



UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
TESIS DE MAESTRÍA

**Determinación de Entradas en Redes Neuronales
Convolucionales para la Detección en Tiempo Real
de Reductores de Velocidad Usando Sensores
Inerciales**

Autor:

María Fernanda López Barajas

Director de Tesis:

Julio César Rodríguez Quiñonez

Co-Director de Tesis:

Moisés Jesús Castro Toscano

*Tesis presentada para cumplir con los requisitos
para el grado de Maestría en Ciencias*

a el área de

Instrumentación y mediciones automáticas

Facultad de Ingeniería
Campus Mexicali

06 de junio del 2025

Declaración de Autoría

Yo, María Fernanda López Barajas, certifico que esta tesis titulada «Determinación de Entradas en Redes Neuronales Convolucionales para la Detección en Tiempo Real de Reductores de Velocidad Usando Sensores Inerciales », y el trabajo presentado en ella son de mi autoría. Declaro que:

- Este trabajo educativo fue realizado íntegramente mientras estuve inscrito a un programa académico de esta Universidad.
- Cualquier parte de esta tesis presentada previamente para otro grado o reconocimiento está claramente identificada.
- Todas las referencias a trabajos previos han sido citadas y atribuidas correctamente.
- Salvo las citas indicadas, el contenido es original y de mi autoría.
- He reconocido todas las fuentes de apoyo relevantes.
- En colaboraciones, he especificado mi aporte y el de otros.

Firmado:

Fecha:

«No podemos resolver nuestros problemas con el mismo pensamiento que usamos cuando los creamos.»

Albert Einstein

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

Resumen

Facultad de Ingeniería
Campus Mexicali

Maestría en Ciencias

Determinación de Entradas en Redes Neuronales Convolucionales para la Detección en Tiempo Real de Reductores de Velocidad Usando Sensores Inerciales

por María Fernanda López Barajas

Esta tesis de maestría presenta un sistema de detección en tiempo real de reductores de velocidad, utilizando una Red Neuronal Convolutiva unidimensional (1D-CNN) y datos de aceleración obtenidos de una Unidad de Medición Inercial (IMU). A diferencia de los enfoques existentes, que no operan en tiempo real debido a un desfase de entre 2 y 5 segundos, el sistema propuesto ofrece una respuesta inmediata y precisa.

La metodología se centró en optimizar las estructuras de entrada de la CNN utilizando ventanas de datos de 3×308 , donde las filas representan los ejes de aceleración (x, y, z) y las columnas contienen muestras de series temporales. Un enfoque de ventana deslizante con incrementos de 7 muestras permitió capturar completamente la interacción vehículo-tope. El análisis comparativo de técnicas de normalización ($[0,1]$, $[-1,1]$ y $[0,255]$) reveló que la normalización en escala de grises $[0-255]$ fue la más efectiva para la extracción de características.

La validación experimental incluyó pruebas extensivas en condiciones reales con un Honda Civic 2012 en las calles urbanas de Mexicali bajo diversas condiciones de tráfico. El modelo final alcanzó una precisión de clasificación del 99.61 % con una latencia de $<10\text{ms}$, manteniendo un rendimiento robusto a través de variaciones de velocidad y escenarios de conducción complejos (frenado, giros).

Reconocimientos

A la Secretaría de Ciencia, Humanidades, Tecnología e Innovación (SECIH-TI), por brindar el apoyo económico para financiar la investigación y el trabajo, y al constante apoyo con becas para poder continuar con los estudios de posgrados en México.

A la Universidad Autónoma de Baja California (UABC), por la oportunidad para estudiar un posgrado y obtener el grado de Maestría en Ciencias. A la Facultad de Ingeniería de la UABC, por las aulas, laboratorios y profesores para la formación académica de estudios de posgrado.

Al Dr. Julio César Rodríguez Quiñonez, mi director de tesis, mi más sincero agradecimiento por brindarme la oportunidad de llevar a cabo este proyecto de investigación. Su apoyo durante todo mi trayecto académico ha sido fundamental, y su generosidad al compartir su vasto conocimiento y experiencia ha marcado una diferencia significativa en mi formación profesional.

Al Dr. Moisés Jesús Castro Toscano, por su colaboración con el Dr. Julio César Rodríguez Quiñonez en el desarrollo de este proyecto de investigación. Aprecio su tiempo, apoyo y los valiosos consejos que ha brindado, los cuales han sido esenciales para el avance de este trabajo.

A mi comité de tesis, conformado por el Dr. Daniel Hernández Balbuena, el Dr. Fabian Natanael Murrieta Rico y la Dra. Wendy Flores Fuentes, mi más sincero agradecimiento por su constante apoyo y valiosa retroalimentación durante cada semestre, especialmente en las presentaciones de avance de tesis. Su orientación ha sido importante para el desarrollo y mejora de este trabajo.

Índice general

Declaración de Originalidad	I
Resumen	III
Reconocimientos	IV
1. Introducción	1
1.1. Antecedentes	3
1.2. Planteamiento del Problema	5
1.3. Justificación y Uso de los Resultados	7
1.4. Objetivos de la Investigación	9
1.4.1. Obejtivo General	9
1.4.2. Objetivos Específicos	9
1.5. Hipótesis	10
1.6. Metodología	13
2. Marco Teórico	15
2.1. Introducción al Marco Teórico	15
2.2. Reductores de Velocidad	15
2.3. Sistemas de Unidad de Medición Inercial (IMU)	19
2.4. Modelo Matemático de un Acelerómetro	21
2.4.1. Ecuación de Movimiento en Tres Dimensiones	21
2.4.2. Mediciones de Aceleración en Tres Ejes	22
2.4.3. Representación Vectorial	23
2.5. Acelerómetros MEMS	24
2.5.1. Modelo Mecánico del Acelerómetro MEMS	24
2.5.2. Principios de Funcionamiento de los Acelerómetros MEMS	25
Acelerómetros Piezoeléctricos	26
Acelerómetros Capacitivos	29
2.6. MPU 9255	30
2.7. Acelerómetro del MyRIO-1900	32
2.8. Redes Neuronales Convolucionales (CNN)	35

2.8.1.	Fundamentos Matemáticos de una CNN-1D	35
2.8.2.	Arquitectura de una CNN en 1D	36
	Capa de Convolución	36
2.8.3.	Funciones de Activación	37
2.8.4.	Capa de Pooling	38
2.8.5.	Capa Fully Connected	39
2.9.	Preprocesamiento de Señales Inerciales para la Detección de Reductores de Velocidad	40
3.	Procedimiento de Investigación	44
3.1.	Selección de Hardware y Software para la Adquisición de Datos	45
3.1.1.	Selección y Evaluación del Acelerómetro	46
	Protocolo de Comunicación para la MPU-9255	47
	Protocolo de Comunicación para el Acelerómetro del MyRIO	49
	Programa en LabVIEW para la Adquisición y Recopilación de Datos de Ambos Sensores Inerciales	49
3.1.2.	Evaluación del Acelerómetro del MyRIO vs el MPU-9255	50
3.2.	Organización y Estructuración de los Datos	52
3.3.	Preprocesamiento de los Datos	57
3.4.	Conjunto de Datos de Entrenamiento	58
3.5.	Diseño de la Red Neuronal Convolutiva Unidimensional en Python	59
3.5.1.	Entrenamiento y Compilación de la CNN-1D en Python	60
3.6.	Desarrollo de Programas en LabVIEW	62
3.6.1.	Programa en LabVIEW para la Adquisición de Señales	62
3.6.2.	Clasificación de Reductores de Velocidad en LabVIEW Mediante una CNN-1D en Python	65
3.6.3.	Optimización del Umbral de Decisión	69
4.	Experimentos y Análisis de Resultados	71
4.1.	Experimentación con Diferentes Modelos	71
4.1.1.	Modelos Iniciales y Análisis de sus Limitaciones	71
	Modelo 1	71
	Modelo 2	72
	Modelo 3	72
	Modelo 4	72
4.1.2.	Modelos Mejorados con Base en Resultados Previos	73
	Modelo 5	73

Modelo 6	73
4.2. Evaluación del Impacto de la Normalización en cada Modelo	74
4.3. Análisis de Robustez del Modelo 6	75
4.3.1. Proceso de Entrenamiento	75
4.3.2. Evaluación del Desempeño	76
4.4. Análisis de las Ventanas de Entrada en la CNN-1D para Detec- ción de Reductores de Velocidad	77
4.4.1. Configuración de las Ventanas de Entrada	77
4.4.2. Visualización Temporal de las Señales	77
4.4.3. Representación en Escala de Grises	79
4.5. Análisis del Tiempo de Ventaneo en Clasificación en Tiempo Real	81
5. Conclusiones	83
5.1. Conclusiones Generales	83
5.2. Cumplimiento de Objetivos	84
5.2.1. Recolección de Datos de Aceleración Desde una IMU .	84
5.2.2. Diseño e Implementación del Preprocesamiento de Datos	84
5.2.3. Identificación de Características Relevantes para la CNN	85
5.2.4. Metodología para Estructurar las Entradas Adecuadas para la CNN	85
5.2.5. Entrenamiento y Validación del Modelo CNN	85
5.2.6. Integración del Sistema en Tiempo Real	85
5.3. Consideraciones Finales	85
5.4. Recomendaciones y Trabajo a Futuro	86
Bibliografía	88

Índice de figuras

1.1. Diagrama de métodos basados en vibraciones en sistemas de SHM.	2
1.2. Vehículo Honda Civic modelo 2012 pasando por un reductor de velocidad en la ciudad de Mexicali, Baja California.	7
1.3. Metodología para la implementación de un sistema de clasificación de reductores de velocidad utilizando LabVIEW, Python y redes neuronales convolucionales.	14
2.1. Diseño de reductor de velocidad. (a) Plástico prefabricado, (b) En forma trapezoidal.	16
2.2. Geometría y parámetros de diseño de tres tipos de reductores de velocidad.	17
2.3. Diagrama de entrada y salida de un sistema IMU típico.	20
2.4. Representación de un acelerómetro triaxial en un sistema de coordenadas xyz.	22
2.5. Modelo masa-resorte-amortiguador de un acelerómetro.	25
2.6. Circuito equivalente de un acelerómetro piezoeléctrico. (a) Modelo de carga. (b) Modelo de voltaje.	27
2.7. Modelo capacitivo de un acelerómetro.	30
2.8. Diagrama de bloques de un MPU 9255.	31
2.9. Diagrama de bloques del hardware NI MyRIO-1900.	32
2.10. Diagrama del proceso de convolución unidimensional en una red neuronal.	36
2.11. Comportamiento de las funciones de activación ReLU, sigmoide y tanh.	38
2.12. Operación de Max Pooling de dimensión 2x2 en una matriz de entrada. Cada región de la ventana deslizante selecciona el valor máximo para construir la matriz de salida.	39
2.13. Esquema de una red neuronal convolucional (CNN) en el que se lleva a cabo la extracción de características.	40

