al 1 % de su valor original ( $\alpha = 0.01$ ). Esta condición de carga es ideal para evaluar si la red neuronal es capaz de aprender y reproducir las transiciones abruptas de rigidez sin acumular errores de deriva (*drift*).

### Continuidad Metodológica

Se mantiene la arquitectura de Perceptrón Multicapa (MLP) con funciones de activación tanh y la estrategia de normalización Z-score descrita en la Fase I. La única modificación estructural es la dimensión de la capa de entrada y salida, que ahora procesa vectores de  $\mathbb{R}^3$  para acomodar la variable histerética  $z$ . Los datos de referencia (Ground Truth) se obtuvieron mediante integración numérica de alta fidelidad utilizando el solver ‘LSODA’ (implementado en `scipy.integrate.odeint`), capaz de manejar la rigidez numérica de las ecuaciones diferenciales constitutivas.

### Resultados y Discusión del Caso B (Sistema de Aislamiento Boucwen)

El modelo fue evaluado en el conjunto de prueba de 18 trayectorias bajo excitación de velocidad inicial (sismo impulsivo). Los resultados revelan una dicotomía crítica entre la capacidad de aprendizaje de la física local y la estabilidad global de la integración.

**Desempeño Cuantitativo** La Tabla 4.8 resume las métricas de error. Se observa un fenómeno revelador:

- **Excelente Aprendizaje de la Ley Constitutiva:** Las variables que dependen directamente del gradiente aprendido ( $\dot{v} \propto$  Aceleración y  $\dot{z} \propto$  Histéresis) presentan errores extremadamente bajos, del orden del 1 % al 3 %. Esto confirma que la red neuronal  $\mathcal{N}_\theta$  ha aprendido exitosamente la compleja ecuación diferencial de Bouc-Wen.
- **Falla en el Desplazamiento:** A pesar de la precisión en las derivadas, la variable integral (desplazamiento  $u$ ) exhibe un error masivo del **27.95 %**, con casos extremos superando el 250 %.

Tabla 4.8: Resumen de Métricas de Desempeño (Promedio en Test - Bouc-Wen)

Métrica	RMSE (Abs)	L2RE (Rel)	Max L2RE
Desplazamiento ( $u$ )	$2.57 \times 10^{-3}$ m	<b>27.95 %</b>	255.97 %
Velocidad ( $v$ )	$7.05 \times 10^{-4}$ m/s	1.37 %	5.99 %
Aceleración ( $a$ )	$1.01 \times 10^{-2}$ m/s <sup>2</sup>	2.90 %	4.75 %
Variable Hist. ( $z$ )	$1.99 \times 10^{-3}$	1.60 %	6.36 %

**Análisis del Drift Numérico** La Figura 4.14 ilustra la causa física de esta discrepancia. Mientras que la velocidad y aceleración (paneles inferiores) se superponen perfectamente a la referencia, el desplazamiento (panel superior) muestra una separación progresiva de la línea base.

Este fenómeno, conocido como **Drift Numérico**, ocurre porque el integrador de Euler ( $u_{t+1} = u_t + v_t \Delta t$ ) carece de corrección de errores. En un sistema rígido como este, pequeños errores de overshoot en la velocidad al cruzar el punto de fluencia generan un sesgo constante (bias). Al integrar una velocidad con un pequeño sesgo constante, se genera una rampa lineal de error en el desplazamiento, alejando la estructura de su posición de equilibrio real.

**Validación de la Respuesta Fuerza-Desplazamiento** La Figura 4.15 muestra la reconstrucción de la fuerza restauradora  $f_s$  utilizando los estados predichos por la red neuronal. Dado que esta trayectoria específica corresponde a una excitación de amplitud moderada, el sistema opera predominantemente en su régimen lineal-elástico, comportándose casi como un oscilador simple ( $f_s \approx ku$ ).

Sin embargo, se observa una discrepancia significativa en la trayectoria de carga y descarga. Mientras que la referencia analítica (línea gris) traza una recta limpia que pasa por el origen, la predicción neuronal (línea roja discontinua) exhibe un comportamiento errático con bucles abiertos que se desplazan horizontalmente. Este fenómeno no implica que la red haya aprendido una ley constitutiva incorrecta, sino que es un artefacto directo del *drift* en el desplazamiento ( $u_{pred}$ ). Al evaluar la fuerza  $f_s(u_{pred}, z_{pred})$  con un desplazamiento que se aleja progresivamente del real, la curva de fuerza se *desliza* sobre el eje de abscisas, generando una falsa apariencia de endurecimiento o ablandamiento que no corresponde a la física del material, sino al error acumulado del integrador numérico.

**Robustez y Distribución de Errores** El diagrama de caja (Figura 4.16) refuerza el diagnóstico. La dispersión de los errores en velocidad, aceleración y parámetro de histéresis  $z$  es mínima y se mantiene consistentemente bajo el 5 %. Por el contrario, el desplazamiento presenta una varianza enorme con múltiples valores atípicos.

Esto demuestra que **la red neuronal es robusta** (predice consistentemente bien la dinámica instantánea), pero el **esquema numérico es frágil**. La calidad de la predicción de desplazamiento depende fuertemente de si la trayectoria específica acumula errores en una dirección preferente o si se cancelan mutuamente por azar.

**Diagnóstico de Sensibilidad** Finalmente, el análisis de error vs. amplitud (Figura 4.17) muestra una tendencia contraintuitiva: los mayores errores relativos se concentran en las amplitudes pequeñas. Esto se debe a que, en sismos impulsivos de baja intensidad, la estructura puede quedar oscilando con un pequeño desplazamiento residual permanente. Si el modelo no captura este residual exacto, el error relativo ( $|u_{pred} - u_{real}|/|u_{real}|$ ) tiende a infinito cuando  $u_{real} \rightarrow 0$ . Sin embargo, el problema estructural de fondo sigue siendo el drift acumulado en todas las amplitudes.

**Conclusión del Caso B:** El sistema de aislamiento sísmico ha proporcionado la prueba definitiva. La red neuronal demostró una capacidad excepcional para aprender ecuaciones constitutivas rígidas (Bouc-Wen), prediciendo fuerzas y aceleraciones con error  $< 3 \%$ . Sin embargo, la arquitectura EulerNet falló en la tarea

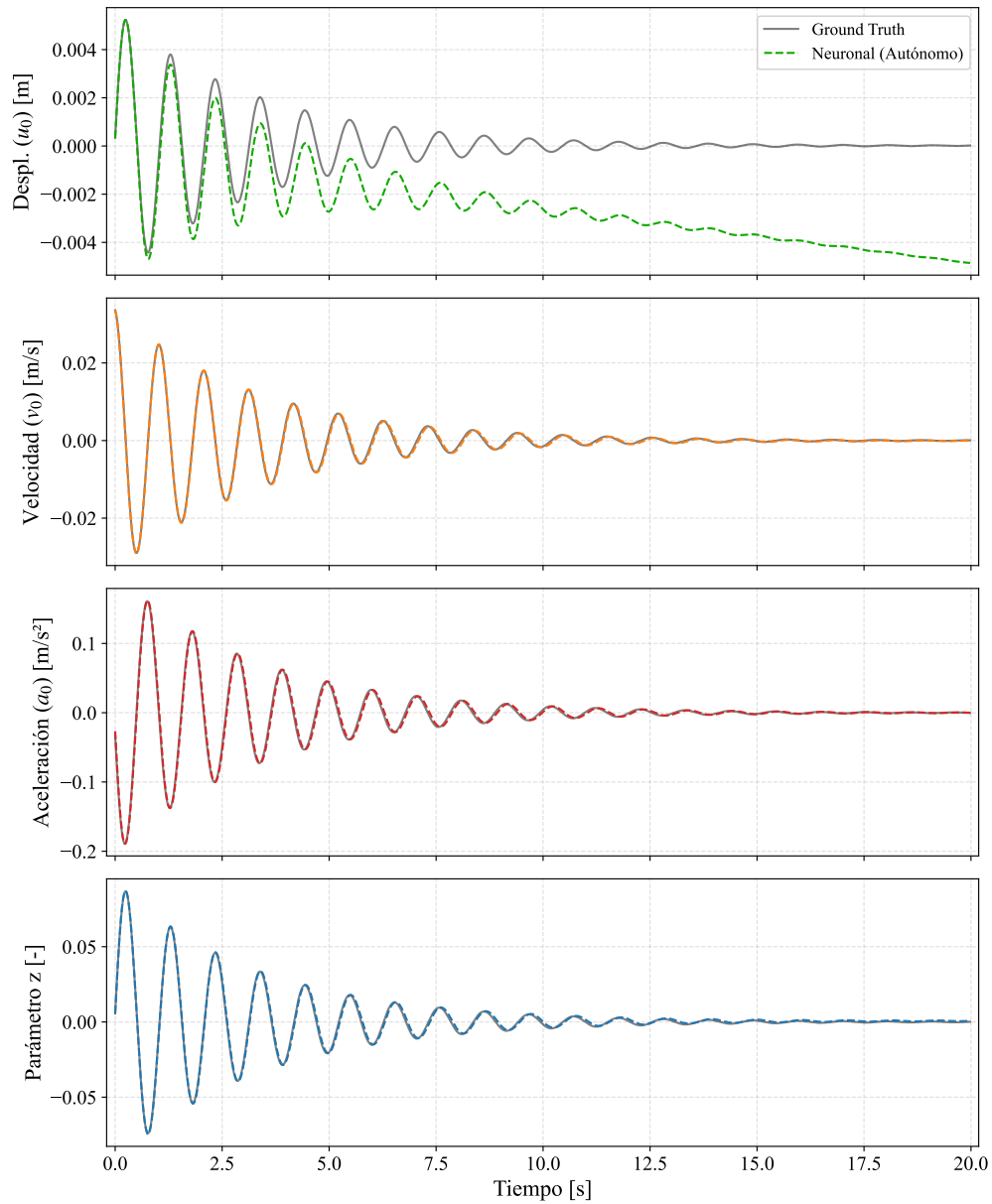


Figura 4.14: Historia de Tiempo del Sistema Aislado. Obsérvese cómo la velocidad y aceleración son predichas con alta precisión (paneles inferiores), mientras que el desplazamiento (panel superior) sufre una deriva o drift significativo hacia valores negativos, evidenciando la falla de la integración explícita.

de integración temporal, generando desplazamientos espurios (drift). Esto confirma que para aplicaciones de ingeniería real, la inteligencia artificial debe ser acoplada a solvers numéricos de mayor orden (como Runge-Kutta) que garanticen la estabilidad global que el método de Euler no puede proveer.

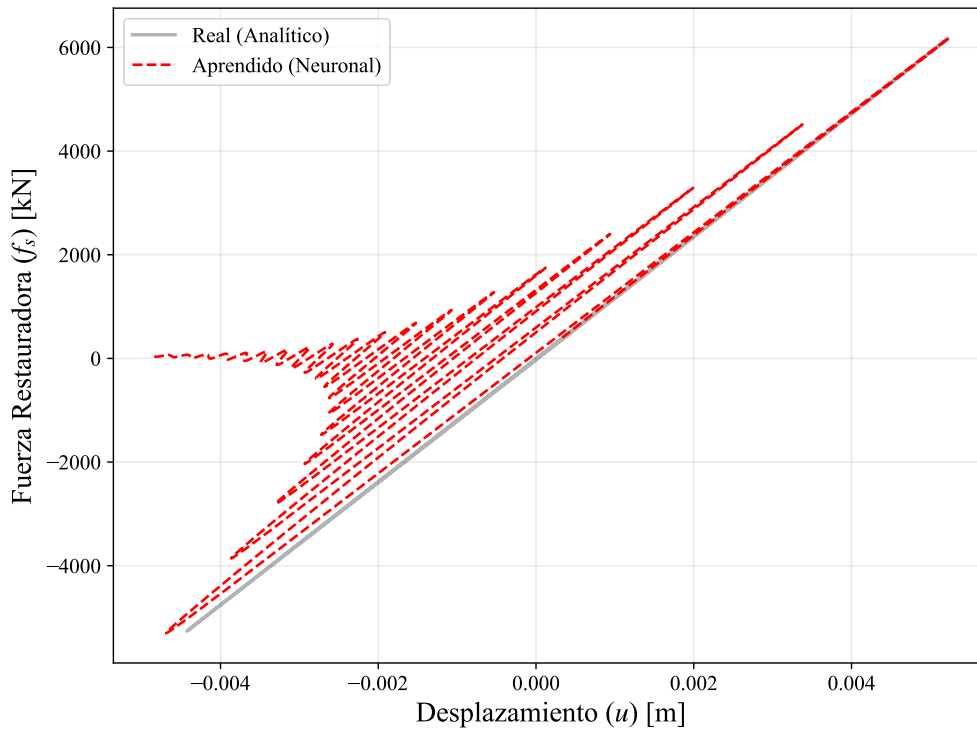


Figura 4.15: Ciclos de Histéresis. La red replica la forma constitutiva compleja de Bouc-Wen (fuerza vs. desplazamiento relativo), pero el error de integración induce un desplazamiento horizontal de los ciclos, impidiendo el cierre perfecto de los lazos.

#### 4.4.4. Metodología de Diagnóstico: Análisis de Convergencia del Esquema Numérico

Ante la degradación del desempeño observada en los sistemas complejos (Casos A y B), surge la necesidad de discriminar la causa raíz del error: ¿Es una insuficiencia en la capacidad de aproximación universal de la red neuronal ( $\mathcal{N}_\theta \neq f$ ), o es una limitación estructural impuesta por el esquema de integración de primer orden?

Para responder a esto, se diseñó un experimento de convergencia numérica mediante reentrenamiento. La hipótesis fundamental es que, dado que la arquitectura EulerNet integra la regla de Euler directamente en su función de pérdida, el límite inferior del error alcanzable por el modelo debe estar gobernado por el error de truncamiento global del método ( $O(\Delta t)$ ), independientemente de la calidad de los datos de entrenamiento.