3.1. Diagrama de flujo del programa en LabVIEW para la adquisición y recopilación de datos de ambos sensores inerciales.	50
3.2. Distribución normal de los datos del eje z: comparación entre el acelerómetro MyRIO y el MPU 9255.	51
3.3. Segmentación de la señal de aceleración en el eje z, donde W_2 representa la ventana correspondiente al <i>Reductor de velocidad</i> . Las demás ventanas contienen datos previos y posteriores, etiquetados como <i>Nada</i>	52
3.4. Segmentación con ventanas deslizantes, donde cada W_i captura información previa y actual para mejorar la detección del reductor de velocidad en tiempo real.	53
3.5. Segmentación con ventanas deslizantes, donde cada W_i captura información previa y actual para mejorar la detección del reductor de velocidad en tiempo real.	55
3.6. Estructura del arreglo utilizado como entrada para el entrenamiento de la CNN-1D.	57
3.7. Diagrama del modelo de red neuronal convolucional 1D (CNN-1D) con una capa convolucional, una capa densa intermedia y una capa de salida para clasificación binaria.	60
3.8. Código en Python para la compilación, entrenamiento y almacenamiento de un modelo de red neuronal convolucional 1D.	61
3.9. Estructura del productor en LabVIEW, donde se adquieren las señales del acelerómetro del MyRIO y se convierten a unidades de m/s^2	63
3.10. SubVI encargado de la conversión de aceleraciones de g a m/s^2 en LabVIEW.	63
3.11. Estructura del productor en LabVIEW, donde las aceleraciones se visualizan, se guardan con su etiqueta y se envían en tiempo real a otro programa mediante variables compartidas.	64
3.12. Panel frontal en LabVIEW para adquisición de señales de aceleración en los ejes x, y, z . Incluye gráficas por eje, indicadores numéricos, botones de guardado y parada.	65
3.13. Diagrama del ciclo productor en LabVIEW. Muestra la inicialización de colas, lectura de la variable compartida y envío de datos de aceleración en tiempo real.	66
3.14. Arquitectura consumidor en LabVIEW: formación de ventanas temporales traslapadas (3×308) para alimentar la CNN-1D, con actualización cada 8.75 ms.	67

3.15. Arquitectura del ciclo de clasificación. (1) Recepción de matriz 3×308 , (2) Invocación del modelo Python (Algoritmo 3), (3) Decisión binaria basada en umbral, y (4) Visualización de resultados.	68
3.16. Interfaz de usuario: visualización de datos en tiempo real y resultado de la clasificación.	69
3.17. Curva ROC del Modelo 6. El punto óptimo (rojo) en $\theta = 0,97$ maximiza la diferencia entre sensibilidad y especificidad.	70
4.1. Curvas de pérdida para entrenamiento y validación del Modelo 6.	76
4.2. Matriz de confusión para el conjunto de prueba ($n=2,587$).	77
4.3. Señal temporal al inicio de un reductor. Los últimos 7 muestreos contienen los primeros indicios del evento.	78
4.4. Señal temporal al finalizar un reductor. Donde aproximadamente 140 puntos pertenecen al evento de pasar el reductor de velocidad.	78
4.5. Señal temporal de referencia sin presencia de reductores. La amplitud constante muestra el patrón característico de conducción normal.	79
4.6. Representación en escala de grises de la Figura 4.3. Esta señal corresponde al evento de comenzar a pasar el reductor de velocidad.	79
4.7. Representación en escala de grises de la Figura 4.4. Esta señal corresponde al evento de estar terminando de pasar el reductor de velocidad.	80
4.8. Representación en escala de grises de la Figura 4.5. Esta señal corresponde al evento denominado <i>Nada</i> , donde la ventana no contiene ninguna aceleración asociada al paso por un reductor de velocidad.	80

Índice de Tablas

3.1. Comparación de la media, desviación estándar y error cuadrático medio para las aceleraciones medidas en los ejes x , y y z de los sensores MyRIO y MPU-9255.	51
3.2. Duración promedio del cruce sobre un reductor de velocidad con una frecuencia de muestreo de 800 Hz.	54
3.3. Ejemplo de segmentación y etiquetado de ventanas para un reductor de velocidad.	56
3.4. Distribución de etiquetas en los conjuntos de datos de entrenamiento.	59
4.1. Comparación del desempeño de los diferentes modelos con diferentes técnicas de normalización.	74
4.2. Métricas de evaluación del Modelo 6.	76

Esta tesis de maestría está dedicada a...

A mis padres, mi hermana, familiares y amigos, por su apoyo incondicional y por estar siempre a mi lado, brindándome su cariño, comprensión y fortaleza a lo largo de este proceso.

A mi esposo, Genaro Dávalos, mi mayor inspiración para emprender este camino. Gracias por tu respaldo incondicional, por acompañarme en tantas noches de desvelo, por tu fortaleza emocional y por compartir conmigo tus valiosos conocimientos. Tu presencia ha sido fundamental en cada paso de este recorrido.

Capítulo 1

Introducción

Los sistemas de Monitoreo de Salud Estructural (SHM, por sus siglas en inglés) han sido un tema relevante de investigación desde la década de 1990 Güemes, 2013. El propósito principal de estos sistemas es monitorear la integridad de diversas estructuras para optimizar costos y garantizar un mantenimiento adecuado. Estos sistemas se han aplicado, por ejemplo, en la supervisión continua de carreteras, el monitoreo de las condiciones de aeronaves y la evaluación de la calidad estructural en edificios.

En sus primeras aplicaciones, los SHM se basaban en métodos convencionales que empleaban enfoques numéricos, como la correlación de frecuencias Carden y Fanning, 2004. Aunque el análisis en el dominio de la frecuencia es el más común, también se han desarrollado métodos para realizar análisis en el dominio del tiempo Hou y Xia, 2021; Carden y Fanning, 2004. Entre los enfoques utilizados se encuentran los métodos de frecuencia natural, los basados en la forma modal, la curvatura de la forma modal y los métodos no lineales. Sin embargo, estos métodos convencionales presentan limitaciones significativas, ya que suelen ser susceptibles a errores humanos. Esto se debe a que muchos de ellos requieren tanto la extracción manual de características como la resolución manual de ecuaciones matemáticas para monitorear la salud de las estructuras Hou y Xia, 2021; Moaveni et al., 2013; Chen y Bicanic, 2000.

Con el avance de la inteligencia artificial (IA), se han desarrollado nuevos sistemas SHM más robustos y eficientes. Las técnicas de IA han demostrado ser especialmente efectivas en la extracción automática de características y en la realización de clasificaciones precisas. Estas tecnologías han sido particularmente útiles en la detección de daños estructurales, empleando modelos como Redes Neuronales Artificiales (ANN), Máquinas de Soporte Vectorial (SVM), Bosque Aleatorio (RF) y Redes Neuronales Convolucionales (CNN). Estos modelos, cada vez más integrados en los sistemas SHM, ofrecen ventajas significativas en términos de precisión, exactitud y automatización.

Esta evolución metodológica se resume en la Figura 1.1, la cual presenta los principales métodos basados en vibraciones utilizados en sistemas SHM,

agrupados en dos categorías: métodos numéricos y métodos basados en inteligencia artificial.



FIGURA 1.1: Diagrama de métodos basados en vibraciones en sistemas de SHM.

Con los avances tecnológicos, el monitoreo estructural ha evolucionado hacia el uso de sensores más sofisticados y métodos automatizados. En particular, los sistemas de Unidad de Medición Inercial (IMU, por sus siglas en inglés) han demostrado ser herramientas versátiles para capturar datos dinámicos, como aceleraciones, que pueden ser utilizados no solo en aplicaciones estructurales, sino también en contextos relacionados con el análisis de la interacción vehículo-infraestructura. Este último aspecto resulta especialmente relevante para la identificación y clasificación de características en las vialidades, como los reductores de velocidad.

En este contexto, el presente trabajo se centra en el desarrollo de un método para estructurar y procesar adecuadamente los datos de aceleración obtenidos de un sensor Inercial. La aportación principal radica en que la segmentación de los datos en ventanas temporales bien definidas permitirá a una CNN clasificar de manera precisa los reductores de velocidad, alcanzando una exactitud superior al 90%. Este estudio no solo busca validar el uso de IMUs en aplicaciones vehiculares, sino también establecer una metodología en la preparación de datos para sistemas basados en inteligencia artificial, contribuyendo así al desarrollo de soluciones más robustas y confiables para la clasificación en tiempo real.

La implementación de técnicas basadas en inteligencia artificial, como las Redes Neuronales Convolucionales (CNN), ha abierto nuevas posibilidades para procesar y analizar grandes volúmenes de datos provenientes de sensores como los acelerómetros. Estas redes tienen la capacidad de aprender patrones complejos directamente de los datos, eliminando la necesidad de realizar un procesamiento manual extenso y reduciendo los posibles errores humanos. Sin embargo, un desafío clave en este enfoque es la determinación de los parámetros óptimos de entrada para maximizar el rendimiento de la CNN, asegurando tanto la precisión como la eficiencia del sistema.

1.1. Antecedentes

Una de las áreas de aplicación de los sistemas de SHM es el monitoreo de vialidades. En la actualidad, se han desarrollado diversos sistemas SHM que integran técnicas de IA con el objetivo de identificar y localizar irregularidades en las carreteras. Sin embargo, estos sistemas suelen requerir una alta capacidad de cómputo, lo que incrementa significativamente su costo (Bellloch Díaz, 2019). Para abordar esta limitación, se han propuesto alternativas más accesibles, aunque aún presentan áreas de mejora. Entre estas alternativas se incluyen los siguientes enfoques:

- El estudio de Celaya-Padilla et al., 2018 propone un sistema para la detección de reductores de velocidad utilizando datos provenientes de un giroscopio, un acelerómetro y un Sistema de Posicionamiento Global (GPS, por sus siglas en inglés) montados en un vehículo. Los datos fueron procesados mediante un algoritmo genético, que generó un modelo logístico multivariado con una exactitud del 97.14%. Sin embargo, el sistema no opera en tiempo real, ya que analiza ventanas de datos con una duración de 2 segundos, lo que genera un retraso equivalente en la detección. El autor también sugiere que el sistema podría simplificarse utilizando únicamente un sensor, aunque al intentarlo obtuvieron una exactitud menor. Por lo tanto, a futuro se propone mejorar las entradas del algoritmo de inteligencia artificial para permitir un funcionamiento eficiente con un solo sensor.
- El trabajo de Peralta-López et al., 2023 presenta un sistema basado en una red neuronal convolucional para detectar baches y reductores de velocidad en carreteras. Utiliza imágenes capturadas por una cámara estereoscópica ZED colocada en el frente de un SUV. Se creó una base

de datos con 714 imágenes que fueron etiquetadas manualmente. Se entrenó con el 70 % de las imágenes y se validó con el 30 %, logrando una exactitud del 98.13 %. Sin embargo el procesamiento es lento, lo que no permite a este sistema trabajar en tiempo real.