Se generaron tres conjuntos de datos independientes (*Ground Truth*) variando la resolución temporal de la integración numérica de referencia:

1. **Dataset de Alta Resolución** ( $\Delta t = 0.001$  s): Paso de tiempo fino para minimizar el error de discretización.
2. **Dataset Nominal** ( $\Delta t = 0.01$  s): Paso de tiempo estándar utilizado en la Fase I.
3. **Dataset de Baja Resolución** ( $\Delta t = 0.05$  s): Paso de tiempo grueso para exacerbar los errores numéricos.

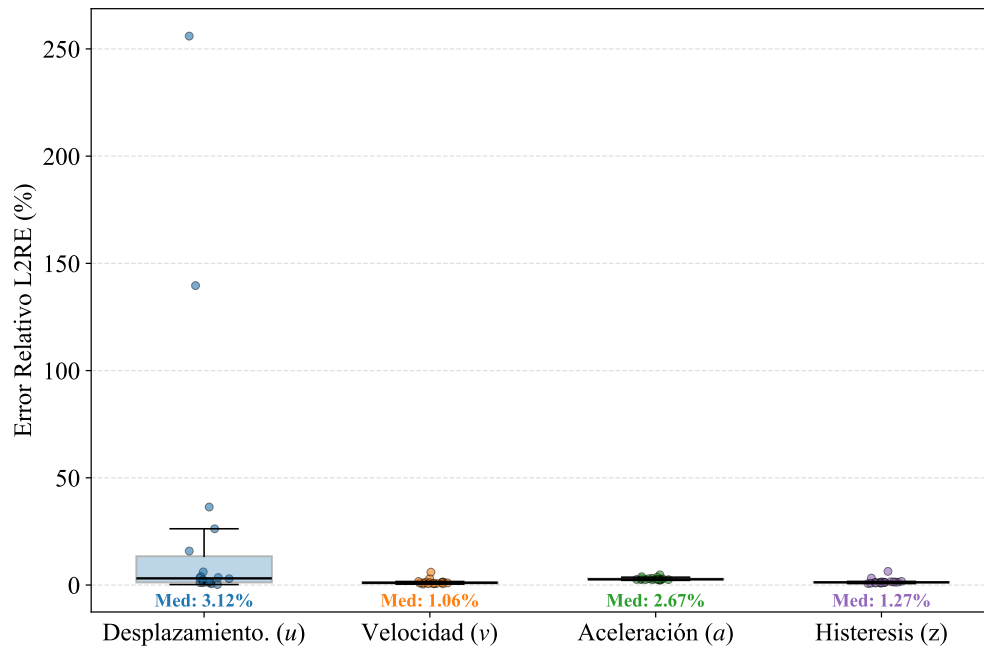


Figura 4.16: Distribución de Errores por Variable. Se evidencia el contraste dramático entre la alta precisión de las variables derivadas ( $v, a, z$ ) y la alta dispersión de la variable integral ( $u$ ).

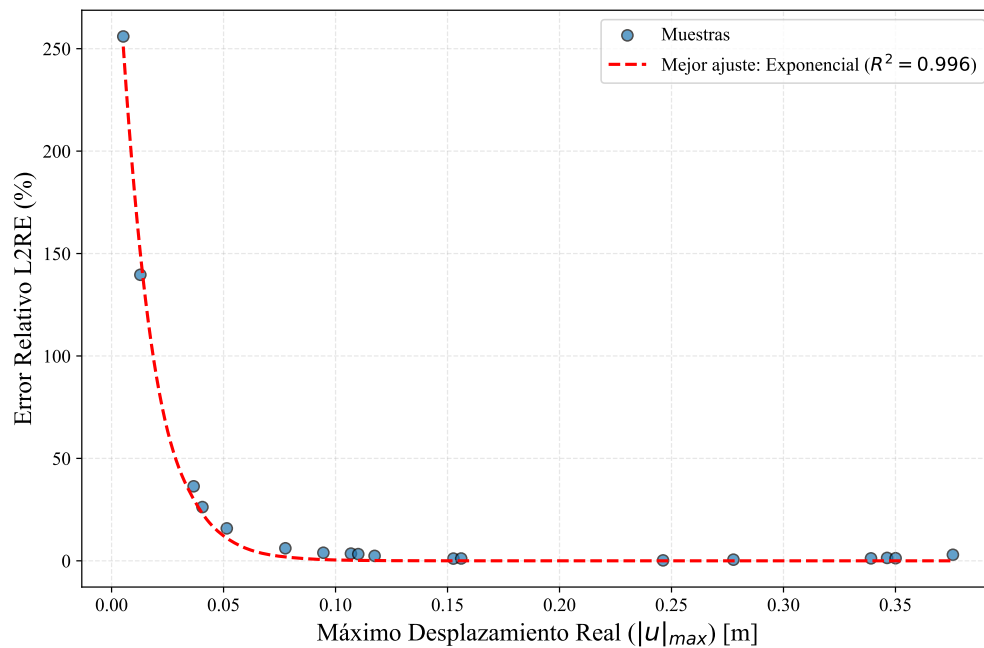


Figura 4.17: Sensibilidad del Error a la Amplitud. Los errores masivos a bajas amplitudes reflejan la dificultad de predecir desplazamientos residuales pequeños con un integrador que acumula deriva constante.

**Protocolo de Reentrenamiento:** A diferencia de un simple test de sensibilidad, aquí se entrenaron tres modelos independientes, cada uno entrenado específicamente para su resolución temporal correspondiente.

Esto asegura que la red neuronal tenga la oportunidad óptima de aprender la dinámica en cada escala, aislando así el desempeño del algoritmo de integración de cualquier problema de generalización de datos.

### Resultados del Análisis de Convergencia

A continuación, se presenta el análisis comparativo de los modelos reentrenados.

**Caso A: Oscilador Duffing (Sensibilidad al Caos)** La Tabla 4.9 resume el desempeño del modelo en función de la resolución temporal.

Tabla 4.9: Análisis de Convergencia - Duffing Doble Pozo (Error Relativo L2RE en %)

Variable	$\Delta t$ Fino (0.001s)	$\Delta t$ Nominal (0.01s)	$\Delta t$ Grueso (0.05s)
Desplazamiento ( $u$ )	<b>12.40 %</b>	25.61 %	30.63 %
Velocidad ( $v$ )	16.74 %	40.68 %	56.31 %
Aceleración ( $a$ )	19.41 %	49.48 %	76.81 %

**Análisis Profundo:** Los resultados muestran una mejora significativa al refinar el paso de tiempo, reduciendo el error de desplazamiento a la mitad (del 25.6 % al 12.4 %). Sin embargo, el hecho de que el error persista por encima del 10 % incluso con un  $\Delta t = 0.001s$  revela la naturaleza del sistema caótico. En la vecindad del punto de silla inestable ( $u = 0$ ), la dinámica del sistema es exponencialmente divergente. El método de Euler, al aproximar la derivada como constante en el intervalo, introduce un pequeño error de fase en cada paso. En un sistema caótico, estos errores microscópicos se amplifican exponencialmente en el tiempo (efecto mariposa). Aunque la red neuronal aprende correctamente el campo vectorial, la naturaleza iterativa de la EulerNet presenta dificultades para conservar la energía total del sistema con precisión absoluta en los umbrales críticos de salto. Esta pequeña desviación energética impide que el modelo determine correctamente si la trayectoria posee el impulso suficiente para cruzar de un pozo a otro, provocando los errores observados en la predicción de los saltos de fase.

**Caso B: Sistema Bouc-Wen (Validación de la Hipótesis)** Los resultados para el sistema de aislamiento sísmico (Tabla 4.10) constituyen la evidencia más contundente de esta investigación.

Tabla 4.10: Análisis de Convergencia - Bouc-Wen (Error Relativo L2RE en %)

Variable	$\Delta t$ Fino (0.001s)	$\Delta t$ Nominal (0.01s)	$\Delta t$ Grueso (0.05s)
Desplazamiento ( $u$ )	<b>3.82 %</b>	27.95 %	35.98 %
Velocidad ( $v$ )	0.34 %	1.37 %	1.22 %
Variable Hist. ( $z$ )	0.38 %	1.60 %	1.47 %

**Discusión y Relación con el Método de Euler:** Existe un colapso dramático del error de desplazamiento, que desciende del **27.95 %** al **3.82 %** al refinar el paso de tiempo. Este hallazgo permite establecer tres conclusiones mecánicas y numéricas fundamentales:

1. **La Red Neuronal es Competente:** El hecho de que el modelo con  $\Delta t = 0.001$  logre un error menor al 4 % demuestra que la arquitectura MLP es perfectamente capaz de aprender la topología de la histéresis de Bouc-Wen. Esto descarta la hipótesis de que la red no tuviera suficiente capacidad de representación para modelar el sistema rígido.
2. **El Integrador es el Cuello de Botella:** La variación del error sigue la tendencia teórica del método de Euler. Sabemos que el error local de truncamiento de Euler es proporcional a  $\frac{1}{2}\ddot{u}(\xi)\Delta t^2$ . En el sistema Bouc-Wen, durante las transiciones elasto-plásticas, el *jerk* (la tasa de cambio de la aceleración respecto al tiempo) alcanza valores extremadamente altos. Dado que el error de truncamiento del método de Euler depende de las derivadas de orden superior, estos picos de variabilidad en la aceleración exigen pasos de tiempo significativamente menores para mantener la estabilidad de la integración.
  - Con  $\Delta t$  grandes (0.05s y 0.01s), este término de error es dominante, generando el *drift* observado.
  - Con  $\Delta t$  pequeño (0.001s), el término  $\Delta t^2$  se vuelve despreciable, permitiendo que la integración siga la trayectoria real.
3. **Validación de la Causa del Drift:** Dado que los errores en las variables derivadas ( $v, z$ ) se mantienen bajos en todos los casos, se confirma que el error masivo en desplazamiento ( $u$ ) es puramente acumulativo. La red predice correctamente la *flecha* (velocidad), pero al usar Euler, estamos aproximando la curva integral mediante segmentos rectos. Al reducir el tamaño de estos segmentos, la aproximación poligonal converge a la curva real, eliminando la deriva artificial.

#### 4.4.5. Conclusión Final de la Fase II

El estudio de sensibilidad mediante la variación del paso de tiempo ( $\Delta t$ ) ha permitido aislar con éxito la fuente principal del error acumulado. Al evaluar modelos independientes entrenados con distintos intervalos temporales, se demuestra que **EulerNet captura correctamente la física subyacente del sistema**; sin embargo, su precisión global se encuentra matemáticamente acotada por el error de truncamiento inherente al esquema de integración de Euler.

Este análisis confirma que el cuello de botella no reside en la capacidad representacional de la red neuronal para aprender el campo vectorial, sino en la discretización temporal del integrador. Como se observó en los experimentos, una reducción en el  $\Delta t$  disminuye proporcionalmente el error local, permitiendo que la solución numérica converja hacia la trayectoria real, especialmente en regímenes de alto *jerk* donde las variaciones de aceleración son abruptas.

Para lograr simulaciones precisas (error < 5 %) en sistemas rígidos utilizando esta arquitectura, sería necesario entrenar y operar con pasos de tiempo del orden de  $10^{-3}$  s o inferiores. Sin embargo, esto aumenta el costo computacional de la inferencia en un factor de 10x, anulando la ventaja de velocidad que se busca con los modelos surrogados.

Por consiguiente, la solución óptima no es refinar infinitamente el  $\Delta t$  (fuerza bruta), sino mejorar la *inteligencia* del esquema numérico. Esto justifica de manera rigurosa la necesidad de implementar la arquitectura **RK4Net** (Runge-Kutta de 4to Orden) en la Fase III, la cual promete alcanzar la misma precisión del modelo

fino, pero manteniendo el paso de tiempo nominal ( $\Delta t = 0.01$  s), desacoplando así la precisión de la carga computacional.

## 4.5. Fase III: Arquitectura Avanzada RK4Net y Solución al Drift

Habiendo diagnosticado que el error en la Fase II no proviene de la falta de capacidad de aprendizaje de la red, sino de la simplicidad del integrador de Euler, esta sección presenta una posible solución: la arquitectura **RK4Net**.

La hipótesis central de esta fase sostiene que, al incorporar un esquema de Runge-Kutta de cuarto orden (RK4) directamente en el grafo computacional de la red neuronal, es posible elevar la precisión del modelo sin recurrir a un refinamiento extremo del paso de tiempo. Matemáticamente, esto implica transitar de un esquema de Euler, cuyo error de truncamiento local es  $O(\Delta t^2)$ , hacia un esquema RK4 con un error local de  $O(\Delta t^5)$ .

Esta mejora en la resolución por paso permite reducir el error acumulado a lo largo de toda la trayectoria (error global) de  $O(\Delta t)$  a  $O(\Delta t^4)$ . Al mitigar significativamente el *drift* energético mediante una estructura interna más robusta, se espera capturar dinámicas complejas, manteniendo un compromiso óptimo entre el costo computacional de entrenamiento y la fidelidad de la predicción autorregresiva de largo plazo.

### 4.5.1. Formulación Matemática del Modelo RK4Net

La arquitectura propuesta, `RK4StateIntegrator`, mantiene el mismo núcleo de aproximación universal que EulerNet: una red Perceptrón Multicapa (MLP)  $\mathcal{N}_\theta$  que estima el campo vectorial  $f(\mathbf{x}, t)$ . La innovación radica en cómo se utiliza esta red para propagar el estado en el tiempo.

En lugar de una única evaluación por paso, el modelo realiza cuatro consultas estratégicas al campo vectorial para estimar una pendiente promedio ponderada. Matemáticamente, el paso *forward* de la red se define por el siguiente algoritmo secuencial:

$$\mathbf{k}_1 = \mathcal{N}_\theta(\mathbf{x}_t) \quad (4.35)$$

$$\mathbf{k}_2 = \mathcal{N}_\theta(\mathbf{x}_t + 0.5\Delta t \cdot \mathbf{k}_1) \quad (4.36)$$

$$\mathbf{k}_3 = \mathcal{N}_\theta(\mathbf{x}_t + 0.5\Delta t \cdot \mathbf{k}_2) \quad (4.37)$$

$$\mathbf{k}_4 = \mathcal{N}_\theta(\mathbf{x}_t + \Delta t \cdot \mathbf{k}_3) \quad (4.38)$$

Donde cada  $\mathbf{k}_i$  representa una estimación de la derivada en puntos intermedios del intervalo. La actualización final del estado se calcula como:

$$\hat{\mathbf{x}}_{t+\Delta t} = \mathbf{x}_t + \frac{\Delta t}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4) \quad (4.39)$$

### Interpretación Física de la Arquitectura

Esta formulación dota a la red neuronal de una capacidad de *previsión* geométrica superior:

- **$k_1$  (Euler):** Predicción basada en la pendiente actual.
- **$k_2, k_3$  (Corrección de Curvatura):** La red *tantea* el punto medio del intervalo. Si la trayectoria es curva (alta no-linealidad), estas pendientes corregirán la proyección lineal inicial.
- **$k_4$  (Cierre):** Evalúa la derivada al final del intervalo predicho para asegurar un cierre suave de la trayectoria.

Al promediar estas cuatro evaluaciones, el término de error de truncamiento local escala con  $\Delta t^5$ , lo que en la práctica elimina el sesgo sistemático (drift) observado en el sistema Bouc-Wen.

### 4.5.2. Entrenamiento Directo (Cold Start) y la Paradoja de Optimización

Para validar la superioridad teórica del esquema RK4, se procedió a entrenar la arquitectura RK4StateIntegrator desde cero (*Cold Start*), inicializando los pesos de la red  $\mathcal{N}_\theta$  con la distribución estándar de Xavier/Glorot. Se utilizó el mismo conjunto de datos del sistema Bouc-Wen y los mismos hiperparámetros de optimización (tasa de aprendizaje, épocas) que en la Fase II.

#### Resultados Preliminares: Falla de Convergencia

Contrariamente a la expectativa teórica inicial, el modelo RK4Net entrenado desde cero no logró superar consistentemente el desempeño del modelo EulerNet simple. La Tabla 4.11 muestra los resultados de error en el conjunto de prueba para el sistema Bouc-Wen.

Tabla 4.11: Desempeño Comparativo: Euler vs. RK4 (Cold Start) en Bouc-Wen

Métrica (Desplazamiento $u$ )	EulerNet (Fase II)	RK4Net (Cold Start)
L2RE Promedio	27.95 %	<b>24.28 %</b>
Max L2RE (Peor caso)	255.97 %	219.84 %

Aunque existe una mejora marginal (del 27.9 % al 24.2 %), el error sigue siendo inaceptablemente alto. El *drift* en el desplazamiento persiste y la red no logra aprovechar la capacidad de corrección de cuarto orden para rectificar la trayectoria.

**Colapso en el Sistema Duffing** La situación fue aún más crítica para el oscilador de Duffing. La Tabla 4.12 revela una falla catastrófica en la estabilidad del entrenamiento directo.

Tabla 4.12: Resumen de Rendimiento - Duffing Doble Pozo (RK4 Cold Start)

Variable	RMSE (Abs)	L2RE (Rel)	Max L2RE
Desplazamiento ( $u$ )	$9.06 \times 10^{-3}$ m	30.91 %	192.75 %
Velocidad ( $v$ )	$1.08 \times 10^{-1}$ m/s	<b>107.23 %</b>	687.24 %
Aceleración ( $a$ )	1.58 m/s <sup>2</sup>	87.09 %	472.05 %

Se observa que el error relativo promedio en la velocidad excede el **100 %**, con picos cercanos al 700 %. Físicamente, esto indica que el modelo no solo es impreciso, sino numéricamente inestable: la red neuronal predice incrementos de energía espurios que se amplifican en cada sub-paso del integrador RK4, llevando a velocidades divergentes.

La Figura 4.18 confirma este diagnóstico visualmente. A diferencia del sistema Bouc-Wen donde el error se concentraba en el desplazamiento (drift), aquí la incertidumbre se propaga violentamente a las derivadas ( $v, a$ ), evidenciando que el optimizador no logró encontrar un mínimo local estable en la rugosa superficie de pérdida del integrador de cuarto orden.

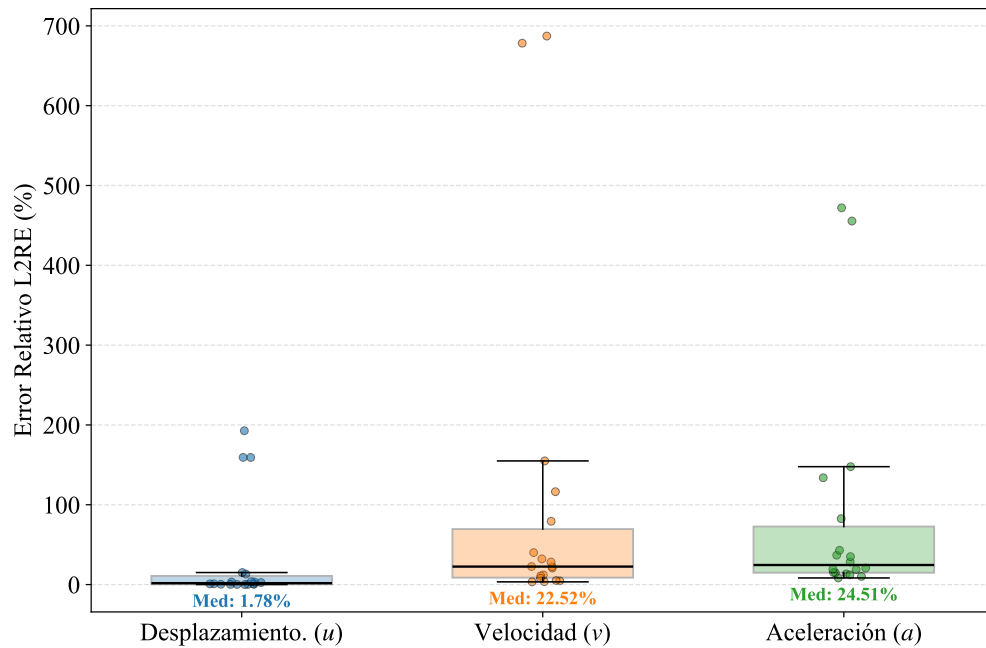


Figura 4.18: Distribución de errores para Duffing con RK4 (Cold Start). Obsérvese la escala masiva del error en velocidad (columna central), con una mediana superior al 20 % y valores atípicos extremos, síntoma de una optimización fallida.

### Diagnóstico: Superficies de Pérdida Rugosas

Se identifica un fenómeno contraintuitivo que denominaremos en este trabajo como la **Paradoja de la Complejidad**: la incorporación de un esquema numérico de mayor orden (RK4), que teóricamente reduce el error de truncamiento local, resulta en una inestabilidad severa durante el entrenamiento directo.

Este comportamiento es consistente con el *problema de degradación* descrito por He et al. (2016), y se alinea con las limitaciones de entrenabilidad observadas en Neural ODEs cuando aumenta el número de evaluaciones de la función (Chen et al., 2019). Computacionalmente, el esquema RK4 implica que la red neuronal  $\mathcal{N}_\theta$  se evalúa cuatro veces de manera recursiva en cada paso de tiempo ( $\mathbf{k}_1 \rightarrow \mathbf{k}_2 \rightarrow \mathbf{k}_3 \rightarrow \mathbf{k}_4$ ). Esto cuadruplica la profundidad efectiva del grafo computacional respecto al esquema de Euler, exacerbando el

problema de desvanecimiento del gradiente, ya que la señal de error debe propagarse a través de muchas más operaciones no-lineales, diluyéndose antes de llegar a los pesos.

Como consecuencia directa de este aumento en la profundidad, se genera una superficie de función de pérdida altamente no convexa o rugosa”. Al iniciar el entrenamiento con pesos aleatorios (*Cold Start*), el optimizador es incapaz de navegar esta topología compleja y tiende a quedar atrapado en mínimos locales subóptimos, donde la red aprende un promedio burdo de la dinámica en lugar de la función precisa.

En síntesis, aunque el esquema RK4 ofrece una precisión teórica superior, su implementación introduce una barrera de optimización significativa. La complejidad estructural del integrador transforma esta ventaja potencial en un obstáculo práctico para la convergencia, lo que sugiere que la optimización simultánea de la topología del campo vectorial y la integración de alto orden presenta una complejidad irreductible si no se cuenta con una inicialización adecuada.

### 4.5.3. Solución Propuesta: Estrategia de Transferencia (Warm Start)

Para superar la barrera de optimización, se propone una estrategia de entrenamiento curricular o de transferencia de aprendizaje, denominada **Warm Start** (Inicio en Caliente) (Bengio et al., 2009).

La hipótesis es la siguiente: La topología del campo vectorial  $f(\mathbf{x})$  es única e independiente del integrador. Por lo tanto, una red neuronal que ya ha aprendido una aproximación decente de  $f(\mathbf{x})$  usando un método simple (Euler), debería servir como un punto de partida ideal para un método complejo (RK4).

#### Protocolo Metodológico:

1. **Fase 1 (Pre-entrenamiento):** Entrenar la red bajo el esquema EulerNet durante un número reducido de épocas. Esto es computacionalmente barato y la superficie de error es más suave, permitiendo que la red encuentre rápidamente la *forma* general de la dinámica.
2. **Fase 2 (Transferencia):** Copiar los pesos  $\theta_{euler}$  resultantes a la arquitectura RK4Net.
3. **Fase 3 (Refinamiento):** Continuar el entrenamiento bajo el esquema RK4 con una tasa de aprendizaje reducida. En esta etapa, la red ya no necesita descubrir la física desde cero, sino que solo debe realizar ajustes finos (*fine-tuning*) para corregir los términos de orden superior que Euler ignoraba.

#### Resultados de la Estrategia Warm Start

Se implementó la estrategia de transferencia de aprendizaje en tres etapas. Primero, se pre-entrenó la red bajo el esquema Euler durante un número reducido de épocas para capturar la topología global  $f(\mathbf{x})$ . Posteriormente, se transfirieron los pesos a la arquitectura RK4Net y se procedió al refinamiento (*fine-tuning*) utilizando una tasa de aprendizaje reducida ( $\eta = 10^{-5}$ ), dos órdenes de magnitud menor que la utilizada en el entrenamiento inicial, para evitar la destrucción de las características aprendidas.

A continuación, se presentan los resultados comparativos que demuestran la eficacia de esta metodología.

**Sistema Bouc-Wen (Reducción del Drift)** La Tabla 4.13 resume la evolución del error a través de las tres configuraciones evaluadas.

Tabla 4.13: Comparación Final de Rendimiento - Sistema Bouc-Wen (L2RE Promedio)

Variable	EulerNet (Base)	RK4Net (Cold Start)	RK4Net (Warm Start)
Desplazamiento ( $u$ )	27.95 %	24.29 %	<b>6.76 %</b>
Velocidad ( $v$ )	1.37 %	1.73 %	<b>0.67 %</b>
Aceleración ( $a$ )	2.91 %	3.05 %	<b>2.91 %</b>
Variable Hist. ( $z$ )	1.60 %	2.15 %	<b>0.80 %</b>

**Análisis de Resultados:** El impacto de la estrategia Warm Start es drástico. El error de desplazamiento se reduce del  $\approx 28\%$  al **6.76%**. Este resultado confirma la hipótesis de que el *conocimiento* físico (los gradientes de fuerza y aceleración) ya estaba presente en la red entrenada con Euler (cuyos errores en  $v$  y  $a$  eran bajos), pero faltaba la precisión de integración. Al activar el esquema RK4 sobre los pesos pre-entrenados, el modelo logra corregir la curvatura de la trayectoria, mitigando drásticamente el fenómeno de deriva (*drift*) numérica. Este avance es particularmente visible en la variable interna  $z$  (error de 0.80%), cuya correcta evolución es la que gobierna la fuerza restauradora inelástica y, por ende, la estabilidad de la posición final.

La Figura 4.19 muestra la historia de tiempo para una de las trayectorias con mayor error acumulado del set de prueba. Resulta instructivo observar que, a pesar de ser uno de los casos que incrementa el promedio global de error, la predicción del desplazamiento se mantiene en fase con la referencia durante todo el registro. Esto demuestra que incluso en los escenarios donde el modelo presenta un ligero desfase residual, la estabilidad estructural aportada por el esquema RK4 impide la divergencia catastrófica que caracterizaba a la EulerNet base.

La Figura 4.20 ilustra la robustez. La mediana del error de desplazamiento se sitúa en un excelente **1.29%**, con una dispersión mucho menor que en los experimentos previos. Los valores atípicos se han reducido significativamente en magnitud.

**Oscilador Duffing (Estabilización del Caos)** Para el sistema de doble pozo, la estrategia *Warm Start* permitió estabilizar un entrenamiento que resultaba divergente bajo la modalidad *Cold Start*, logrando capturar la dinámica cerca de los puntos de equilibrio inestables.