- García-Ramírez et al., 2022 realizó un sistema para la detección de irregularidades en vialidades, donde utiliza el Video VBOX Lite, que ofrece datos más precisos. Sin embargo, su alto costo lo hace menos accesible, y su exactitud comienza a disminuir cuando el vehículo supera los 40 km/h.
- Menegazzo y Wangenheim, 2022 propone un modelo para la detección de reductores de velocidad utilizando sensores inerciales, cuyo análisis se realiza mediante una red neuronal CNN-LSTM (Convolutional Neural Network - Long Short-Term Memory). Este modelo alcanzó una exactitud del 98.59 % y un f1-score de 94.7 % para la clase “Sin Reductor”, y 99.19 % para la clase “Con Reductor”. No obstante, el sistema podría presentar un mayor margen de error en la etiqueta “Sin Reductor” debido a las irregularidades en la vialidad que afectan la detección. Además, el uso combinado de las redes CNN y LSTM incrementa significativamente el costo computacional, lo que puede ser un desafío en entornos de procesamiento en tiempo real.
- Otro sistema desarrollado por el autor Galeano, Maldonado et al., 2019, es el de procesamiento de imágenes empleando un dron Phantom 4 para capturar imágenes a una altitud de 100 metros, obteniendo una Distancia de Muestreo del Suelo (GSD, por sus siglas en inglés) inferior a 3 cm. Estas imágenes se procesan para generar ortofotos, lo que permite una visualización en tiempo real de buena resolución y la clasificación de baches y grietas. No obstante, la presencia de parches o manchas en las vías puede ocasionar falsas detecciones durante la clasificación.
- El sistema propuesto por Cifuentes, Marulanda y Thomson, 2021, para la detección de grietas en el pavimento, basado en técnicas de procesamiento de imágenes y redes neuronales artificiales, alcanza una exactitud del 95.56 % y una precisión del 94.44 %. Sin embargo, requiere capturar fotografías con un dispositivo móvil, lo que introduce un factor de variabilidad debido a las condiciones de iluminación.
- El sistema desarrollado por Salman y Mian, 2023 utiliza sensores de smartphones para detectar y clasificar tipos de reductores de velocidad

(sinusoidales, planos, termoplásticos y redondeados), alcanzando una exactitud del 98.92 %. Aunque actualmente no funciona en tiempo real, el futuro trabajo busca implementar un modelo pre-entrenado para clasificar los reductores de velocidad en tiempo real.

- Asimismo, se ha desarrollado un sistema basado en IoT por Ugale et al., 2021, que utiliza un microcontrolador, una IMU, un GPS, una fuente de energía y un módulo Wi-Fi. Este sistema presenta, a través de una aplicación, el estado de las calles en tiempo real, clasificándolas en dos categorías mediante colores:
 - Rojo: calles con irregularidades, como baches, reductores de velocidad, montículos, lodazales, superficies rugosas u obstáculos.
 - Verde: calles en buen estado.

Aunque el trabajo de Ugale et al., 2021 proporciona una solución eficiente, presenta algunas desventajas, como la dependencia de la nube y una exactitud limitada en la clasificación.

A pesar de las ventajas que ofrecen los sistemas actuales para la detección de reductores de velocidad, irregularidades en las vialidades y calles en mal estado, todos ellos presentan áreas de mejora que limitan su aplicación en escenarios reales. Las principales limitaciones incluyen la dependencia de condiciones externas como la iluminación, el alto costo de los equipos, la capacidad limitada de clasificación y la generación de falsas detecciones

Gracias a los avances en los sistemas SHM, centrados en el análisis de vialidades, es posible desarrollar e implementar soluciones cada vez más robustas y accesibles. En particular, la integración de sensores inerciales y tecnologías IoT (Internet of Things, por sus siglas en inglés) ha permitido mejorar la capacidad de capturar datos de alta precisión en tiempo real, detectando reductores de velocidad de manera más eficiente. La combinación de estas tecnologías con enfoques como las redes neuronales ofrece un panorama prometedor para abordar los desafíos actuales y avanzar hacia sistemas más completos y funcionales.

1.2. Planteamiento del Problema

En el contexto de los sistemas de monitoreo de salud estructural (SHM) aplicados al análisis vial, surge la necesidad de abordar los desafíos relacionados con el mantenimiento y la gestión de calles en zonas geográficas con

condiciones extremas. Mexicali, Baja California, México, es un ejemplo de una ciudad que enfrenta problemáticas específicas en su infraestructura vial, derivadas de su clima cálido y seco, así como de su ubicación en una zona sísmica activa. Esta ciudad se encuentra ubicada entre las latitudes $30^{\circ} 51'$ y $32^{\circ} 44'$ norte, y las longitudes $114^{\circ} 43'$ y $115^{\circ} 41'$ oeste. En este escenario, la implementación de soluciones tecnológicas avanzadas cobra especial relevancia, particularmente en el manejo y procesamiento de datos generados por sensores inerciales.

Los sensores inerciales, como los acelerómetros, producen datos crudos que presentan retos significativos, entre ellos el ruido y la variabilidad en las condiciones de adquisición. Si estos problemas no se abordan adecuadamente, pueden limitar la eficacia de los algoritmos de inteligencia artificial (IA), como las redes neuronales convolucionales (CNN), afectando la precisión y exactitud de los resultados.

El rendimiento de una CNN depende en gran medida de la calidad y estructura de las entradas que recibe. Por ello, procesar y estructurar de manera adecuada los datos de aceleración obtenidos de los sensores que integran una IMU se convierte en un aspecto crucial. Este proceso implica filtrar el ruido, minimizar las distorsiones y seleccionar las características más relevantes para optimizar el desempeño del modelo.

En este contexto, se plantea investigar la estructura óptima de los datos provenientes del acelerómetro para alimentar una CNN y lograr clasificaciones precisas y exactas. Aunque el caso de estudio aborda la clasificación de reductores de velocidad como aplicación práctica, el enfoque radica en desarrollar una metodología que permita preprocesar y estructurar los datos crudos, minimizando las limitaciones inherentes al uso de sensores inerciales. Con esta investigación, se pueden sentar las bases para desarrollar sistemas más robustos, capaces de clasificar diferentes irregularidades sobre las vialidades, basándose en el método desarrollado.

Asimismo, el análisis incluye la propuesta de un sistema de monitoreo para la identificación y clasificación de reductores de velocidad, que podría facilitar la integración de mapas inteligentes para optimizar las rutas de viaje, considerando las características del terreno. Esto contribuiría a reducir el consumo de combustible y la contaminación ambiental.

La implementación de esta tecnología tiene el potencial de transformar la gestión de la infraestructura urbana en Mexicali, integrando tecnologías de IoT y aprendizaje automático para optimizar el monitoreo vial y mejorar la calidad de las vialidades. Sin embargo, para que una CNN alcance un

desempeño adecuado en la clasificación de reductores de velocidad, es fundamental definir cómo se procesan y organizan los datos de aceleración antes de ser utilizados como entrada en el modelo.

Dado lo anterior, surge la siguiente pregunta de investigación:

- ¿Qué técnicas de preprocesamiento y organización de datos de acelerómetros permiten mejorar el desempeño de una CNN para obtener una exactitud superior al 90 % en la clasificación de reductores de velocidad?



FIGURA 1.2: Vehículo Honda Civic modelo 2012 pasando por un reductor de velocidad en la ciudad de Mexicali, Baja California.

1.3. Justificación y Uso de los Resultados

La ciudad de Mexicali enfrenta importantes desafíos en el mantenimiento y la gestión de sus vialidades debido a una combinación de factores geográficos, climáticos y ambientales. Su ubicación en una región sísmica activa, el clima extremo con temperaturas que superan los 45°C y su condición como una de las ciudades más contaminadas del país aceleran el deterioro de su infraestructura vial. Este contexto afecta directamente la seguridad vial, la comodidad de los usuarios y la calidad de vida de los habitantes.

Dentro de las soluciones implementadas para mejorar la seguridad vial, los reductores de velocidad desempeñan un papel crucial al moderar el tráfico en zonas críticas. Sin embargo, su diseño inconsistente, falta de estandarización y, en algunos casos, instalación no autorizada generan problemáticas adicionales:

- **Riesgos para la seguridad vial:** Los reductores de velocidad mal diseñados o ubicados pueden ocasionar frenadas bruscas, colisiones y pérdida de control del vehículo.
- **Mayor desgaste vehicular:** El paso constante sobre reductores de velocidad mal construidos incrementa el desgaste de neumáticos, suspensión y otros componentes del vehículo, resultando en costos de reparación elevados.
- **Impacto negativo en la calidad de vida:** Los reductores de velocidad mal diseñados generan ruido excesivo, vibraciones y molestias en las comunidades aledañas.
- **Deterioro ambiental:** Las constantes aceleraciones y frenadas aumentan las emisiones de partículas contaminantes y gases de efecto invernadero.
- **Problemas de salud pública:** La exposición prolongada al ruido y la contaminación generada por los reductores de velocidad afecta la salud respiratoria y mental de los habitantes cercanos.
- **Colocación ilegal:** En muchos casos, los reductores de velocidad se instalan sin autorización, incumpliendo normas oficiales y aumentando los riesgos de accidentes.

Ante estas limitaciones, el desarrollo de sistemas tecnológicos basados en sistemas IMUs y algoritmos de aprendizaje automático representa una solución innovadora. Los sensores inerciales, como los acelerómetros, generan grandes volúmenes de datos que, si se estructuran y procesan adecuadamente, pueden ser utilizados para entrenar redes neuronales convolucionales (CNN) capaces de clasificar reductores de velocidad con alta precisión y exactitud.

Este proyecto se enfoca en la estructuración de los datos de entrada para entrenar una CNN, utilizando información proveniente de acelerómetros y abordando desafíos como el ruido y la variabilidad en las condiciones de

adquisición. La metodología propuesta busca garantizar que la exactitud del sistema supere el 90 %, ofreciendo una herramienta eficiente y confiable para la gestión vial.

Además, esta investigación no solo se enfoca en resolver el problema específico de la clasificación de reductores de velocidad, sino que también establece las bases para seguir desarrollando el sistema, garantizando una clasificación exacta que pueda adaptarse a diversos contextos y ampliando su capacidad para clasificar diferentes irregularidades en las vialidades.

Los resultados obtenidos tienen aplicaciones prácticas directas en la mejora de la gestión vial. Un sistema capaz de clasificar reductores de velocidad con alta precisión, exactitud y en tiempo real puede permitir a las autoridades priorizar el mantenimiento de vialidades, optimizar el diseño de estas estructuras y garantizar que cumplan con los estándares establecidos. Además, los datos generados podrían integrarse en plataformas de monitoreo urbano inteligente, apoyando la planificación de rutas más eficientes y reduciendo costos asociados al desgaste vehicular y emisiones contaminantes.

1.4. Objetivos de la Investigación

1.4.1. Obejtivo General

Determinar la estructura adecuada de las entradas para una red neuronal convolucional (CNN) utilizando datos de aceleración provenientes de un sensor inercial, con el fin de lograr una clasificación precisa de los reductores de velocidad. El objetivo es mejorar el preprocesamiento y la segmentación de los datos para alcanzar una precisión superior al 90 % en la clasificación en tiempo real.