Tabla 4.14: Comparación Final de Rendimiento - Duffing Doble Pozo (L2RE Promedio)

Variable	EulerNet (Base)	RK4Net (Cold Start)	RK4Net (Warm Start)
Desplazamiento ( $u$ )	25.61 %	30.91 %	<b>13.25 %</b>
Velocidad ( $v$ )	40.68 %	107.23 %	<b>23.25 %</b>
Aceleración ( $a$ )	49.49 %	87.09 %	<b>32.13 %</b>

**Interpretación:** El error promedio en el desplazamiento se redujo drásticamente respecto al modelo base (25.61%  $\rightarrow$  13.25%). Sin embargo, la Figura 4.21 revela que este valor promedio está influenciado por la naturaleza caótica del sistema: mientras que la **mediana del error de desplazamiento** es de apenas **0.83%**, indicando una precisión casi absoluta en la mayoría del conjunto de prueba, el promedio se eleva debido a

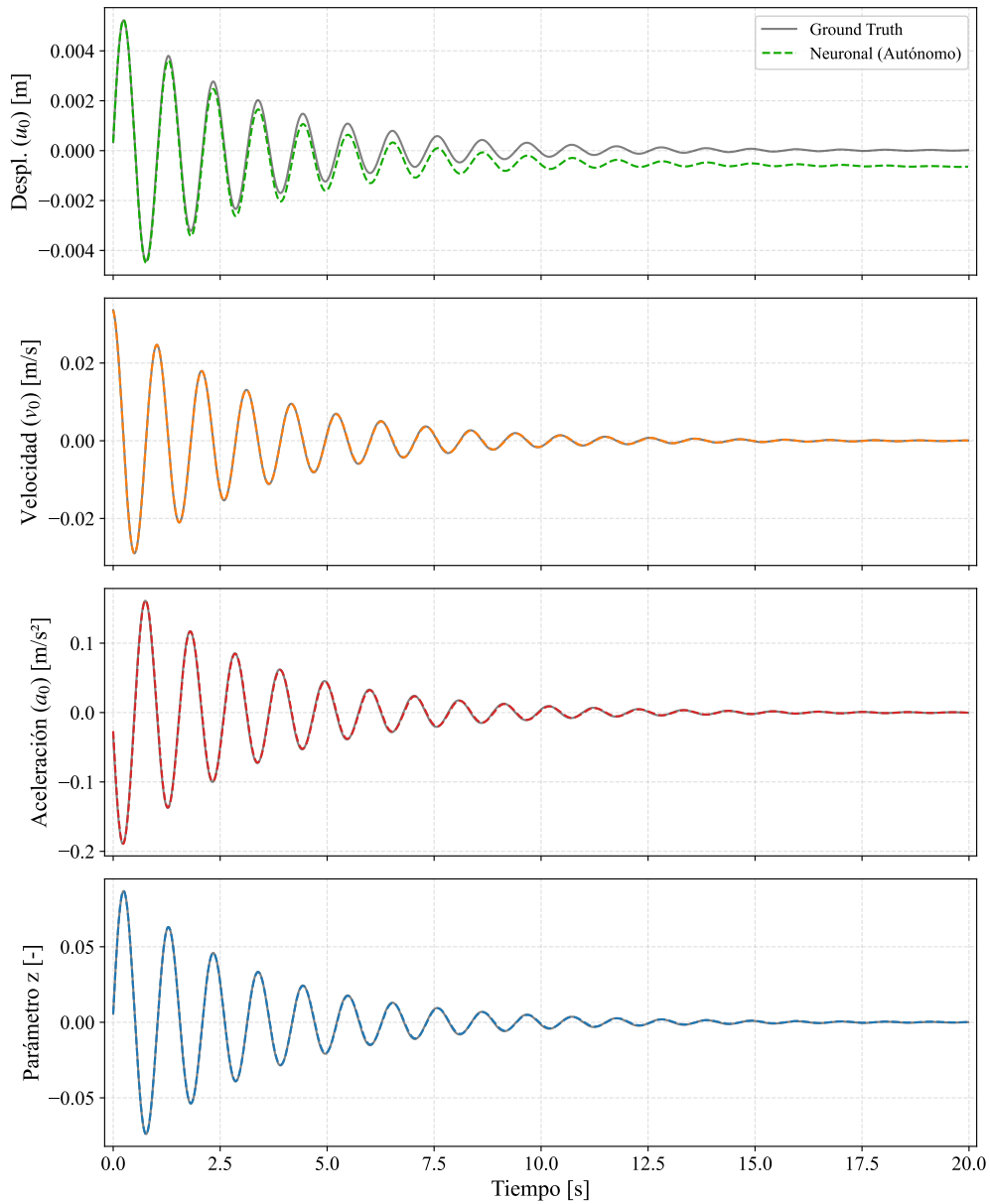


Figura 4.19: Respuesta Estructural con RK4Net (Warm Start). La deriva ha sido mitigada eficazmente. El modelo ahora es capaz de predecir desplazamientos permanentes con precisión ingenieril.

trayectorias críticas. En estas, mínimos desajustes numéricos en la vecindad del punto de equilibrio inestable se amplifican, provocando errores de fase significativos al decidir el salto de pozo.

Más notable aún es la corrección en la velocidad, donde el error descendió al **23.25 %** (mediana de **17.56 %**). Esto indica que el esquema RK4 es capaz de conservar mejor la energía del sistema, proporcionando una estimación más precisa de la velocidad cerca del equilibrio inestable, lo que minimiza las bifurcaciones erróneas (saltos de pozo incorrectos). Como se observa en el retrato de fase de la Figura 4.22, el modelo produce órbitas más estables y densas, respetando el **umbral de transición energética entre pozos** y minimizando la

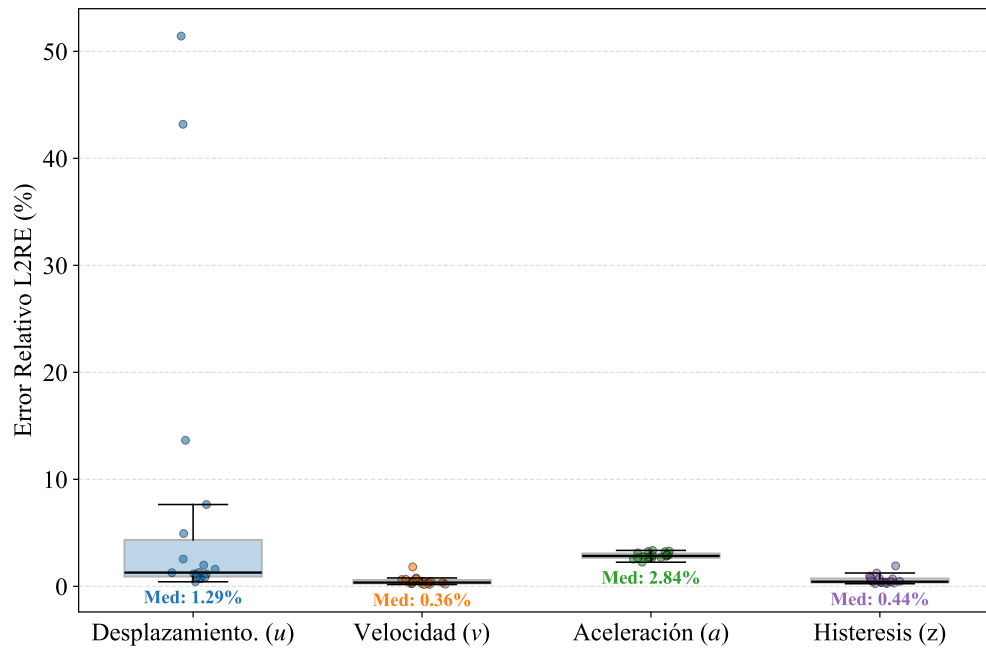


Figura 4.20: Distribución de Errores Finales (Bouc-Wen). La mediana del error en todas las variables se encuentra por debajo del 2.9 %, validando la confiabilidad del modelo.

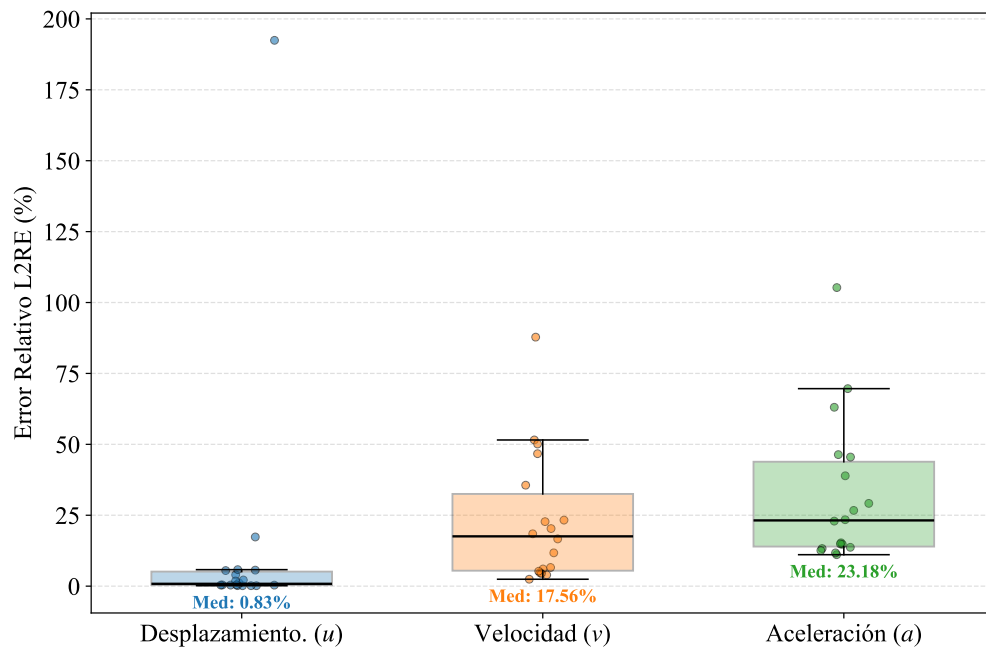


Figura 4.21: Distribución de Errores para el sistema de doble pozo (RK4Net). Se observa una precisión superior en el desplazamiento (mediana < 1 %), mientras que la velocidad y aceleración reflejan la dificultad de modelar las derivadas en los saltos de pozo.

disipación numérica espuria. Esta mejora en la preservación del espacio de fase permite que la red capture correctamente los ciclos de oscilación sin converger prematuramente hacia los atractores.

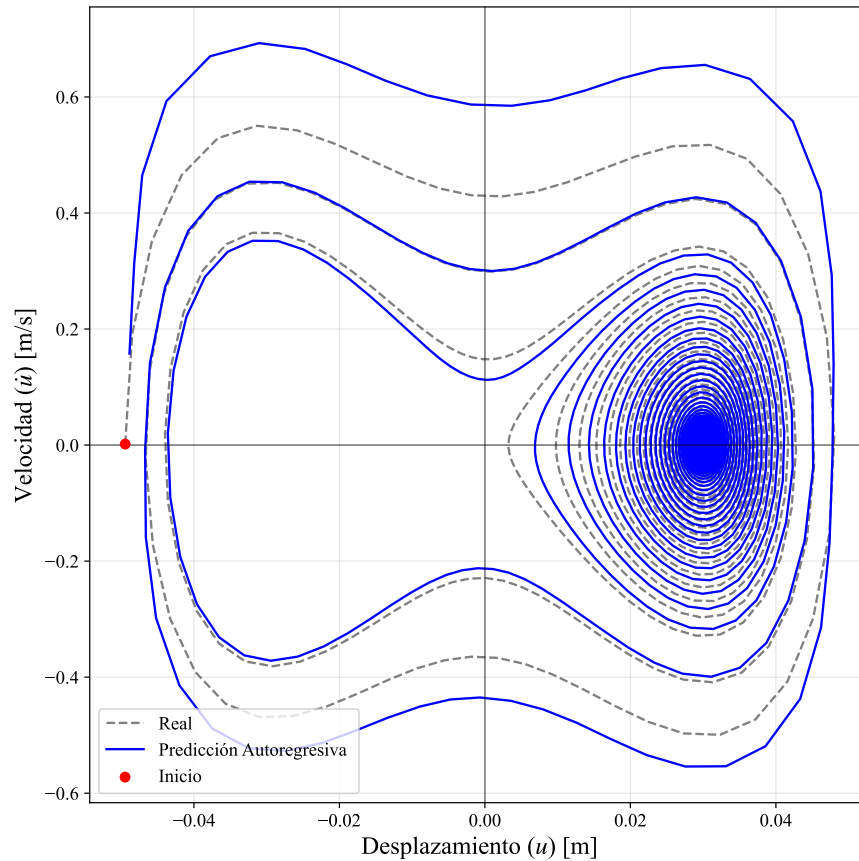


Figura 4.22: Retrato de Fase con RK4Net (Warm Start). La estabilidad orbital ha mejorado significativamente, capturando con mayor fidelidad la dinámica cerca de la separatriz.

#### 4.5.4. Conclusión General del Capítulo

Este capítulo ha demostrado que la integración de redes neuronales profundas con esquemas numéricos avanzados no es trivial, pero es altamente efectiva cuando se gestiona correctamente la optimización.

1. **La Arquitectura Importa:** Se demostró que para sistemas gobernados por ecuaciones diferenciales numéricamente rígidas (*stiff equations*), tal como ocurre en la dinámica no-lineal de aisladores sísmicos, los integradores neuronales de primer orden (EulerNet) son insuficientes debido al error de truncamiento acumulativo (*drift*), independientemente de la calidad del entrenamiento. La arquitectura RK4Net, con su error local  $O(\Delta t^5)$ , resolvió este problema estructural.
2. **La Optimización Importa:** Se identificó el fenómeno denominado *Paradoja de la Complejidad*, donde el entrenamiento directo de arquitecturas profundas (RK4) falla debido a superficies de pérdida mal condicionadas. La estrategia de **Warm Start** (transferencia Euler  $\rightarrow$  RK4) probó ser la clave para

desbloquear el potencial del integrador de alto orden, logrando reducir el error en un factor de 4x en sistemas histeréticos.

3. **Resultado Final:** Se ha obtenido un integrador neuronal autónomo capaz de predecir la respuesta sísmica no-lineal compleja con una **fidelidad típica superior al 98.7 %** (error mediano  $< 1.3 %$ ) en el desplazamiento global. Aunque en regímenes de alta demanda inelástica el error residual puede aumentar, la mitigación sistemática del *drift* confirma que el modelo ha superado la principal barrera técnica para el uso de redes neuronales en simulaciones dinámicas de larga duración.

## 4.6. Conclusiones y Trabajo Futuro

El desarrollo de este estudio validó el potencial de las redes neuronales profundas para actuar como integradores numéricos autónomos en dinámica estructural no-lineal. La implementación de la arquitectura propuesta **RK4Net**, potenciada por una estrategia de entrenamiento curricular (*Warm Start*), demostró ser una alternativa efectiva para superar las barreras de precisión y estabilidad que limitan a los enfoques tradicionales de primer orden.

Los resultados experimentales permiten establecer que, para sistemas gobernados por ecuaciones diferenciales numéricamente rígidas (*stiff equations*), los integradores neuronales de primer orden (EulerNet) resultan insuficientes debido al error de truncamiento acumulativo, independientemente de la calidad del entrenamiento. La arquitectura RK4Net resolvió este problema estructural al incorporar un esquema de cuarto orden, logrando reducir el error de desplazamiento en sistemas histeréticos complejos a una mediana cercana al 1.3 %, mitigando drásticamente el error de deriva (*drift*) incluso en trayectorias que presentan incursiones plásticas severas.

Asimismo, se identificó la ‘‘Paradoja de la Complejidad’’ en la optimización de integradores profundos, donde el entrenamiento directo (*Cold Start*) tiende a fallar debido a superficies de pérdida mal condicionadas. La estrategia de transferencia de aprendizaje desde un modelo simple (Euler) hacia uno complejo (RK4) probó ser la clave para desbloquear el potencial del integrador de alto orden, estabilizando la convergencia sin requerir pasos de tiempo excesivamente pequeños.

Como trabajo futuro, se proyecta extender esta metodología hacia sistemas no autónomos sometidos a excitación sísmica arbitraria, aprovechando la capacidad de la red para modelar fuerzas restauradoras internas. Adicionalmente, se sugiere explorar el uso de arquitecturas residuales (*ResNet*) para mitigar el desvanecimiento del gradiente en secuencias de muy larga duración, así como la incorporación de restricciones físicas estrictas en la función de pérdida (Physics-Informed) para garantizar la compatibilidad cinemática y el balance energético en regímenes de alta no-linealidad.

## Capítulo 5

# Conclusiones y Trabajo Futuro

### 5.1. Síntesis General

Esta investigación abordó el desafío de predecir la respuesta sísmica no-lineal y el daño estructural permanente (drift residual) mediante el uso de Redes Neuronales Informadas por la Física (PINNs). El estudio se articuló en dos fases complementarias: primero, una evaluación crítica de las arquitecturas *guided-informed* existentes (PhyCNN y PhyLSTM) bajo escenarios de alta demanda sísmica; y segundo, el desarrollo y validación de una nueva arquitectura *solver-informed* (RK4Net) diseñada para superar las limitaciones de estabilidad identificadas.

A partir de los resultados experimentales y el análisis diagnóstico realizado a lo largo de los capítulos precedentes, se derivan las siguientes conclusiones fundamentales y se proponen líneas de investigación futura.

### 5.2. Conclusiones Principales

#### 5.2.1. Limitaciones de los Enfoques Guided-Informed (PhyCNN/PhyLSTM)

El análisis comparativo realizado en el Capítulo 3 permitió establecer los límites de validez de las arquitecturas que incorporan la física únicamente como penalización en la función de pérdida. En escenarios con no-linealidades suaves (Duffing) y con histéresis moderada (Bouc–Wen excitado con BLWN), las arquitecturas PhyLSTM superan de forma consistente a PhyCNN en la predicción de desplazamientos y velocidades, mientras que las CNN muestran una estabilidad relativamente mayor cuando la fuerza interna se predice como salida directa (PhyCNN-MO). Sin embargo, en el sistema Bouc–Wen excitado con BLWN se observan algunos casos con cambios en el eje de oscilación, y ninguna de las arquitecturas evaluadas logra capturar estos efectos de forma robusta en todos los registros donde se presentan: PhyCNN-SO no es capaz de reproducir el cambio en la media, mientras que las demás variantes solo lo hacen de forma parcial y dependiendo del caso, lo que refleja una debilidad asociada a la naturaleza histerética de la base de datos.

En el escenario más exigente, Bouc–Wen representativo de aislación sísmica bajo sismos chilenos de larga duración, todas las arquitecturas se ven tensionadas: aumentan la media y la dispersión de los errores y aparecen derivas residuales espurias en algunos casos. Aun así, las PhyLSTM preservan mejor la geometría histerética en sus mejores registros, en tanto que las PhyCNN presentan sesgos sistemáticos. Las pérdidas físicas/kinemáticas ayudan a estabilizar trayectorias y a sostener la dinámica global; sin embargo, la fuerza

como salida explícita sin supervisión directa resulta frágil, pues el modelo puede satisfacer el residuo sin anclar correctamente la amplitud. En tales condiciones, reconstruir la fuerza desde la derivada de la velocidad entrega resultados más fiables.

La generalización hacia registros largos y con desplazamiento residual sigue siendo el principal desafío: se observan errores de fase acumulados y cambios de eje no deseados, especialmente cuando el entrenamiento es reducido y no cubre la diversidad espectro-temporal ni la duración propias de la sismicidad chilena. Cabe destacar que, aun en presencia de una restricción cinemática explícita entre desplazamiento y velocidad en la función de pérdida, los modelos son capaces de ajustar relativamente bien la velocidad pero siguen produciendo derivas de desplazamiento incompatibles con la integral de dicha señal, lo que evidencia que no logran internalizar de manera robusta la relación kinemática entre  $u(t)$  y  $v(t)$ . Las curvas de pérdida muestran que las PhyLSTM convergen a valores inferiores, aunque con oscilaciones a medida que aumentan las longitudes de secuencia, mientras que las PhyCNN tienden a estancarse tempranamente; el paso temporal, el operador de derivación y la ponderación relativa de las pérdidas condicionan de forma determinante el desempeño final.

### 5.2.2. Ventajas de la Arquitectura Solver-Informed (RK4Net)

La propuesta metodológica desarrollada en el Capítulo 4, basada en la integración explícita de un esquema numérico de alto orden dentro de la red neuronal, demostró ser una solución efectiva para los problemas de estabilidad. Si bien la Fase II demostró que es posible reducir el error de integración refinando el paso de tiempo ( $\Delta t \rightarrow 0$ ), esta estrategia conlleva un aumento prohibitivo en el costo computacional de entrenamiento, dado que las series temporales se vuelven excesivamente largas para el algoritmo de *Backpropagation*. La arquitectura **RK4Net** resuelve este compromiso eficiencia-precisión. Al aumentar la complejidad algorítmica del paso *forward* (4 evaluaciones por paso) en lugar de la resolución temporal, se logra una precisión de cuarto orden manteniendo la longitud original de las series. Esto permite entrenar modelos robustos con un costo computacional manejable, viabilizando su uso en problemas de gran escala, en línea con avances recientes que buscan integrar esquemas Runge-Kutta en arquitecturas recurrentes para predicción de respuesta (Zhai et al., 2024).

Se comprobó que el uso de un integrador Runge-Kutta de cuarto orden (RK4Net) reduce el error de truncamiento local de  $O(\Delta t^2)$  a  $O(\Delta t^5)$ . Esto permitió eliminar casi en su totalidad el drift numérico acumulativo que afectaba a los modelos basados en esquemas de Euler, logrando errores de desplazamiento inferiores al 7% en sistemas con histéresis severa. Asimismo, se demostró que el entrenamiento directo (*Cold Start*) de integradores profundos es inestable debido a la rugosidad de la superficie de pérdida. La estrategia de transferencia de aprendizaje propuesta (*Warm Start*), que inicia el entrenamiento con un modelo simple (Euler) y lo refina hacia uno complejo (RK4), probó ser esencial para alcanzar la convergencia y desbloquear el potencial de precisión del método de alto orden.

Una contribución distintiva de este trabajo respecto a la literatura existente sobre *Neural ODEs* radica en el carácter agnóstico del modelo. Mientras que otras implementaciones de integradores Runge-Kutta neuronales suelen incorporar conocimiento a priori de los parámetros físicos (masa, rigidez inicial) (Wang et al., 2023b) o se enfocan explícitamente en la estimación de parámetros constitutivos (Zhai et al., 2023), el modelo aquí

propuesto opera como una *Caja Gris* generalizable. La red aprende la topología del campo vectorial  $f(x)$  directamente de los datos cinemáticos, sin requerir la entrada explícita de coeficientes constitutivos. Esto sugiere que la arquitectura es aplicable a una amplia gama de sistemas no-lineales desconocidos, desde estructuras con daño oculto hasta sistemas de aislamiento con propiedades inciertas.

### 5.2.3. Generalización y Robustez

La comparación global de resultados valida la hipótesis central de la tesis: para sistemas estructurales gobernados por ecuaciones diferenciales numéricamente rígidas (*stiff equations*), la arquitectura de la red neuronal debe mimetizar la estructura del solver numérico. Los modelos puramente recurrentes o convolucionales, aunque flexibles, carecen del sesgo inductivo necesario para conservar la energía y la compatibilidad cinemática en regímenes altamente no-lineales. La arquitectura RK4Net, al operar como una Caja Gris que aprende la topología del campo vectorial, ofrece el equilibrio óptimo entre la eficiencia de los datos y el rigor físico.

Aunque este capítulo se restringió al régimen autónomo (vibración libre), los resultados permiten extrapolar con optimismo su aplicación a sistemas forzados (sismos). El argumento central es que la red neuronal ya ha resuelto la parte más compleja del problema: modelar la fuerza restauradora interna no-lineal  $f_s(u, v, z)$ . Matemáticamente, la inclusión de una excitación sísmica  $\ddot{u}_g(t)$  es una operación aditiva lineal en la ecuación de estado. Si el modelo es capaz de predecir correctamente la disipación de energía interna y la histéresis de forma autónoma, posee la capacidad intrínseca para reaccionar ante una entrada exógena, lo cual constituye el siguiente paso natural de esta línea de investigación.

## 5.3. Limitaciones y Recomendaciones Metodológicas

La metodología planteada por los autores de referencia en los modelos *guided-informed* no sigue completamente los protocolos estándar, lo que introduce incertidumbre en la comparabilidad. Falta una normalización sistemática y explícita de entradas y salidas con escalas físicas o estadísticas robustas. Las variables relevantes presentan órdenes de magnitud muy distintos; si se emplean pesos unitarios en la pérdida de datos, los términos asociados a aceleración tienden a dominar tanto el valor de  $\mathcal{L}_{\text{data}}$  como sus gradientes, favoreciendo la reducción del error de fuerza normalizada a costa de la precisión en desplazamiento, que es crítica para el *drift* residual. Esta heterogeneidad en escala implica gradientes de distinto orden de magnitud que pueden inducir oscilaciones en el entrenamiento, convergencia lenta o mínimos locales subóptimos.

El conjunto de validación fue pequeño y pasivo (sin *early stopping* ni selección de pesos por mejor validación), por lo que puede no ser representativo de la sismicidad chilena. Aunque se estratificó por presencia de residuales, la muestra de entrenamiento y validación podría no cubrir adecuadamente la distribución conjunta de duración, intensidad y presencia de drift, lo que se traduce en una brecha de generalización apreciable entre desempeño en validación y en prueba, y en una alta varianza de errores entre registros individuales. Además, la sensibilidad al operador de derivadas y al paso temporal afecta tanto la PSD como la penalización cinemática: al reconstruir  $\ddot{u}$  con diferencias finitas, pequeñas oscilaciones o ruido de alta

frecuencia en el desplazamiento se traducen en aceleraciones y fuerzas internas sistemáticamente más ruidosas, con un piso espectral elevado en el rango de altas frecuencias.

En tercer lugar, la función de pérdida total utiliza ponderaciones estáticas entre datos, restricción cinemática y restricción dinámica. En esta tesis se fijan pesos constantes  $\{\lambda_{\text{kin}}, \lambda_{\text{dyn}}\}$  durante todo el entrenamiento, pese a que la importancia relativa de cada término evoluciona con la época de entrenamiento: pesos dinámicos altos en etapas muy tempranas pueden frenar el aprendizaje de patrones básicos, mientras que pesos demasiado bajos en etapas tardías favorecen el sobreajuste a los datos. Las curvas de pérdida muestran precisamente descensos rápidos seguidos de oscilaciones en PhyLSTM y mesetas tempranas en PhyCNN, coherentes con un balance subóptimo de los términos de pérdida.

Finalmente, las arquitecturas PhyLSTM predicen la fuerza interna como salida de la red sin supervisión directa, es decir, sólo a través de la restricción dinámica. Esto genera un problema de identificabilidad: dado que la ecuación de movimiento se impone de forma aproximada y con derivadas numéricas, existen múltiples relaciones fuerza–desplazamiento que pueden satisfacer  $\mathcal{L}_{\text{dyn}}$  con un residuo pequeño pero ser físicamente incorrectas, lo que se refleja en la baja calibración de los lazos histeréticos. Finalmente, la evaluación privilegia NRMSE, NMAE, L2RE y L2RE<sub>2D</sub>, pero carece de métricas específicas de histéresis, de error de desplazamiento residual y de error de picos, que serían más sensibles precisamente a estos modos de fallo.

A corto plazo, conviene fortalecer el protocolo experimental: (i) normalizar de forma consistente  $\mathbf{a}_g$ ,  $\mathbf{u}$ ,  $\mathbf{v}$ ,  $\mathbf{a}$  y  $\mathbf{g}$  con escalas físicas y/o estadísticas robustas (normalización por registro para estabilizar gradientes y por base completa para coherencia global); (ii) pasar a una validación activa: ampliar el conjunto de validación con muestreo estratificado por duración e IMs, emplear *early stopping* y seleccionar pesos por mejor desempeño en validación; (iii) asegurar que la selección de entrenamiento cubra explícitamente duración, intensidad, baja frecuencia y presencia de residuales. Para la fuerza, cuando exista referencia numérica, incorporar supervisión directa o imponer consistencia fuerte entre la salida predicha y la calculada por la ley constitutiva, complementando esto con términos de pérdida que actúen de manera local sobre la histéresis. La ponderación entre términos de pérdida debiera ser adaptativa, por ejemplo, mediante GradNorm o esquemas de balance dinámico (Chen et al., 2018; Xiang et al., 2022; Wang et al., 2021; Bischof y Kraus, 2025). El manejo de secuencias largas puede mejorar con ventanas solapadas y transmisión de estados, y con operadores de derivación de mayor orden o regularizados, con el fin de reducir la amplificación de ruido asociada a la reconstrucción de la fuerza interna. Además, es recomendable penalizar explícitamente el error de desplazamiento final y el promedio temporal de desplazamiento, así como incluir una pérdida sobre la coherencia fuerza–desplazamiento. Esta penalización adicional se justifica porque la pérdida dinámica, al imponer la ecuación de movimiento sólo de forma aproximada, controla mejor el contenido fuertemente dinámico que la componente cuasi-estática, de modo que el *drift* residual requiere un anclaje explícito en la función de pérdida.