1.4.2. Objetivos Específicos

- **Recolección de datos de aceleración desde una IMU:** Configurar y calibrar un sistema de adquisición de datos que adquiera las aceleraciones (A_x , A_y y A_z) de la IMU, para detectar los cambios en la superficie de las vialidades relacionados con los reductores de velocidad, asegurando una medición precisa y exacta en tiempo real.
- **Diseño e implementación del preprocesamiento de datos:** Desarrollar métodos para normalizar las señales y segmentar los datos en ventanas

temporales, asegurando su preparación adecuada como entradas para la red neuronal convolucional (CNN).

- **Identificación de características relevantes para la CNN:** Analizar los datos preprocesados para identificar y seleccionar las características clave que permitan maximizar la capacidad de la CNN para distinguir reductores de velocidad con precisión y exactitud.
- **Metodología para estructurar las entradas adecuadas para la CNN:** Evaluar diferentes formas de representar los datos de aceleración, así como ajustar diferentes parámetros de configuración en una CNN para determinar la estructura más efectiva como entrada al modelo de clasificación.
- **Entrenamiento y validación del modelo CNN:** Diseñar y entrenar diferentes modelos de red neuronal convolucional utilizando las entradas adecuadas, escogiendo el modelo con mejor exactitud, validando su rendimiento en la clasificación de reductores de velocidad con una exactitud superior al 90 %.
- **Integración del sistema en tiempo real:** Implementar el modelo de CNN junto con el sistema de adquisición de datos en una plataforma que permita la clasificación continua de reductores de velocidad en tiempo real, evaluando su desempeño en condiciones reales.

1.5. Hipótesis

Se plantea que la clasificación de reductores de velocidad mediante una red neuronal convolucional (CNN) depende en gran medida de la correcta estructuración de las entradas a partir de los datos de aceleración obtenidos por un sensor inercial. En el desarrollo del sistema se espera enfrentar los siguientes desafíos:

- **Determinación de entradas adecuadas para la CNN:** El reto principal radica en encontrar la mejor forma de estructurar los datos de aceleración recogidos por el sensor inercial para ser utilizados como entradas en la red neuronal. Se investigarán diferentes representaciones, como el uso de ventanas temporales, con el fin de mejorar la precisión y exactitud de la clasificación.

- **Filtrado y procesamiento de datos:** Los datos de la IMU pueden verse afectados por ruidos o interferencias. Por ello, será necesario implementar técnicas de preprocesamiento que optimicen la calidad de los datos sin perder información relevante para la clasificación de los reductores de velocidad.
- **Entrenamiento de la CNN:** El éxito de la red neuronal dependerá de la calidad del conjunto de datos de entrenamiento y de la capacidad para ajustar los parámetros de la CNN. Será necesario realizar iteraciones y pruebas continuas para encontrar la mejor configuración para la clasificación de los reductores de velocidad.
- **Etiquetado adecuado del conjunto de datos:** Asegurar que los datos de entrenamiento estén correctamente etiquetados es esencial para entrenar la red neuronal de manera eficiente. Un etiquetado adecuado permitirá que la CNN aprenda de manera precisa a reconocer los reductores de velocidad en las vialidades.
- **Integración y comunicación entre programas:** Un desafío clave será lograr una comunicación eficiente y estable entre los programas desarrollados en Python, LabVIEW y el sistema MyRIO. Esto incluye garantizar la adquisición continua de datos en tiempo real desde el sensor inercial, mientras se ejecuta paralelamente el proceso de clasificación mediante la CNN, manteniendo la sincronización y el rendimiento óptimo en ambas tareas.

A pesar de los desafíos anticipados, para lograr una clasificación precisa y en tiempo real con una exactitud superior al 90 %, se propone un enfoque que abarca desde la recolección de datos hasta la integración del sistema en tiempo real. A continuación, se presentan las hipótesis específicas correspondientes a cada uno de los objetivos planteados:

- **Recolección de datos de aceleración desde la IMU:** Se plantea que la correcta configuración y calibración del sistema de adquisición de datos de la IMU permitirá obtener aceleraciones (A_x , A_y y A_z) precisas y exactas en tiempo real, facilitando la detección de cambios en la superficie de las vialidades relacionados con los reductores de velocidad.
- **Diseño e implementación del preprocesamiento de datos:** Se propone que, mediante el desarrollo de métodos adecuados para la normalización y segmentación de los datos en ventanas temporales, se podrá

garantizar que las señales de aceleración sean preparadas de forma eficiente como entradas para la CNN, obteniendo un mejor rendimiento del modelo.

- **Identificación de características relevantes para la CNN:** Se plantea que el análisis de los datos preprocesados permitirá identificar las características clave que optimizarán la capacidad de la CNN para diferenciar reductores de velocidad con alta precisión, mejorando la efectividad de la clasificación.
- **Metodología para estructurar las entradas adecuadas para la CNN:** Se propone que evaluar diferentes representaciones de los datos de aceleración y ajustar los parámetros de configuración de la CNN permitirá determinar la estructura más adecuada para alimentar al modelo, mejorando su desempeño en la clasificación.
- **Entrenamiento y validación del modelo CNN:** Se plantea que el entrenamiento y validación de modelos de CNN con las entradas estructuradas adecuadamente, seleccionando el modelo con mayor exactitud, resultará en un sistema de clasificación de reductores de velocidad con una exactitud superior al 90 %, capaz de operar eficazmente en tiempo real.
- **Integración del sistema en tiempo real:** Se propone que la integración del modelo de CNN con el sistema de adquisición de datos en una plataforma de tiempo real permitirá la clasificación continua de reductores de velocidad, proporcionando resultados precisos y fiables bajo condiciones reales de operación.

La hipótesis general del proyecto plantea que, para lograr una clasificación precisa y en tiempo real de los reductores de velocidad con una exactitud superior al 90 %, es fundamental abordar de manera efectiva los desafíos relacionados con la estructuración de los datos, el preprocesamiento y el entrenamiento de la red neuronal convolucional (CNN). La correcta recolección y calibración de los datos de aceleración obtenidos por el sensor inercial, junto con un adecuado diseño de las entradas para la CNN y un proceso de filtrado y preprocesamiento eficiente, son elementos clave para garantizar un rendimiento óptimo del modelo. Asimismo, la identificación precisa de las características relevantes en los datos y la correcta integración de los sistemas de adquisición y clasificación en tiempo real permitirán la implementación exitosa de un sistema capaz de operar bajo condiciones reales

con una alta precisión, dejando las bases para seguir desarrollando esta investigación y extendiéndola a la clasificación de otro tipo de irregularidades en las vialidades.

1.6. Metodología

En la Figura 1.3 se presenta un esquema de la metodología propuesta para este proyecto. De manera general, el proceso inicia con la definición del hardware necesario para la adquisición de datos provenientes de un acelerómetro. Se desarrolló un programa en LabVIEW que permite obtener lecturas de aceleración en tiempo real utilizando un MyRIO, lo que facilita la experimentación en campo y el almacenamiento de los datos obtenidos.

Posteriormente, se determinó la manera óptima de estructurar las aceleraciones como entradas para una red neuronal convolucional (CNN). Este paso incluyó la generación de diferentes conjuntos de datos (datasets) a partir de ventanas temporales de las aceleraciones recolectadas, los cuales se utilizaron para entrenar diversas CNN. El objetivo fue identificar tanto la estructura de entrada más adecuada como el modelo de CNN con la mayor exactitud de clasificación.

Una vez entrenada la CNN seleccionada, se creó una función en Python que permite cargar y utilizar el modelo guardado. Además, se desarrolló un programa en LabVIEW que establece comunicación con Python, lo que posibilita invocar la CNN desde LabVIEW. El sistema completo involucra tres programas que trabajan de forma sincronizada en tiempo real sobre un vehículo en marcha:

1. **Adquisición de datos:** Un programa en LabVIEW comunica el MyRIO con el acelerómetro, capturando las lecturas de aceleración en tiempo real.
2. **Procesamiento de datos:** Otro programa en LabVIEW organiza las lecturas en ventanas temporales, preparándolas como entradas para la CNN.
3. **Clasificación en tiempo real:** La función en Python realiza las predicciones basadas en las entradas procesadas, devolviendo a LabVIEW la probabilidad de que los datos pertenezcan a la clase “Reductor”. La clasificación se muestra en una interfaz gráfica en LabVIEW.

Finalmente, se llevó a cabo experimentación en campo para evaluar la precisión y exactitud del sistema bajo condiciones reales. Los resultados obtenidos fueron analizados y documentados para su publicación.

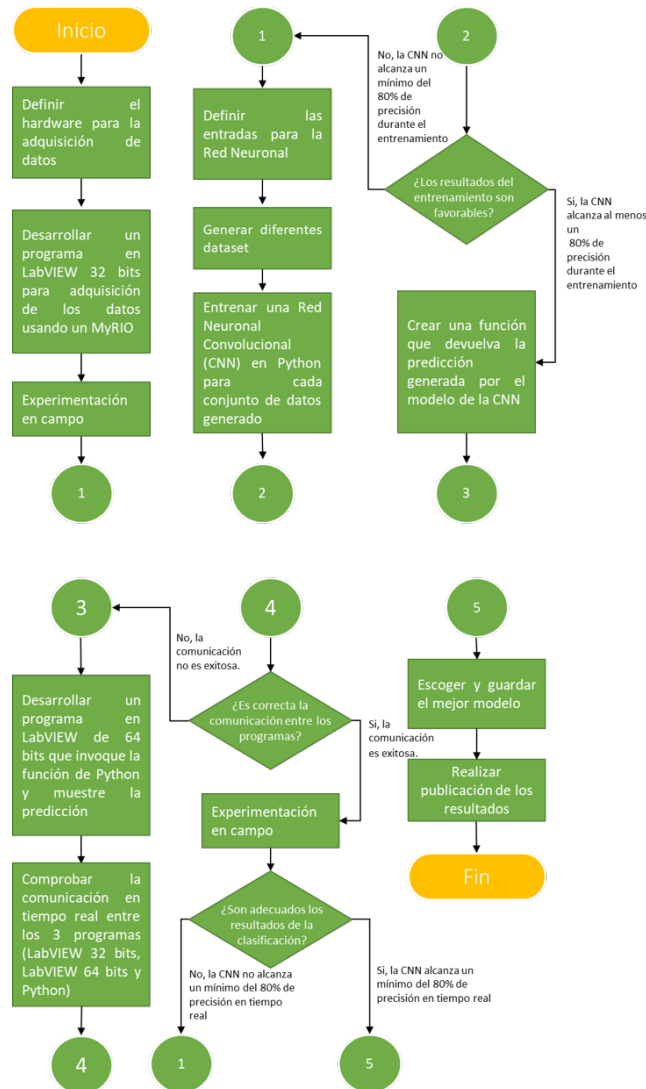


FIGURA 1.3: Metodología para la implementación de un sistema de clasificación de reductores de velocidad utilizando LabVIEW, Python y redes neuronales convolucionales.