## 5.4. Trabajo Futuro

Los avances logrados abren nuevas rutas de investigación para consolidar el uso de estas herramientas en la práctica profesional. Si bien los resultados obtenidos con la estrategia *Warm Start* son satisfactorios para una prueba de concepto ( $L2RE < 7\%$  en histéresis severa), existe margen para perfeccionar la precisión y robustez. Se identifican las siguientes rutas de mejora para trabajos futuros:

### 5.4.1. Extensión a Sistemas Exógenos y MDOF

El modelo RK4Net ha sido validado en régimen autónomo (vibración libre) para sistemas de un grado de libertad. El paso natural siguiente es extender la formulación para incluir la excitación sísmica  $\ddot{u}_g(t)$  como una entrada exógena dependiente del tiempo en la ecuación de estado. Asimismo, la generalización a sistemas de múltiples grados de libertad (MDOF) requerirá adaptar la arquitectura para manejar vectores de estado de mayor dimensión, explorando técnicas de reducción de orden o redes graph-based para mantener la eficiencia computacional. Como proyección, un *multi-task learning* con cabezas especializadas para  $\{u, \dot{u}, \ddot{u}, g\}$  podría coordinar objetivos y calibraciones, siguiendo la línea de trabajos como el de Cho et al. (2023) y Bonnet Eymard et al. (2024). Para secuencias largas, modelos de espacio de estados modernos o Transformers eficientes con atención lineal y recurrencia por segmentos (Katharopoulos et al., 2020), así como modelos Mamba basados en espacios de estados selectivos (Gu y Dao, 2024), prometen capturar dependencias de muy largo alcance con costo  $O(N)$  en lugar de  $O(N^2)$ , ofreciendo mayor estabilidad y eficiencia. De forma complementaria, los híbridos *solver-informed* que emulan integradores dentro de la celda recurrente (Sadeghi Eshkevari et al., 2021; Zhai et al., 2024) son claves para garantizar estabilidad. Finalmente, las formulaciones Neural ODE con estados latentes físicos habilitan esquemas *grey-box* útiles para identificación y predicción autoregresiva en condiciones no estacionarias (Chen et al., 2019; Lai et al., 2021).

### 5.4.2. Mejoras en la Arquitectura y Entrenamiento

Para robustecer aún más el desempeño en secuencias de muy larga duración, se sugiere:

- **Arquitecturas con Conexiones Residuales (ResNet):** Uno de los desafíos críticos en el entrenamiento de sistemas dinámicos recurrentes es el problema del desvanecimiento del gradiente (*vanishing gradient*) cuando las series temporales son extensas. Para mitigar esto, se propone sustituir el Perceptrón Multicapa simple (MLP) por una arquitectura interna basada en Bloques Residuales (*ResNet*). Las conexiones de salto (*skip connections*) permitirían que el gradiente fluya sin atenuación a través de las capas profundas y los pasos de tiempo, facilitando el aprendizaje de dinámicas de largo plazo sin perder estabilidad numérica.
- **Restricciones Físicas en la Función de Pérdida (Physics-Informed):** Siguiendo el paradigma de las PINNs y estrategias de actualización en tiempo real (Li y Sun, 2025), se puede enriquecer el entrenamiento penalizando la inconsistencia física. Una restricción clave sería forzar la compatibilidad cinemática, asegurando que la salida de velocidad de la red coincida exactamente con la derivada temporal

del desplazamiento predicho ( $\hat{v} \equiv d\hat{u}/dt$ ), un enfoque que ha mostrado mejorar la convergencia en sistemas no-lineales (Zhai et al., 2023).

- **Restricciones de Estado Final:** Se observó que el modelo tiende a acumular error hacia el final de la serie (equilibrio). Dado que el entrenamiento se realiza con datos conocidos, es posible agregar un término de pérdida que penalice fuertemente la desviación en el último paso de tiempo ( $t_{final}$ ). Esto actuaría como una condición de borde temporal (*shooting method*), obligando a la red a corregir las trayectorias que derivan lejos del equilibrio.
- **Refinamiento con Optimizadores de Segundo Orden:** El optimizador Adam es excelente para la exploración global, pero a menudo oscila cerca del mínimo. Una estrategia híbrida, que utilice un optimizador de segundo orden como **L-BFGS** en la etapa final del entrenamiento, podría refinar los pesos para reducir el error residual y ajustar con mayor precisión la curvatura de los lazos de histéresis.

### 5.4.3. Validación Experimental y Aplicaciones en Tiempo Real

Finalmente, es fundamental validar los modelos propuestos con datos experimentales provenientes de mesas vibratoras o ensayos híbridos. Esto permitirá evaluar la capacidad de la red para manejar ruido de medición real e incertidumbre en los parámetros. El bajo costo computacional de la inferencia con RK4Net la posiciona como una candidata ideal para aplicaciones de Simulación Híbrida en Tiempo Real (RTHS) y Monitoreo de Salud Estructural (SHM), donde la rapidez y la estabilidad son críticas.

# Referencias

- Amin, Mohammad y Alfredo H.-S. Ang (1968). Nonstationary Stochastic Models of Earthquake Motions. En: *Journal of the Engineering Mechanics Division* **94**, no. 2. \_eprint: <https://ascelibrary.org/doi/pdf/10.1061/JMCEA3.0000969>, págs. 559-584. doi: [10.1061/JMCEA3.0000969](https://doi.org/10.1061/JMCEA3.0000969).
- Arias, Arturo (1970). “A Measure of Earthquake Intensity”. En: *Seismic Design for Nuclear Power Plants*. Ed. por Robert J. Hansen. MIT Press, págs. 438-483. url: <https://cir.nii.ac.jp/crid/1570572699586425216>.
- Baber, Thomas T. y Mohammad N. Noori (ago. de 1985). Random Vibration of Degrading, Pinching Systems. en. En: *Journal of Engineering Mechanics* **111**, no. 8, págs. 1010-1026. doi: [10.1061/\(ASCE\)0733-9399\(1985\)111:8\(1010\)](https://doi.org/10.1061/(ASCE)0733-9399(1985)111:8(1010)).
- Baber, Thomas T. y Yi-Kwei Wen (dic. de 1981). Random Vibration of Hysteretic, Degrading Systems. En: *Journal of the Engineering Mechanics Division* **107**, no. 6, págs. 1069-1087. doi: [10.1061/JMCEA3.0002768](https://doi.org/10.1061/JMCEA3.0002768).
- Bas, Elif Ecem y Mohamed A. Moustafa (23 de oct. de 2020). Real-Time Hybrid Simulation with Deep Learning Computational Substructures: System Validation Using Linear Specimens. En: *Machine Learning and Knowledge Extraction* **2**, no. 4, págs. 469-489. doi: [10.3390/make2040026](https://doi.org/10.3390/make2040026).
- Bas, Elif Ecem, Mohamed A. Moustafa y Gokhan Pekcan (mar. de 2020). Compact Hybrid Simulation System: Validation and Applications for Braced Frames Seismic Testing. en. En: *Journal of Earthquake Engineering* **26**, no. 3, págs. 1565-1594. doi: [10.1080/13632469.2020.1733138](https://doi.org/10.1080/13632469.2020.1733138).
- Bengio, Yoshua, Jérôme Louradour, Ronan Collobert y Jason Weston (2009). Curriculum learning. En: *Proceedings of the 26th Annual International Conference on Machine Learning*. ICML '09. Montreal, Quebec, Canada: Association for Computing Machinery, págs. 41-48. doi: [10.1145/1553374.1553380](https://doi.org/10.1145/1553374.1553380).
- Bischof, Rafael y Michael A. Kraus (mayo de 2025). Multi-Objective Loss Balancing for Physics-Informed Deep Learning. en. En: *Computer Methods in Applied Mechanics and Engineering* **439**, pág. 117914. doi: [10.1016/j.cma.2025.117914](https://doi.org/10.1016/j.cma.2025.117914).
- Bonnet Eymard, Damien, Augustin Persoons, Matthias G.R. Faes y David Moens (nov. de 2024). Separable Physics-Informed Neural Networks for Robust Inverse Quantification in Solid Mechanics. en. En: Publisher: Zenodo. doi: [10.5281/ZENODO.14039660](https://doi.org/10.5281/ZENODO.14039660).
- Bouc, R. (1971). *Mathematical Model for Hysteresis*. Marseille, France: Centre de Recherches Physiques, págs. 16-25.
- Bouc, Rodney (1967). Forced Vibration of Mechanical Systems with Hysteresis. En: url: <https://api.semanticscholar.org/CorpusID:133694382>.

- Buckle, Ian, Matias Hube, Genda Chen, Wen-Huei Yen y Juan Arias (jun. de 2012). Structural Performance of Bridges in the Offshore Maule Earthquake of 27 February 2010. en. En: *Earthquake Spectra* **28**, no. 1S1, págs. 533-552. doi: [10.1193/1.4000031](https://doi.org/10.1193/1.4000031).
- Carrion, Juan y Billie Spencer (ene. de 2007). Model-based Strategies for Real-time Hybrid Testing. En. Castro, Sebastián, Roberto Benavente, Jorge G. F. Crempien, Gabriel Candia y Juan Carlos De La Llera (sep. de 2022). A Consistently Processed Strong-Motion Database for Chilean Earthquakes. en. En: *Seismological Research Letters* **93**, no. 5, págs. 2700-2718. doi: [10.1785/0220200336](https://doi.org/10.1785/0220200336).
- Charalampakis, Aristotelis (ene. de 2010). Parameters of Bouc–Wen hysteretic model revisited. en. En: 9th HSTAM International Congress on Mechanics.
- Charney, Finley A. (abr. de 2008). Unintended Consequences of Modeling Damping in Structures. En: *Journal of Structural Engineering* **134**, no. 4, págs. 581-592. doi: [10.1061/\(ASCE\)0733-9445\(2008\)134:4\(581\)](https://doi.org/10.1061/(ASCE)0733-9445(2008)134:4(581)).
- Chen, Ricky T. Q., Yulia Rubanova, Jesse Bettencourt y David Duvenaud (dic. de 2019). Neural Ordinary Differential Equations. en. arXiv:1806.07366 [cs]. doi: [10.48550/arXiv.1806.07366](https://doi.org/10.48550/arXiv.1806.07366).
- Chen, Youjun, Zeyang Sun, Ruiyang Zhang, Liuzhen Yao y Gang Wu (jun. de 2023). Attention mechanism based neural networks for structural post-earthquake damage state prediction and rapid fragility analysis. En: *Computers & Structures* **281**, pág. 107038. doi: [10.1016/j.compstruc.2023.107038](https://doi.org/10.1016/j.compstruc.2023.107038).
- Chen, Zhao, Vijay Badrinarayanan, Chen-Yu Lee y Andrew Rabinovich (jul. de 2018). GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks. en. arXiv:1711.02257 [cs]. doi: [10.48550/arXiv.1711.02257](https://doi.org/10.48550/arXiv.1711.02257).
- Cho, Junwoo et al. (oct. de 2023). Separable Physics-Informed Neural Networks. en. arXiv:2306.15969 [cs]. doi: [10.48550/arXiv.2306.15969](https://doi.org/10.48550/arXiv.2306.15969).
- Chopra, Anil K. (2017). Dynamics of structures: theory and applications to earthquake engineering. Fifth edition. Prentice-Hall international series in civil engineering and engineering mechanics. Hoboken, NJ: Pearson. 960 págs.
- Chung, J. y G. M. Hulbert (jun. de 1993). A Time Integration Algorithm for Structural Dynamics With Improved Numerical Dissipation: The Generalized- $\alpha$  Method. en. En: *Journal of Applied Mechanics* **60**, no. 2, págs. 371-375. doi: [10.1115/1.2900803](https://doi.org/10.1115/1.2900803).
- Clough, Ray W. y Joseph Penzien (1993). Dynamics of Structures. 2.<sup>a</sup> ed. McGraw-Hill.
- Craig, Roy R. y Andrew Kurdila (2006). Fundamentals of structural dynamics. en. 2nd ed. Hoboken, N.J.: John Wiley.
- Deierlein, Gregory G., Andrei M. Reinhorn y Michael R. Willford (2010). *Nonlinear Structural Analysis for Seismic Design*. NEHRP Seismic Design Technical Brief No. 4 NIST GCR 10-917-5. Produced by the NEHRP Consultants Joint Venture, a partnership of the Applied Technology Council and the Consortium of Universities for Research in Earthquake Engineering. Gaithersburg, MD: National Institute of Standards y Technology.
- Dong, Xiaohui, Zhenyun Tang y Xiuli Du (sep. de 2022). State of the art and development trends in numerical simulation for real-time hybrid simulation. En: *Earthquake Engineering and Resilience* **1**, no. 3. Publisher: Wiley, págs. 245-267. doi: [10.1002/eer2.25](https://doi.org/10.1002/eer2.25).

- Elnashai, Amr S. et al. (nov. de 2012). The Maule (Chile) earthquake of February 27, 2010: Development of hazard, site specific ground motions and back-analysis of structures. en. En: *Soil Dynamics and Earthquake Engineering* **42**, págs. 229-245. doi: [10.1016/j.soildyn.2012.06.010](https://doi.org/10.1016/j.soildyn.2012.06.010).
- Farea, Amer, Olli Yli-Harja y Frank Emmert-Streib (29 de ago. de 2024). Understanding Physics-Informed Neural Networks: Techniques, Applications, Trends, and Challenges. En: *AI* **5**, no. 3. Publisher: MDPI AG, págs. 1534-1557. doi: [10.3390/ai5030074](https://doi.org/10.3390/ai5030074).
- Farrar, Charles R y Keith Worden (feb. de 2007). An introduction to structural health monitoring. en. En: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **365**, no. 1851, págs. 303-315. doi: [10.1098/rsta.2006.1928](https://doi.org/10.1098/rsta.2006.1928).
- Fleuret, François (2024). The Little Book of Deep Learning. en. University of Geneva.
- Gora, Anita, Mingfeng Huang, Chunhe Wang y Ruoyu Zhang (2 de mayo de 2025). Wind-Induced Dynamic Performance Evaluation of Tall Buildings Considering Future Wind Climate. En: *Applied Sciences* **15**, no. 9. Publisher: MDPI AG, pág. 5073. doi: [10.3390/app15095073](https://doi.org/10.3390/app15095073).
- Gu, Albert y Tri Dao (mayo de 2024). Mamba: Linear-Time Sequence Modeling with Selective State Spaces. en. arXiv:2312.00752 [cs]. doi: [10.48550/arXiv.2312.00752](https://doi.org/10.48550/arXiv.2312.00752).
- Guo, Jia, Ryuta Enokida, Dawei Li y Kohju Ikago (jul. de 2023). Combination of physics-based and data-driven modeling for nonlinear structural seismic response prediction through deep residual learning. En: *Earthquake Engineering & Structural Dynamics* **52**, no. 8, págs. 2429-2451. doi: [10.1002/eqe.3863](https://doi.org/10.1002/eqe.3863).
- He, Kaiming, Xiangyu Zhang, Shaoqing Ren y Jian Sun (2016). Deep Residual Learning for Image Recognition. En: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, págs. 770-778. doi: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90).
- Hilber, Hans M., Thomas J. R. Hughes y Robert L. Taylor (jul. de 1977). Improved numerical dissipation for time integration algorithms in structural dynamics. en. En: *Earthquake Engineering & Structural Dynamics* **5**, no. 3, págs. 283-292. doi: [10.1002/eqe.4290050306](https://doi.org/10.1002/eqe.4290050306).
- Ibarra, Luis F., Ricardo A. Medina y Helmut Krawinkler (oct. de 2005). Hysteretic models that incorporate strength and stiffness deterioration. En: *Earthquake Engineering & Structural Dynamics* **34**, no. 12, págs. 1489-1511. doi: [10.1002/eqe.495](https://doi.org/10.1002/eqe.495).
- Kanai, Keiiti (1957). Semi-empirical formula for the seismic characteristics of the ground. En: *Bulletin of the Earthquake Research Institute* **35**, no. 2. Publisher: University of Tokyo, págs. 309-325.
- Karniadakis, George Em et al. (24 de mayo de 2021). Physics-informed machine learning. En: *Nature Reviews Physics* **3**, no. 6, págs. 422-440. doi: [10.1038/s42254-021-00314-5](https://doi.org/10.1038/s42254-021-00314-5).
- Katharopoulos, Angelos, Apoorv Vyas, Nikolaos Pappas y François Fleuret (ago. de 2020). Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention. en. arXiv:2006.16236 [cs]. doi: [10.48550/arXiv.2006.16236](https://doi.org/10.48550/arXiv.2006.16236).
- Lagaros, Nikos D. y Manolis Papadrakakis (feb. de 2012). Neural network based prediction schemes of the non-linear seismic response of 3D buildings. en. En: *Advances in Engineering Software* **44**, no. 1, págs. 92-115. doi: [10.1016/j.advengsoft.2011.05.033](https://doi.org/10.1016/j.advengsoft.2011.05.033).

- Lai, Zhilu, Charilaos Mylonas, Satish Nagarajaiah y Eleni Chatzi (mayo de 2021). Structural identification with physics-informed neural ordinary differential equations. En: *Journal of Sound and Vibration* **508**. MAG ID: 3162045419, pág. 116196. doi: [10.1016/j.jsv.2021.116196](https://doi.org/10.1016/j.jsv.2021.116196).
- Li, Huaguan y Chao Sun (mar. de 2025). Nonlinear time-varying system response modeling via a real-time updated Runge-Kutta physics-informed neural network. en. En: *Engineering Applications of Artificial Intelligence* **144**, pág. 110067. doi: [10.1016/j.engappai.2025.110067](https://doi.org/10.1016/j.engappai.2025.110067).
- Liao, Yuchen, Rong Lin, Ruiyang Zhang y Gang Wu (ene. de 2023). Attention-based LSTM (AttLSTM) neural network for Seismic Response Modeling of Bridges. en. En: *Computers & Structures* **275**, pág. 106915. doi: [10.1016/j.compstruc.2022.106915](https://doi.org/10.1016/j.compstruc.2022.106915).
- Liel, Abbie B., Curt B. Haselton, Gregory G. Deierlein y Jack W. Baker (mar. de 2009). Incorporating modeling uncertainties in the assessment of seismic collapse risk of buildings. En: *Structural Safety* **31**, no. 2. Publisher: Elsevier BV, págs. 197-211. doi: [10.1016/j.strusafe.2008.06.002](https://doi.org/10.1016/j.strusafe.2008.06.002).
- Liu, Fangyu, Junlin Li y Linbing Wang (oct. de 2023). PI-LSTM: Physics-informed long short-term memory network for structural response modeling. En: *Engineering Structures* **292**, pág. 116500. doi: [10.1016/j.engstruct.2023.116500](https://doi.org/10.1016/j.engstruct.2023.116500).
- Málaga-Chuquitaype, Christian (9 de feb. de 2022). Machine Learning in Structural Design: An Opinionated Review. En: *Frontiers in Built Environment* **8**, pág. 815717. doi: [10.3389/fbuil.2022.815717](https://doi.org/10.3389/fbuil.2022.815717).
- Mayergoyz, I. D. (1991). "The Classical Preisach Model of Hysteresis". En: *Mathematical Models of Hysteresis*. New York, NY: Springer New York, págs. 1-63. doi: [10.1007/978-1-4612-3028-1\\_1](https://doi.org/10.1007/978-1-4612-3028-1_1).
- Mccormick, Jason, Hiroshi Aburano, Masahiro Ikenaga y Masayoshi Nakashima (ene. de 2008). Permissible residual deformation levels for building structures considering both safety and human elements. En.
- Mekaoui, Nabil y Taiki Saito (mar. de 2022). A Deep Learning-Based Integration Method for Hybrid Seismic Analysis of Building Structures: Numerical Validation. en. En: *Applied Sciences* **12**, no. 7, pág. 3266. doi: [10.3390/app12073266](https://doi.org/10.3390/app12073266).
- Mottershead, J.E. y M.I. Friswell (oct. de 1993). Model Updating In Structural Dynamics: A Survey. en. En: *Journal of Sound and Vibration* **167**, no. 2, págs. 347-375. doi: [10.1006/jsvi.1993.1340](https://doi.org/10.1006/jsvi.1993.1340).
- Newmark, Nathan M. (jul. de 1959). A Method of Computation for Structural Dynamics. en. En: *Journal of the Engineering Mechanics Division* **85**, no. 3, págs. 67-94. doi: [10.1061/JMCEA3.0000098](https://doi.org/10.1061/JMCEA3.0000098).
- Nguyen, Hoang D., Nhan D. Dao y Myoungsu Shin (jul. de 2022). Machine learning-based prediction for maximum displacement of seismic isolation systems. en. En: *Journal of Building Engineering* **51**, pág. 104251. doi: [10.1016/j.jobe.2022.104251](https://doi.org/10.1016/j.jobe.2022.104251).
- Ning, Chunxiao, Yazhou Xie y Lijun Sun (jul. de 2023). LSTM, WaveNet, and 2D CNN for nonlinear time history prediction of seismic responses. En: *Engineering Structures* **286**. doi: [10.1016/j.engstruct.2023.116083](https://doi.org/10.1016/j.engstruct.2023.116083).
- Núñez, F, D Ruiz y J Cortés (2021). Nonlinear dynamic analysis of steel buildings subjected to earthquakes. Análisis dinámico no lineal de edificios de acero sometidos a sismos. En: **36**.
- Oh, Byung Kwan, Branko Glisic, Sang Wook Park y Hyo Seon Park (mar. de 2020). Neural network-based seismic response prediction model for building structures using artificial earthquakes. En: *Journal of Sound and Vibration* **468**. doi: [10.1016/j.jsv.2019.115109](https://doi.org/10.1016/j.jsv.2019.115109).

- PEER (2006). OpenSees. url: <https://opensees.berkeley.edu/>.
- Palacio-Betancur, Alejandro y Marianonieta Gutierrez Soto (abr. de 2022). Recent Advances in Computational Methodologies for Real-Time Hybrid Simulation of Engineering Structures. doi: [10.1007/s11831-022-09848-y](https://doi.org/10.1007/s11831-022-09848-y).
- Pellisetti, M.F. (10 de mayo de 2009). Parallel processing in structural reliability. En: *Structural Engineering and Mechanics* **32**, no. 1, págs. 95-126. doi: [10.12989/SEM.2009.32.1.095](https://doi.org/10.12989/SEM.2009.32.1.095).
- Phuvoravan, Kitjapat y Elisa D. Sotelino (abr. de 2005). Nonlinear Finite Element for Reinforced Concrete Slabs. En: *Journal of Structural Engineering* **131**, no. 4, págs. 643-649. doi: [10.1061/\(ASCE\)0733-9445\(2005\)131:4\(643\)](https://doi.org/10.1061/(ASCE)0733-9445(2005)131:4(643)).
- Pinho, Rui (2007). “Nonlinear Dynamic Analysis of Structures Subjected to Seismic Action”. En: *Advanced Earthquake Engineering Analysis*. Ed. por Alain Pecker. Vol. 494. Series Title: CISM International Centre for Mechanical Sciences. Vienna: Springer Vienna, págs. 63-89. doi: [10.1007/978-3-211-74214-3\\_5](https://doi.org/10.1007/978-3-211-74214-3_5).
- Raissi, Maziar, Paris Perdikaris y George Em Karniadakis (1 de feb. de 2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. En: *Journal of Computational Physics* **378**. MAG ID: 2899283552, págs. 686-707. doi: [10.1016/j.jcp.2018.10.045](https://doi.org/10.1016/j.jcp.2018.10.045).
- Sadeghi Eshkevari, Soheil, Martin Takáč, Shamim N. Pakzad y Majid Jahani (feb. de 2021). DynNet: Physics-based neural architecture design for nonlinear structural response modeling and prediction. En: *Engineering Structures* **229**, pág. 111582. doi: [10.1016/j.engstruct.2020.111582](https://doi.org/10.1016/j.engstruct.2020.111582).
- Salehi, Hadi y Rigoberto Burgueño (sep. de 2018). Emerging artificial intelligence methods in structural engineering. En: *Engineering Structures* **171**. Publisher: Elsevier BV, págs. 170-189. doi: [10.1016/j.engstruct.2018.05.084](https://doi.org/10.1016/j.engstruct.2018.05.084).
- Salgado, Rafael A. y Serhan Guner (oct. de 2018). A comparative study on nonlinear models for performance-based earthquake engineering. En: *Engineering Structures* **172**. Publisher: Elsevier BV, págs. 382-391. doi: [10.1016/j.engstruct.2018.06.034](https://doi.org/10.1016/j.engstruct.2018.06.034).
- Su, Cheng, Jingwei Liang y Zengsheng He (dic. de 2024). E-PINN: A fast physics-informed neural network based on explicit time-domain method for dynamic response prediction of nonlinear structures. En: *Engineering Structures* **321**. Publisher: Elsevier BV, pág. 118900. doi: [10.1016/j.engstruct.2024.118900](https://doi.org/10.1016/j.engstruct.2024.118900).
- Al-Subaihawi, Safwan, James Ricles, Spencer Quiel, Thomas Marullo y Faisal Malik (dic. de 2024). Real-time hybrid simulation of structural systems with soil-foundation interaction effects using neural networks. en. En: *Earthquake Engineering & Structural Dynamics* **53**, no. 15, págs. 4688-4718. doi: [10.1002/eqe.4236](https://doi.org/10.1002/eqe.4236).
- Sun, Han, Henry V. Burton y Honglan Huang (ene. de 2021). Machine learning applications for building structural design and performance assessment: State-of-the-art review. En: *Journal of Building Engineering* **33**, pág. 101816. doi: [10.1016/j.jobe.2020.101816](https://doi.org/10.1016/j.jobe.2020.101816).
- Tajimi, Hiroyuki (1960). A statistical method of determining the maximum response of a building structure during an earthquake. En: *Proceedings of the 2nd World Conference on Earthquake Engineering*. Vol. 2. Tokyo, Japan, págs. 781-797.
- Thai, Huu-Tai (abr. de 2022). Machine learning for structural engineering: A state-of-the-art review. En: *Structures* **38**, págs. 448-491. doi: [10.1016/j.istruc.2022.02.003](https://doi.org/10.1016/j.istruc.2022.02.003).