Capítulo 2

Marco Teórico

2.1. Introducción al Marco Teórico

El desarrollo de este marco teórico tiene como objetivo establecer los conceptos fundamentales y el estado del arte necesarios para abordar la clasificación de reductores de velocidad utilizando redes neuronales convolucionales (CNN). Se presenta una revisión de los sistemas de Unidad de Medición Inercial (IMU), el procesamiento de señales y los enfoques modernos en redes neuronales aplicadas a la clasificación de patrones, con énfasis en la representación adecuada de los datos para una CNN de una dimensión.

2.2. Reductores de Velocidad

Los reductores de velocidad son estructuras diseñadas para disminuir la velocidad de los vehículos, ya que el exceso de velocidad es uno de los principales factores que contribuyen a los accidentes automovilísticos. Su uso ha incrementado significativamente debido a su eficacia inmediata, su bajo costo de implementación y su carácter pasivo, al no requerir supervisión constante. Sin embargo, estas estructuras también generan efectos negativos, como el desgaste en la suspensión de los vehículos y el aumento de partículas contaminantes (ppm) en el aire debido a las constantes aceleraciones y desaceleraciones de los vehículos que circulan sobre ellos.

A pesar de su efectividad en la reducción de velocidad, su instalación y mantenimiento suelen carecer de una regulación adecuada. En muchos casos, se instalan sin autorización gubernamental, lo que genera problemas adicionales, como su ubicación en zonas de alto tránsito donde obstaculizan la circulación y ocasionan complicaciones viales. La falta de normativas claras sobre su diseño y mantenimiento agrava esta situación (Goenaga y Fuentes, 2020).

Baltrėnas, Janušėvičius y Chlebnikovas, 2017 analizó dos tipos de reductores de velocidad, los trapezoidales y los circulares de plástico, y encontró que la concentración de partículas en el aire aumenta entre un 18 % y un 80 %

al paso de los vehículos. Para mitigar este efecto, se recomienda instalar dos reductores de velocidad en secuencia, evitando que los conductores aceleren después del primer reductor de velocidad y reduciendo así la emisión de contaminantes. La Figura 2.1 muestra las medidas de los tipos de reductores de velocidad utilizados en dicha investigación.

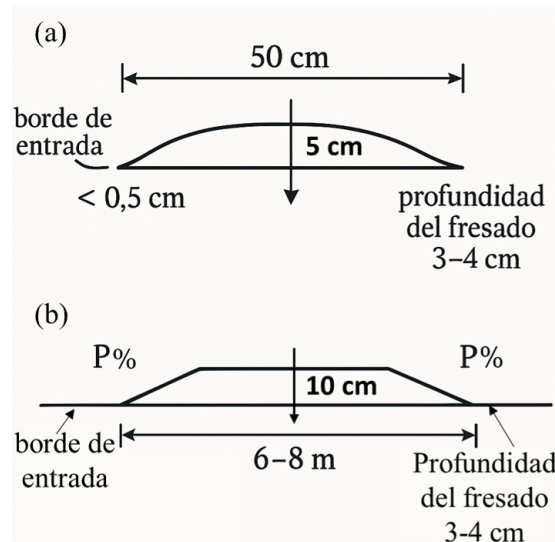


FIGURA 2.1: Diseño de reductor de velocidad. (a) Plástico pre-fabricado, (b) En forma trapezoidal.

Los reductores de velocidad se clasifican principalmente según su longitud, altura y ancho, características que determinan su diseño y funcionalidad. La geometría del reductor de velocidad influye directamente en la aceleración experimentada por los ocupantes del vehículo, por lo que es necesario un análisis matemático riguroso para su optimización. Baltrėnas, Januševičius y Chlebnikovas, 2017 en su investigación menciona los principales tipos de reductores de velocidad:

- **Sinusoidal:** Presenta perfiles sinusoidales, armónicos o armónicos modificados.
- **Trapezoidal:** Consta de una sección plana en la parte superior y rampas de entrada y salida con perfiles sinusoidales o armónicos.
- **Polinomial:** Su perfil se ajusta a una curva polinomial para mejorar la transición del vehículo y reducir el impacto en la suspensión.

Estos tres tipos reductores de velocidad se ilustran en la Figura 2.2, donde los parámetros geométricos principales que los caracterizan incluyen:

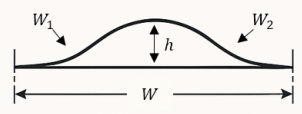
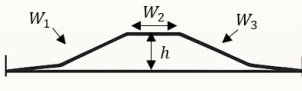
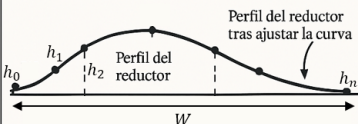
Tipo de Reductor	Forma del Reductor	Parámetros De Diseño
Sinusoidal		h= altura W ₁ = longitud de subida W ₂ = longitud de bajada
Trapezoidal	Perfiles de ascenso y descenso del reductor de velocidad: • Sinusoidal • Armónico • Armónico modificado 	h= altura W ₁ = longitud de subida W ₂ = longitud de la parte superior W ₃ = longitud de bajada
Polinomial		W= longitud del reductor h _i = altura i= 1, 2, ...,n

FIGURA 2.2: Geometría y parámetros de diseño de tres tipos de reductores de velocidad.

- h : Altura del reductor de velocidad.
- W : Longitud total del reductor de velocidad.
- W_1 W_2 : Longitud de las rampas de subida y bajada.
- W_3 : Longitud de la superficie plana (en caso de reductores de velocidad trapezoidales).

La respuesta del vehículo a un reductor de velocidad depende de su forma geométrica, la cual puede ser modelada a partir de una ecuación matemática para describir su forma, como la Ecuación 2.1 y 2.2.

Forma sinusoidal:

$$y(x) = \frac{h}{2} \left(1 - \cos \left(\frac{\pi x}{W_1} \right) \right), \quad 0 \leq x \leq W_1 \quad (2.1)$$

donde:

- $y(x)$: Altura del reductor de velocidad en función de la posición horizontal x .
- h : Altura máxima del reductor de velocidad.
- x : Posición horizontal a lo largo del reductor, medida desde el inicio de la rampa.

- W_1 : Anchura de la rampa de subida (o bajada) del reductor.
- $\cos\left(\frac{\pi x}{W_1}\right)$: Función coseno que define la forma suave y curva del perfil sinusoidal.

La Ecuación 2.1 describe un reductor de velocidad con un perfil suave y curvo, típico de los reductores de velocidad de tipo sinusoidal. La función coseno asegura que la transición entre la superficie plana y la parte más alta del reductor de velocidad sea gradual, lo que reduce el impacto en los vehículos y mejora la comodidad de los ocupantes.

Forma polinomial:

$$y(x) = h \sum_{i=1}^n a_i x^i, \quad 0 \leq x \leq W \quad (2.2)$$

- $y(x)$: Altura del reductor de velocidad en función de la posición horizontal x .
- h : Altura máxima del reductor de velocidad.
- x : Posición horizontal a lo largo del reductor de velocidad.
- W : Ancho total del reductor.
- a_i : Coeficientes del polinomio que definen la forma exacta del perfil.
- n : Grado del polinomio.

La Ecuación 2.2 describe un reductor de velocidad con un perfil definido por una curva polinomial. Los coeficientes a_i permiten ajustar la forma del reductor de velocidad para optimizar la transición del vehículo y minimizar el impacto en la suspensión. Este tipo de perfil es más flexible que el sinusoidal, ya que puede adaptarse a diferentes necesidades de diseño.

Baltrėnas, Januševičius y Chlebnikovas, 2017 plantea que la optimización del diseño del reductor de velocidad se basa en minimizar la diferencia entre la aceleración máxima percibida por el conductor y una aceleración deseada. La función se expresa como:

$$\min_{W_1, W_2, h} |a_{\max} - a_d| \quad (2.3)$$

donde a_{\max} es la aceleración máxima experimentada en la cabeza del conductor y a_d es el valor objetivo para garantizar confort. Las restricciones incluyen límites superiores e inferiores para W_1 , W_2 , h y para cumplir con normativas viales.

Para analizar el impacto de un reductor de velocidad en la dinámica vehicular, se modela el vehículo como un sistema de masas y resortes, utilizando la Ecuación de movimiento:

$$m\ddot{z} + c\dot{z} + kz = F_t \quad (2.4)$$

donde:

- m : Es la masa del vehículo.
- c : Es el coeficiente de amortiguamiento de la suspensión.
- k : Es la rigidez de la suspensión.
- $z(t)$: Es el desplazamiento vertical del vehículo.
- $F(t)$: Es la fuerza externa aplicada debido a la interacción con el reductor de velocidad.

El diseño de reductores de velocidad requiere un balance entre seguridad, confort y efectividad en la reducción de velocidad. Mediante modelos matemáticos y optimización numérica, es posible determinar las dimensiones y perfiles más adecuados para cada situación vial. La selección del tipo de reductor de velocidad y su geometría afecta directamente la respuesta del vehículo y la percepción del conductor, justificando el desarrollo de estrategias avanzadas de diseño y evaluación.

2.3. Sistemas de Unidad de Medición Inercial (IMU)

El desarrollo de los sistemas MEMS (Microelectromechanical Systems) en la década de los 80 marcó un punto clave en la evolución de los sensores inerciales (Samatas y Pachidis, 2022). Estos avances, impulsados por la nanotecnología, permitieron integrar componentes mecánicos y electrónicos en un único dispositivo compacto llamado Unidad de Medición Inercial.

Una IMU típica está compuesta por acelerómetros, giroscopios y, en algunos casos, magnetómetros como se muestra en la Figura 2.3. En conjunto, estos sensores permiten medir aceleraciones lineales, velocidades angulares y campos magnéticos, proporcionando datos esenciales para determinar la posición, orientación y movimiento de un objeto en el espacio tridimensional. Este nivel de integración y funcionalidad ha hecho que los sistemas IMU

sean fundamentales en aplicaciones como la navegación, la robótica, la industria automotriz y los dispositivos electrónicos portátiles (Sreevidya et al., 2018).

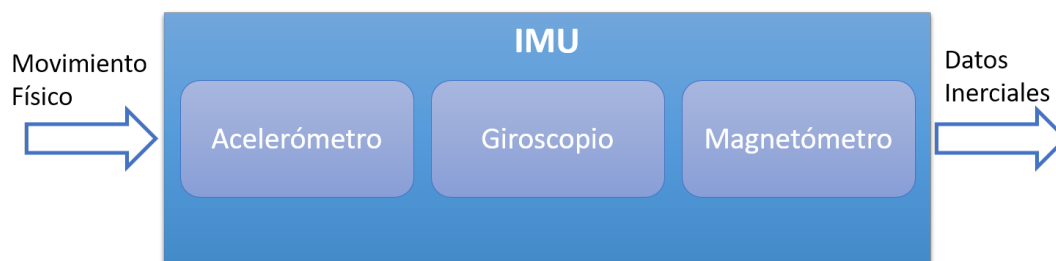


FIGURA 2.3: Diagrama de entrada y salida de un sistema IMU típico.