- Thundiyil, S., S.S. Shalamzari, J. Picone y S. McKenzie (2 de dic. de 2023). Transformers for Modeling Long-Term Dependencies in Time Series Data: A Review. En: *2023 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*. 2023 IEEE Signal Processing in Medicine and Biology Symposium (SPMB). Philadelphia, PA, USA: IEEE, págs. 1-5. doi: [10.1109/spmb59478.2023.10372632](https://doi.org/10.1109/spmb59478.2023.10372632).
- Vanluchene, R. D. y Roufei Sun (sep. de 1990). Neural Networks in Structural Engineering. En: *Computer-Aided Civil and Infrastructure Engineering* **5**, no. 3. Publisher: Wiley, págs. 207-215. doi: [10.1111/j.1467-8667.1990.tb00377.x](https://doi.org/10.1111/j.1467-8667.1990.tb00377.x).
- Vaswani, Ashish et al. (s.f.). Attention is All you Need. en. En: ().
- Vielma-Quintero, Juan Carlos, Jorge Carvallo y Juan Carlos Vielma (18 de jun. de 2024). Evaluación comparativa mediante diseño basado en el desempeño de un edificio estructurado en base a muros. En: *Obras y Proyectos*, no. 35. Publisher: Universidad Catolica de la Santisima Concepcion, págs. 76-84. doi: [10.21703/0718-2813.2024.35.2804](https://doi.org/10.21703/0718-2813.2024.35.2804).
- Wang, Sifan, Yujun Teng y Paris Perdikaris (ene. de 2021). Understanding and Mitigating Gradient Flow Pathologies in Physics-Informed Neural Networks. en. En: *SIAM Journal on Scientific Computing* **43**, no. 5, A3055-A3081. doi: [10.1137/20M1318043](https://doi.org/10.1137/20M1318043).
- Wang, Tianyu et al. (dic. de 2023a). From model-driven to data-driven: A review of hysteresis modeling in structural and mechanical systems. en. En: *Mechanical Systems and Signal Processing* **204**, pág. 110785. doi: [10.1016/j.ymsp.2023.110785](https://doi.org/10.1016/j.ymsp.2023.110785).
- Wang, Tianyu et al. (mar. de 2023b). Seismic response prediction of structures based on Runge-Kutta recurrent neural network with prior knowledge. en. En: *Engineering Structures* **279**, pág. 115576. doi: [10.1016/j.engstruct.2022.115576](https://doi.org/10.1016/j.engstruct.2022.115576).
- Wen, Yi-Kwei (abr. de 1976). Method for Random Vibration of Hysteretic Systems. En: *Journal of the Engineering Mechanics Division* **102**, no. 2, págs. 249-263. doi: [10.1061/JMCEA3.0002106](https://doi.org/10.1061/JMCEA3.0002106).
- Wilches, José et al. (ago. de 2021). Evolution of seismic design codes of highway bridges in Chile. en. En: *Earthquake Spectra* **37**, no. 3, págs. 2174-2204. doi: [10.1177/8755293020988011](https://doi.org/10.1177/8755293020988011).
- Wu, Rih-Teng y Mohammad R. Jahanshahi (1 de ene. de 2019). Deep Convolutional Neural Network for Structural Dynamic Response Estimation and System Identification. En: *Journal of Engineering Mechanics- asce* **145**, no. 1. MAG ID: 2898712259, pág. 04018125. doi: [10.1061/\(asce\)em.1943-7889.0001556](https://doi.org/10.1061/(asce)em.1943-7889.0001556).
- Xiang, Zixue, Wei Peng, Xu Liu y Wen Yao (jul. de 2022). Self-adaptive loss balanced Physics-informed neural networks. en. En: *Neurocomputing* **496**, págs. 11-34. doi: [10.1016/j.neucom.2022.05.015](https://doi.org/10.1016/j.neucom.2022.05.015).
- Xing, Chenxi, Zidong Xu y Hao Wang (ago. de 2024). Structural seismic responses prediction using the gradient-enhanced hybrid PINN. En: *Advances in Structural Engineering* **27**, no. 11. Publisher: SAGE Publications, págs. 1962-1970. doi: [10.1177/13694332241260140](https://doi.org/10.1177/13694332241260140).
- Xu, Kaikai et al. (mayo de 2024). Novel physics-informed neural network approach for dynamic and static displacement reconstruction via strain and acceleration. en. En: *Measurement* **231**, pág. 114588. doi: [10.1016/j.measurement.2024.114588](https://doi.org/10.1016/j.measurement.2024.114588).
- Xu, Zekun, Jun Chen, Jiaxu Shen y Mengjie Xiang (ene. de 2022). Recursive long short-term memory network for predicting nonlinear structural seismic response. En: *Engineering Structures* **250**. Publisher: Elsevier BV, pág. 113406. doi: [10.1016/j.engstruct.2021.113406](https://doi.org/10.1016/j.engstruct.2021.113406).

- Yinfeng, Dong, Li Yingmin, Lai Ming y Xiao Mingkui (abr. de 2008). Nonlinear structural response prediction based on support vector machines. en. En: *Journal of Sound and Vibration* **311**, no. 3-5, págs. 886-897. doi: [10.1016/j.jsv.2007.09.054](https://doi.org/10.1016/j.jsv.2007.09.054).
- Ying, Wang, Wang Chong, Li Hui y Zhao Renda (2009). Artificial Neural Network Prediction for Seismic Response of Bridge Structure. en. En: *2009 International Conference on Artificial Intelligence and Computational Intelligence*. Shanghai, China: IEEE, págs. 503-506. doi: [10.1109/AICI.2009.303](https://doi.org/10.1109/AICI.2009.303).
- Yu, Yang, Houpu Yao y Yongming Liu (nov. de 2020). Structural dynamics simulation using a novel physics-guided machine learning method. En: *Engineering Applications of Artificial Intelligence* **96**, pág. 103947. doi: [10.1016/j.engappai.2020.103947](https://doi.org/10.1016/j.engappai.2020.103947).
- Zhai, Weida, Yuequan Bao y Dongwang Tao (dic. de 2024). State space model-based Runge–Kutta gated recurrent unit networks for structural response prediction. En: *Nonlinear Dynamics* **112**, no. 24, págs. 21901-21921. doi: [10.1007/s11071-024-10229-2](https://doi.org/10.1007/s11071-024-10229-2).
- Zhai, Weida, Dongwang Tao y Yuequan Bao (nov. de 2023). Parameter estimation and modeling of nonlinear dynamical systems based on Runge–Kutta physics-informed neural network. En: *Nonlinear Dynamics* **111**, no. 22, págs. 21117-21130. doi: [10.1007/s11071-023-08933-6](https://doi.org/10.1007/s11071-023-08933-6).
- Zhang, Qingru, Dhananjay Ram, Cole Hawkins, Sheng Zha y Tuo Zhao (2023). Efficient Long-Range Transformers: You Need to Attend More, but Not Necessarily at Every Layer. En: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Findings of the Association for Computational Linguistics: EMNLP 2023. Singapore: Association for Computational Linguistics, págs. 2775-2786. doi: [10.18653/v1/2023.findings-emnlp.183](https://doi.org/10.18653/v1/2023.findings-emnlp.183).
- Zhang, Qingyu et al. (mar. de 2024). Transformer-based structural seismic response prediction. En: *Structures* **61**, pág. 105929. doi: [10.1016/j.istruc.2024.105929](https://doi.org/10.1016/j.istruc.2024.105929).
- Zhang, Ruiyang, Yang Liu y Hao Sun (jul. de 2020a). Physics-guided convolutional neural network (PhyCNN) for data-driven seismic response modeling. En: *Engineering Structures* **215**, pág. 110704. doi: [10.1016/j.engstruct.2020.110704](https://doi.org/10.1016/j.engstruct.2020.110704).
- (sep. de 2020b). Physics-informed multi-LSTM networks for metamodeling of nonlinear structures. En: *Computer Methods in Applied Mechanics and Engineering* **369**, pág. 113226. doi: [10.1016/j.cma.2020.113226](https://doi.org/10.1016/j.cma.2020.113226).
- Zhang, Ruiyang et al. (ago. de 2019). Deep long short-term memory networks for nonlinear structural seismic response prediction. En: *Computers & Structures* **220**, págs. 55-68. doi: [10.1016/j.compstruc.2019.05.006](https://doi.org/10.1016/j.compstruc.2019.05.006).
- Zhu, Mu, Yang Yang, James K. Guest y Michael D. Shields (2017). Topology optimization for linear stationary stochastic dynamics: Applications to frame structures. En: *Structural Safety* **67**, págs. 116-131. doi: <https://doi.org/10.1016/j.strusafe.2017.04.004>.
- Zhuang, Bohan et al. (ago. de 2023). A Survey on Efficient Training of Transformers. En: *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*. Thirty-Second International Joint Conference on Artificial Intelligence {IJCAI-23}. Macau, SAR China: International Joint Conferences on Artificial Intelligence Organization, págs. 6823-6831. doi: [10.24963/ijcai.2023/764](https://doi.org/10.24963/ijcai.2023/764).

## Apéndice A

# Recursos Computacionales y Reproducibilidad

Este anexo detalla la infraestructura computacional, los repositorios de código fuente y los entornos de software necesarios para replicar los experimentos presentados en esta investigación. El proyecto se divide en dos repositorios distintos: uno destinado a la evaluación comparativa de modelos existentes (Capítulo 3) y otro dedicado a la implementación de la arquitectura propuesta RK4Net (Capítulo 4).

### Repositorios de Código Fuente

El código fuente, los conjuntos de datos procesados y los cuadernos de experimentación se encuentran disponibles públicamente en la plataforma GitHub bajo las siguientes direcciones:

- **Fase I - Comparación de Modelos (Capítulo 3):**  
Implementación de PhyCNN, PhyLSTM y DeepLSTM.  
<https://github.com/MatiasCambaraN/Dev-PINNs-for-Nonlinear-Dynamic-Structural-Response>
- **Fase II - Arquitectura Propuesta (Capítulo 4):**  
Implementación de la arquitectura **RK4Net** (SeismicPINN).  
<https://github.com/MatiasCambaraN/seismicPINN>

### Infraestructura de Hardware

Los experimentos se llevaron a cabo utilizando dos entornos de hardware distintos. El entrenamiento intensivo de los modelos del Capítulo 3 se realizó principalmente en una estación de trabajo de escritorio (Desktop PC) con sistema operativo Linux, debido a la optimización de librerías antiguas de TensorFlow. El desarrollo y evaluación de la arquitectura RK4Net (Capítulo 4) se realizó en un equipo portátil (Laptop) bajo ambiente Windows.

La Tabla A.1 detalla las especificaciones técnicas de ambos equipos.

Tabla A.1: Especificaciones de la infraestructura de hardware utilizada.

<b>Componente</b>	<b>Equipo A: Laptop (Desarrollo y Cap. 4)</b>	<b>Equipo B: Desktop PC (Entrenamiento Cap. 3)</b>
<b>Sistema Operativo</b>	Windows 11 Home (v10.0.22631)	Ubuntu 24.04.2 LTS (Noble Numbat)
<b>Kernel/Build</b>	-	Linux 6.8.0-60-generic
<b>CPU</b>	Intel Core i7-9750H @ 2.60GHz (6/12 núcleos)	Intel Core i5-12400 (6/12 núcleos)
<b>Memoria RAM</b>	16 GB (2x 8 GB)	16 GB
<b>GPU Principal</b>	NVIDIA GeForce GTX 1050 (4 GB)	NVIDIA GeForce RTX 3060 (12 GB)
<b>GPU Secundaria</b>	Intel UHD Graphics 630 (1 GB)	-
<b>Drivers GPU</b>	CUDA 10.0 (v10.0.130) / cuDNN 7.4.1	CUDA 12.0 (v12.0.140) / cuDNN 9.2.0

## Apéndice B

# Fundamentos del Entrenamiento de Redes Neuronales

### Algoritmo de Retropropagación (Backpropagation)

#### Motivación Teórica

El entrenamiento de una red neuronal consiste en minimizar una función de pérdida  $\mathcal{L}$  ajustando los pesos sinápticos  $w$ . El desafío radica en calcular eficientemente el gradiente:

$$\nabla_w \mathcal{L} = \frac{\partial \mathcal{L}}{\partial w_{jk}^{(l)}}.$$

En una red profunda, la influencia de un peso en la primera capa sobre el error final es indirecta y pasa a través de múltiples funciones de activación anidadas.

Consideremos una cadena de dependencias:  $w_1 \rightarrow u_1 \rightarrow z_1 \rightarrow u_2 \cdots \rightarrow \mathcal{L}$ . Aplicando la regla de la cadena repetidamente, observamos una estructura recursiva:

$$\frac{\partial \mathcal{L}}{\partial u_1} = \frac{\partial \mathcal{L}}{\partial u_2} \cdot \frac{\partial u_2}{\partial z_1} \cdot \frac{\partial z_1}{\partial u_1}.$$

Esto revela que para calcular los gradientes de la capa  $l$ , necesitamos conocer los gradientes de la capa  $l + 1$ . Por lo tanto, el cálculo eficiente se realiza en reversa: desde la salida hacia la entrada.

#### Ejemplo Ilustrativo Paso a Paso

Para concretar la teoría, analicemos una red mínima con una entrada  $x$ , dos capas ocultas unitarias y una salida  $\hat{y}$ . La pérdida es el Error Cuadrático Medio (MSE):  $\mathcal{L} = \frac{1}{2}(\hat{y} - y)^2$ .

##### 1. Paso hacia adelante (Forward Pass):

$$\begin{aligned} u_1 &= w_1 x + b_1, & z_1 &= \sigma(u_1), \\ u_2 &= w_2 z_1 + b_2, & z_2 &= \sigma(u_2), \\ u_3 &= w_3 z_2 + b_3, & \hat{y} &= u_3. \end{aligned}$$

**2. Paso hacia atrás (Backward Pass):** Definimos el *error local*  $\delta_l = \frac{\partial \mathcal{L}}{\partial u_l}$  para simplificar la notación.

■ **Capa de Salida (3):**

$$\delta_3 = \frac{\partial \mathcal{L}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial u_3} = (\hat{y} - y) \cdot 1.$$

Gradiente:  $\frac{\partial \mathcal{L}}{\partial w_3} = \delta_3 \cdot z_2.$

■ **Capa Oculta (2):** Propagamos el error  $\delta_3$  a través de  $w_3$  y la derivada de la activación  $\sigma'$ :

$$\delta_2 = \delta_3 \cdot w_3 \cdot \sigma'(u_2).$$

Gradiente:  $\frac{\partial \mathcal{L}}{\partial w_2} = \delta_2 \cdot z_1.$

■ **Capa Oculta (1):**

$$\delta_1 = \delta_2 \cdot w_2 \cdot \sigma'(u_1).$$

Gradiente:  $\frac{\partial \mathcal{L}}{\partial w_1} = \delta_1 \cdot x.$

## Conclusión

La retropropagación no es más que la aplicación sistemática de la regla de la cadena utilizando programación dinámica (almacenando los  $\delta$  intermedios). Esto permite calcular el gradiente de todos los parámetros en una sola pasada hacia atrás, haciendo viable el entrenamiento de redes profundas.