Los sistemas IMU ofrecen múltiples ventajas, como su bajo costo, rápida respuesta y la capacidad de operar sin depender de referencias satelitales externas. Cada sensor que compone una IMU, como el acelerómetro, el giroscopio y el magnetómetro, proporciona datos independientes, pero la combinación de esta información permite obtener resultados más precisos. Al integrar las mediciones de los diferentes sensores, es posible reducir errores individuales y aumentar la confiabilidad del sistema. Esta característica ha sido clave en aplicaciones como los sistemas SHM, donde las IMU han demostrado mejorar la robustez de los sistemas (Quispe et al., 2022).

Sin embargo, los datos generados por una IMU son datos en crudo y no son directamente útiles. Es imprescindible aplicar procesos como calibración, filtrado y normalización, además de operaciones avanzadas de procesamiento de señales, para convertir estas lecturas en información práctica. Por ejemplo, a partir de los datos del acelerómetro y giroscopio, se puede calcular la velocidad angular, estimar la orientación de un objeto en un espacio tridimensional y, en ciertos casos, determinar distancias recorridas. Estas capacidades hacen de las IMU una herramienta valiosa en diversas aplicaciones, como navegación inercial y sistemas autónomos.

A pesar de sus beneficios, los sistemas IMU presentan retos significativos. Uno de los principales problemas es el error acumulativo, también conocido como "drift" (Zhao y Wang, 2011). Este error se manifiesta con el tiempo debido a la integración repetida de datos, lo que puede derivar en mediciones inexactas, especialmente en trayectorias largas. Además, los magnetómetros pueden ser sensibles a interferencias electromagnéticas externas, lo que limita su fiabilidad en ciertos entornos.

Aunque las IMU tienen un enorme potencial gracias a su diseño compacto y sus capacidades versátiles, su implementación requiere un procesamiento cuidadoso para minimizar sus limitaciones y maximizar su efectividad en aplicaciones del mundo real.

2.4. Modelo Matemático de un Acelerómetro

Los acelerómetros son sensores inerciales utilizados para medir la aceleración lineal en uno o más ejes. La aceleración, definida como el cambio de velocidad con respecto al tiempo, es una cantidad vectorial expresada en metros por segundo al cuadrado (m/s^2) o en unidades de gravedad terrestre (g), donde :

$$1g = 9.81 \text{ m/s}^2 \quad (2.5)$$

El principio de funcionamiento de los acelerómetros se basa en la segunda ley de Newton (Stewart, 2008):

$$F = ma \quad (2.6)$$

donde F es la fuerza, m es la masa y a es la aceleración. La cual establece que la fuerza es proporcional a la masa y la aceleración. En un acelerómetro, una masa interna se encuentra suspendida por elementos elásticos o flexibles dentro del sensor. Cuando el dispositivo experimenta un cambio de velocidad, la inercia de la masa genera un desplazamiento, el cual es convertido en una señal eléctrica proporcional a la aceleración experimentada (Fajardo Vilorio, 2023).

Los acelerómetros pueden ser uniaxiales, biaxiales o triaxiales, dependiendo del número de ejes en los que pueden medir la aceleración. En la Figura 2.4 se muestran los tres ejes ortogonales de un acelerómetro triaxial, que permiten obtener información tridimensional del movimiento (Chapa et al., 2020). Además de medir aceleraciones debidas a fuerzas externas, también detectan la aceleración gravitacional, lo que permite determinar la inclinación del dispositivo con respecto a la Tierra.

2.4.1. Ecuación de Movimiento en Tres Dimensiones

El desplazamiento relativo de la masa suspendida en un acelerómetro triaxial está influenciado por la aceleración aplicada a lo largo de los tres

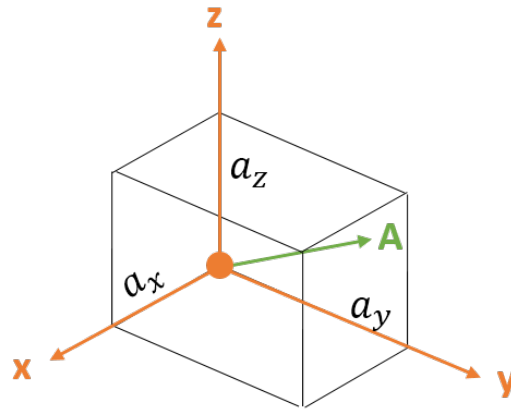


FIGURA 2.4: Representación de un acelerómetro triaxial en un sistema de coordenadas xyz.

ejes. La Ecuación general que describe el movimiento en los tres ejes x , y y z de un acelerómetro triaxial es la siguiente (Stewart, 2008):

$$m \cdot \ddot{x}_i(t) = -k_i \cdot x_i(t) \quad (2.7)$$

donde:

- $x_i(t)$ es el desplazamiento en el eje i (con $i \in \{x, y, z\}$),
- m es la masa de la suspensión,
- k_i es la constante de rigidez en el eje i ,
- $\ddot{x}_i(t)$ es la aceleración que se mide en cada eje.

Dado que un acelerómetro triaxial está compuesto por tres sensores orientados a lo largo de los tres ejes, el movimiento de la masa en cada eje se puede describir de forma independiente, pero las señales de cada eje pueden combinarse para obtener una medición completa del movimiento.

2.4.2. Mediciones de Aceleración en Tres Ejes

Cada acelerómetro triaxial proporciona tres señales de salida, $a_x(t)$, $a_y(t)$ y $a_z(t)$, que corresponden a las aceleraciones en las direcciones x , y y z , respectivamente. Estas señales son proporcionales a las aceleraciones en esas direcciones y se pueden expresar como (Stewart, 2008):

$$a_x(t) = \frac{1}{m} \cdot F_x(t) \quad (2.8)$$

$$a_y(t) = \frac{1}{m} \cdot F_y(t) \quad (2.9)$$

$$a_z(t) = \frac{1}{m} \cdot F_z(t) \quad (2.10)$$

donde:

- $F_x(t)$, $F_y(t)$, y $F_z(t)$ son las fuerzas inerciales actuando sobre el acelerómetro en los ejes x , y y z , respectivamente.

2.4.3. Representación Vectorial

Dado que las aceleraciones se miden en tres direcciones, la información de aceleración se puede representar de manera vectorial. El vector de aceleración total $\mathbf{A}(t)$ en el tiempo t se puede escribir como (Stewart, 2008):

$$\mathbf{A}(t) = \begin{pmatrix} a_x(t) \\ a_y(t) \\ a_z(t) \end{pmatrix} \quad (2.11)$$

Este vector describe el movimiento del acelerómetro en el espacio tridimensional. La magnitud total de la aceleración A se obtiene utilizando la siguiente Ecuación (Fajardo Vilorio, 2023):

$$|A| = \sqrt{a_x^2(t) + a_y^2(t) + a_z^2(t)} \quad (2.12)$$

donde:

- $|A|$ es la magnitud de la aceleración total.
- $a_x(t)$, $a_y(t)$ y $a_z(t)$ son las aceleraciones medidas en los ejes x , y y z , respectivamente.

Los acelerómetros triaxiales son herramientas fundamentales para medir la aceleración en tres dimensiones. Gracias a su capacidad para proporcionar información detallada sobre el movimiento en todas las direcciones espaciales. Existen diferentes tipos de acelerómetros, entre los cuales destacan los piezoeléctricos, capacitivos y MEMS (Castro Toscano, 2019).

2.5. Acelerómetros MEMS

Los Sistemas Microelectromecánicos (MEMS, por sus siglas en inglés) han revolucionado la detección de movimiento al integrar componentes mecánicos y electrónicos a microescala. Entre los dispositivos MEMS más destacados se encuentran los acelerómetros, sensores diseñados para medir la aceleración y detectar cambios en la velocidad o dirección de un objeto.

Estos acelerómetros se basan en una estructura con una masa suspendida. Cuando se aplica una aceleración externa, la masa genera una fuerza que deforma la estructura. Esta deformación es medida a nivel microscópico, generalmente en sustratos de silicio, permitiendo determinar la aceleración experimentada.

Más que un solo tipo de acelerómetro, los MEMS representan una técnica de fabricación que permite distintas formas de medición. Dependiendo del principio físico utilizado para detectar la deformación, existen dos tipos principales de acelerómetros MEMS:

- **Acelerómetros Capacitivos:** Miden la aceleración detectando variaciones en la capacitancia entre una masa móvil y una estructura fija. Al experimentar una aceleración, la masa móvil se desplaza, modificando la distancia entre las placas del condensador y, por lo tanto, su capacitancia. Esta variación se convierte en una señal eléctrica proporcional a la aceleración aplicada.
- **Acelerómetros Piezoeléctricos:** Funcionan mediante el efecto piezoeléctrico, que convierte una fuerza mecánica —provocada por la aceleración— en una carga eléctrica. Esta carga es proporcional a la magnitud de la fuerza aplicada sobre un material cristalino piezoeléctrico, y se traduce en una señal eléctrica utilizada para medir la aceleración.

Los acelerómetros MEMS pueden medir la aceleración en uno o más ejes, y su funcionamiento varía según el principio de detección empleado. Estas diferencias tecnológicas influyen en su precisión, sensibilidad y aplicación, por lo que la elección del tipo de acelerómetro depende del entorno y los requisitos de medición.

2.5.1. Modelo Mecánico del Acelerómetro MEMS

El comportamiento dinámico de los acelerómetros, independientemente de su principio de funcionamiento, puede describirse a través de un modelo mecánico basado en un sistema masa-resorte-amortiguador. Este modelo

resulta útil para representar la respuesta física del sensor ante excitaciones externas. Aunque existen diferentes tipos de acelerómetros, como los piezoeléctricos, capacitivos y MEMS, todos comparten el principio fundamental de un sistema mecánico similar, donde una masa se mueve en respuesta a una aceleración y está unida a una base por un resorte y un amortiguador.

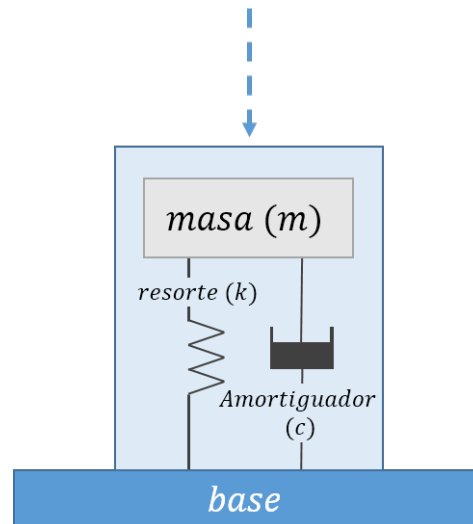


FIGURA 2.5: Modelo masa-resorte-amortiguador de un acelerómetro.

En la Figura 2.5, se muestra un modelo simplificado de un acelerómetro piezoeléctrico, pero el principio es aplicable a los tres tipos. En este modelo, la masa sísmica m está conectada a la base del sensor a través de un resorte de rigidez k y un amortiguador con coeficiente de amortiguamiento c . Este sistema responde a una vibración externa, y la Ecuación diferencial que describe el movimiento relativo de la masa respecto a la base es (Wu et al., 2023):

$$m \frac{d^2x}{dt^2} + c \left(\frac{dx}{dt} - \frac{dy}{dt} \right) + k(x - y) = 0 \quad (2.13)$$

Resolver la Ecuación diferencial (2.13) permite analizar la respuesta de los acelerómetros a distintas frecuencias de excitación, lo cual es crucial para determinar su rango de operación y comportamiento bajo diversas condiciones dinámicas.

2.5.2. Principios de Funcionamiento de los Acelerómetros MEMS

A partir del modelo mecánico de los acelerómetros MEMS, es posible analizar su respuesta ante distintas condiciones dinámicas y determinar su rango de operación. Si bien la estructura mecánica de estos sensores es similar, el

principio de conversión del movimiento en una señal eléctrica varía según el tipo de acelerómetro. En particular, los acelerómetros MEMS pueden basarse en efectos piezoeléctricos o capacitivos.

Acelerómetros Piezoeléctricos

Los acelerómetros piezoeléctricos son sensores utilizados para medir la aceleración mediante el efecto piezoeléctrico, que consiste en la conversión de una fuerza mecánica —producto de una aceleración— en una carga eléctrica. Esta transducción ocurre entre magnitudes mecánicas (fuerza o presión) y eléctricas (carga o voltaje), gracias a materiales cristalinos con propiedades piezoeléctricas. Estos sensores son empleados en aplicaciones industriales y de monitoreo de vibraciones debido a su alta sensibilidad y amplio rango de frecuencias de operación.

Estos sensores son utilizados en el monitoreo de vibraciones de maquinaria industrial, análisis estructural, pruebas de impacto y diagnóstico de fallas en sistemas mecánicos. Su rendimiento depende de factores como la constante piezoeléctrica del material utilizado. Por ejemplo, Pernia-Márquez, 2004 en su investigación menciona al titanato de circonato de plomo - PZT y la configuración mecánica del sensor.

Su rendimiento depende de factores como la constante piezoeléctrica y la constante dieléctrica del material utilizado, ya que estas propiedades determinan la eficiencia en la conversión de energía mecánica en eléctrica. Por ejemplo, Pernia-Márquez, 2004 menciona el uso de titanato de circonato de plomo (PZT), un material ampliamente empleado en sensores piezoeléctricos debido a su alta constante dieléctrica ($\epsilon_r \approx 1200$) y su elevada constante piezoeléctrica. La alta constante dieléctrica del PZT permite una mejor acumulación de carga eléctrica en respuesta a un esfuerzo mecánico, lo que mejora la sensibilidad del sensor. Además, su configuración mecánica influye en la respuesta del dispositivo, optimizando su desempeño en aplicaciones de detección de vibraciones. En este proceso, la fuerza mecánica aplicada sobre el sensor se transmite al material piezoeléctrico, generando una carga eléctrica proporcional a la aceleración experimentada. La carga inducida Q puede expresarse como (Pernia-Márquez, 2004):

$$Q = kF = kma \quad (2.14)$$

donde F es la fuerza aplicada, m es la masa sísmica interna del sensor, a es la aceleración a medir y k es la constante piezoeléctrica del material utilizado.

De acuerdo con Wu et al., 2023, un acelerómetro piezoeléctrico puede modelarse mediante dos circuitos equivalentes: el modelo de carga y el modelo de voltaje. La Figura 2.6 muestra ambas representaciones.

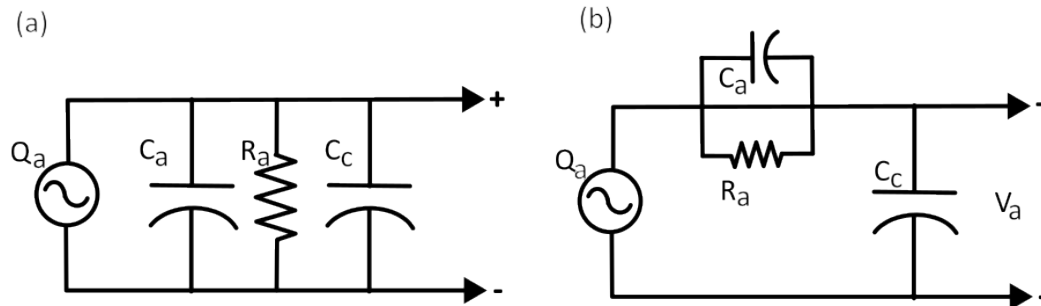


FIGURA 2.6: Circuito equivalente de un acelerómetro piezoeléctrico. (a) Modelo de carga. (b) Modelo de voltaje.

En el circuito (a) se muestra el modelo de carga donde, el acelerómetro piezoeléctrico se representa como una fuente de carga ideal Q_a , la cual está en paralelo con:

- C_a : Capacitancia del sensor piezoeléctrico.
- C_c : Capacitancia del cable de distribución.
- R_a : Resistencia de fuga del sistema.

La señal generada es una carga eléctrica proporcional a la aceleración aplicada, la cual puede ser convertida en voltaje utilizando un circuito de adaptación de impedancia. Este modelo es útil cuando se emplea un amplificador de carga para obtener una señal de salida estable.

En el circuito (b) se muestra el modelo de voltaje donde, la señal del acelerómetro se representa como una fuente de carga Q_a con una capacitancia en serie. Se incluyen los mismos elementos del modelo de carga, pero reorganizados para reflejar la conversión de carga en un voltaje de salida V_a . En este caso:

- C_a representa la capacitancia del sensor.
- R_a es la resistencia de fuga del sistema.
- C_c es la capacitancia del cable.

Este modelo es utilizado cuando se emplean amplificadores de voltaje de alta impedancia, permitiendo medir directamente la señal sin necesidad de convertir la carga.

Ambos modelos describen el comportamiento eléctrico del acelerómetro piezoeléctrico desde distintas perspectivas. El modelo de carga es preferido cuando se utiliza un amplificador de carga, mientras que el modelo de voltaje es más adecuado para mediciones con amplificadores de alta impedancia. La elección del modelo depende de la configuración del sistema de medición y del tipo de procesamiento de señal requerido.

A partir del modelo eléctrico de un acelerómetro piezoeléctrico, se pueden obtener expresiones que describen su sensibilidad y respuesta. La sensibilidad de carga se expresa como:

$$S_q = \frac{Q_a}{a_s} \quad (2.15)$$

mientras que la sensibilidad de voltaje se define como:

$$S_v = \frac{V_a}{a_s} \quad (2.16)$$

donde a_s es la aceleración de entrada, Q_a es la carga de salida y V_a es el voltaje de salida. A partir de la relación entre la carga y la capacitancia total del sistema, se obtiene el voltaje de salida:

$$V_a = \frac{Q_a}{C_a + C_c} \quad (2.17)$$

donde C_a es la capacitancia del material piezoeléctrico y C_c es la capacitancia de distribución del cable de conexión.

Los acelerómetros piezoeléctricos ofrecen alta sensibilidad y un amplio rango de frecuencias. Sin embargo, también presentan ciertas limitaciones inherentes a su principio de funcionamiento:

- No pueden medir aceleraciones estáticas o de muy baja frecuencia debido a la descarga gradual de la señal piezoeléctrica.
- Sensibilidad a variaciones de temperatura, lo que requiere compensación en aplicaciones de alta precisión.
- Requieren circuitos de acondicionamiento para convertir la carga en una señal de voltaje utilizable.

Acelerómetros Capacitivos

Los acelerómetros capacitivos son dispositivos de medición de aceleración que operan bajo el principio de variación de capacitancia debido al desplazamiento relativo de las placas de un condensador. Estos sensores son ampliamente utilizados en aplicaciones industriales, automotrices y aeroespaciales debido a su alta sensibilidad y estabilidad.

El funcionamiento de los acelerómetros capacitivos se basa en la relación matemática que define la capacitancia de un condensador de placas paralelas (Flores Castro, 2023; Cherrez Illescas, 2022):

$$C = \epsilon_0 \cdot \epsilon_r \frac{A}{d} \quad (2.18)$$

donde:

- C es la capacitancia del sistema,
- ϵ_0 es la permitividad del vacío (8.85×10^{-12} F/m),
- ϵ_r es la constante dieléctrica del material aislante,
- A es el área de superposición de las placas,
- d es la distancia entre las placas del condensador.

En un acelerómetro capacitivo, una de las placas está fija mientras que la otra está conectada a una masa inercial suspendida por un resorte flexible. Cuando el sensor es sometido a una aceleración, la masa inercial experimenta una fuerza dada por la Segunda Ley de Newton 2.6.

Esta fuerza genera un desplazamiento relativo Δx entre las placas, modificando la distancia d y, en consecuencia, la capacitancia C . La variación en capacitancia se traduce en una señal eléctrica proporcional a la aceleración aplicada, la cual es procesada para obtener un voltaje de salida. Este principio de funcionamiento se ilustra en la Figura 2.7.

Si bien los acelerómetros capacitivos presentan numerosas ventajas, también poseen ciertas limitaciones que deben considerarse para su correcta aplicación:

- **Interferencias electromagnéticas:** Debido a su principio de operación basado en la capacitancia, estos sensores pueden ser susceptibles a interferencias externas en entornos con alto ruido eléctrico.

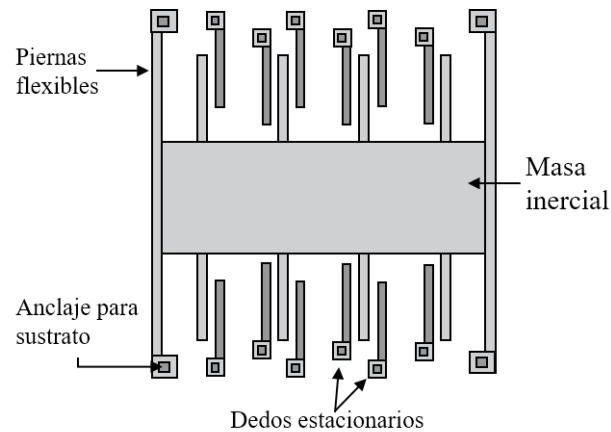


FIGURA 2.7: Modelo capacitivo de un acelerómetro.

- Dependencia de la calibración:** Para garantizar precisión en la medición, es necesario realizar una calibración periódica, especialmente en aplicaciones críticas.

2.6. MPU 9255

Entre los sensores inerciales utilizados en el análisis de vibraciones dentro de sistemas de monitoreo de salud estructural (SHM) se encuentra el MPU-9255. Este sensor de nueve grados de libertad (DOF, por sus siglas en inglés) integra un acelerómetro triaxial, un giroscopio triaxial y un magnetómetro triaxial, permitiendo obtener datos precisos sobre aceleraciones y cambios de orientación en estructuras sometidas a cargas dinámicas (Varanis et al., 2018). Su capacidad para detectar vibraciones en tiempo real lo hace útil en la evaluación de la integridad de puentes, edificios y maquinaria industrial. También es utilizado en aplicaciones de navegación y seguimiento de movimiento, ya que proporciona información precisa sobre la aceleración y la orientación del sistema monitoreado.

El acelerómetro triaxial del MPU-9255 permite la detección de aceleraciones en los ejes x , y y z , con un rango de medición configurable de $\pm 2g$, $\pm 4g$, $\pm 8g$ y $\pm 16g$. Además, cuenta con convertidores analógico-digitales (ADC) de 16 bits para mejorar la resolución y precisión de las mediciones. Su eficiencia energética de corriente es de $450 \mu A$ en operación normal y modos de bajo consumo que permiten reducir el consumo hasta $8.4 \mu A$ en frecuencias de muestreo reducidas.

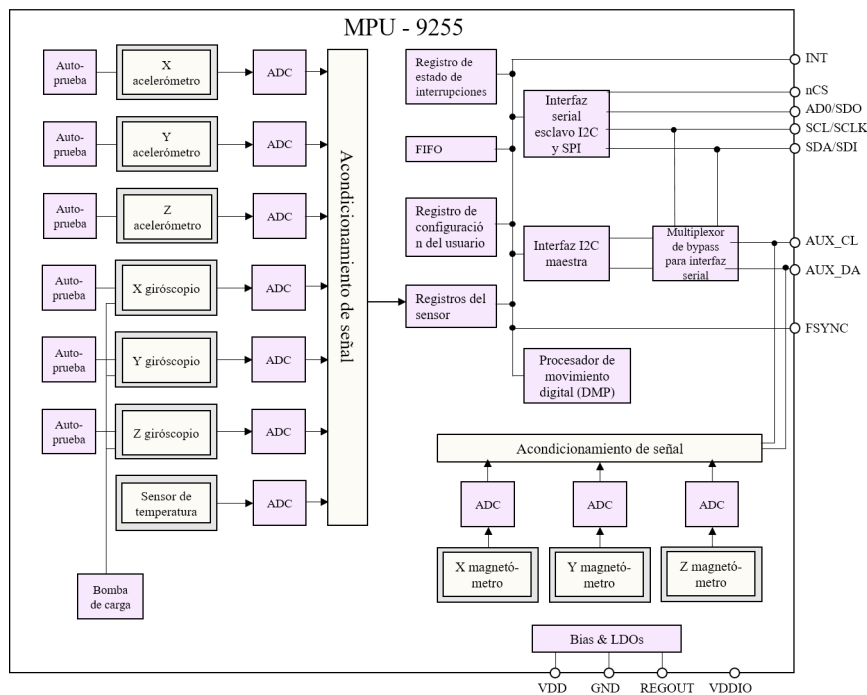


FIGURA 2.8: Diagrama de bloques de un MPU 9255.

En la Figura 2.8 se presenta un diagrama de bloques del sensor MPU-9255, en el cual se detalla su composición interna y la interconexión de sus componentes principales. Este diagrama permite visualizar de manera clara la estructura y funcionamiento interno del dispositivo.

El giroscopio integrado permite la medición de la velocidad angular, lo que, combinado con el magnetómetro triaxial, ofrece una estimación de la orientación del sensor (Nikolaev, Valles y Shcherbakov, 2023; Shcherbachev et al., 2024).

El MPU-9255 también incluye funciones avanzadas como interrupciones programables y un modo de “wake-on-motion”, que optimiza el consumo de energía al activar el procesamiento solo cuando se detecta movimiento.

Además, el sensor se comunica principalmente a través del protocolo I2C (Inter-Integrated Circuit, por su siglas en inglés), lo que lo hace compatible con microcontroladores y sistemas embebidos. La interfaz I2C permite la conexión de múltiples dispositivos en un bus compartido, lo que facilita su integración en sistemas más complejos sin necesidad de un gran número de pines.

Mediante la comunicación I2C, es posible configurar y acceder a los datos de los sensores del MPU-9255 en tiempo real. El sensor permite ajustar el rango de medición, los modos de operación y otros parámetros a través de

registros internos, lo que ofrece flexibilidad para diversas aplicaciones. La elección de I2C es adecuada para proyectos que no requieren una tasa de datos extremadamente alta, pero que buscan simplicidad y eficiencia en la conexión.

2.7. Acelerómetro del MyRIO-1900

El MyRIO-1900 es un dispositivo compacto y reconfigurable de entrada/salida (I/O) desarrollado por National Instruments, diseñado para aplicaciones de control y adquisición de datos en sistemas mecatrónicos y robóticos. Este dispositivo integra un procesador dual-core Xilinx Zynq-7010 a 667 MHz, 256 MB de memoria no volátil, y una variedad de puertos de conexión, incluyendo entradas y salidas analógicas (AI/AO), entradas/salidas digitales (DIO), audio, y puertos de expansión (MXP y MSP). Además, cuenta con conectividad USB, Bluetooth y Wi-Fi, lo que permite su integración con computadoras y sistemas externos.

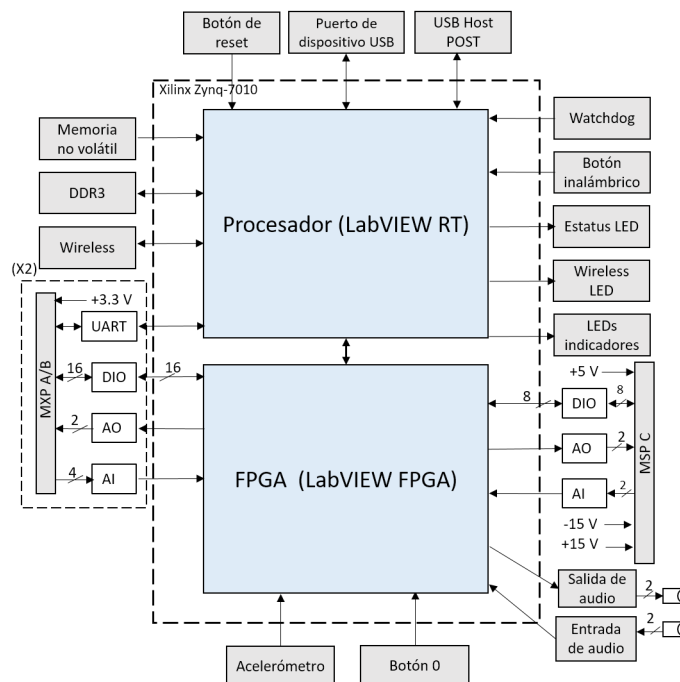


FIGURA 2.9: Diagrama de bloques del hardware NI MyRIO-1900.

Uno de los componentes clave del MyRIO-1900 es su acelerómetro integrado de tres grados de libertad (DOF, por sus siglas en inglés), que permite

medir la aceleración en los ejes x , y , y z . Este sensor es fundamental para aplicaciones que requieren monitoreo de movimiento, vibraciones, o control de orientación en tiempo real (Izquierdo et al., 2022).

El acelerómetro tiene las siguientes especificaciones:

- **Número de ejes:** 3 (x , y , z)
- **Rango de medición:** ± 8 g
- **Resolución:** 12 bits
- **Velocidad de muestreo:** 800 muestras por segundo (S/s)

El acelerómetro proporciona datos en formato digital que representan la aceleración medida en cada eje. Para interpretar estos datos, es necesario convertir los valores digitales en unidades físicas de aceleración (g). La conversión se realiza utilizando la siguiente ecuación:

$$A = D \times LSB \quad (2.19)$$

Donde:

- A es la aceleración en g.
- D es el valor digital obtenido del acelerómetro.
- LSB (Least Significant Bit, por sus siglas en inglés) es el peso del bit menos significativo, que se calcula como:

$$LSB = \frac{\text{Rango nominal}}{2^{\text{Resolución del ADC}}} \quad (2.20)$$

Para el acelerómetro del MyRIO-1900:

- Rango nominal = 16 g (de -8 g a +8 g)
- Resolución del ADC = 12 bits

Dado que el convertidor analógico-digital (ADC) tiene una resolución de 12 bits, el número total de niveles de cuantización es $2^{12} = 4096$. Sin embargo, los valores digitales están representados en un formato con signo, lo que significa que el rango efectivo del ADC va de -2048 a $+2047$.

Por lo tanto:

$$LSB = \frac{16 \text{ g}}{2^{12}} = \frac{16 \text{ g}}{4096} = 0.00390625 \text{ g} \quad (2.21)$$

Esto significa que cada incremento en el valor digital representa aproximadamente 0.00390625 g. Por ejemplo, un valor digital de 2047 correspondería a:

$$A = 2047 \times 0.00390625 \text{ g} \approx 7.996 \text{ g} \quad (2.22)$$

Este cálculo es necesario para interpretar de manera correcta los datos proporcionados por el acelerómetro. El dispositivo MyRIO entrega directamente las aceleraciones en unidades de g lo cual simplifica su implementación y uso por parte de los usuarios.

Gracias a su integración y especificaciones, el acelerómetro del MyRIO-1900 es ideal para diversas aplicaciones, tales como:

- **Detección de inclinación y nivelación:** Utilizando las componentes de aceleración estática para determinar la orientación del dispositivo respecto a la gravedad.
- **Monitoreo de vibraciones:** Analizando las aceleraciones dinámicas para detectar patrones de vibración en máquinas o estructuras.
- **Análisis de movimiento:** Registrando cambios en la aceleración para estudiar el comportamiento dinámico en sistemas robóticos o vehículos.

El acelerómetro integrado en el MyRIO es un sensor basado en tecnología MEMS (Micro-Electro-Mechanical Systems) que emplea principios capacitivos para medir la aceleración en tres ejes. Este sensor detecta cambios en la capacitancia debido al movimiento de microestructuras internas cuando se somete a vibraciones o variaciones de velocidad (Izquierdo et al., 2022). En el contexto del presente proyecto, el MyRIO se utiliza para la adquisición y procesamiento de señales de aceleración en tiempo real mediante LabVIEW, permitiendo la caracterización y clasificación de reductores de velocidad en función de su impacto en la dinámica vehicular. La integración de algoritmos de análisis de vibraciones facilita la identificación de patrones característicos en la respuesta del vehículo, contribuyendo a la detección y clasificación precisa de los reductores de velocidad.

2.8. Redes Neuronales Convolucionales (CNN)

Las Redes Neuronales Convolucionales (CNN, por sus siglas en inglés) han demostrado ser herramientas poderosas para el procesamiento de datos estructurados espacial o temporalmente. Si bien son ampliamente utilizadas en la visión por computadora, donde las CNN 2D (dos dimensiones) dominan el análisis de imágenes, las CNN en una dimensión (1D) han cobrado relevancia en el procesamiento de señales y series temporales. Su arquitectura permite extraer características relevantes de datos secuenciales como señales biomédicas (Zarei, Beheshti y Asl, 2022), datos financieros (Mehtab y Sen, 2022) y sensores industriales (Junior et al., 2022).

2.8.1. Fundamentos Matemáticos de una CNN-1D

Las CNN en 1D operan con convoluciones en una sola dimensión, lo que implica la aplicación de un filtro sobre una serie temporal. El proceso matemático estándar de convolución en redes neuronales está representado por la Ecuación 2.23:

$$y_m(n) = \sum_{k=0}^{N-1} x(m, n+k) \cdot w(k) \quad (2.23)$$

Donde:

- m : Número de fila.
- $x[n]$: Señal de entrada.
- w : Kernel (filtro).
- N : Longitud del kernel.
- $y[n]$: Resultado de la convolución.
- n : Número de columna.
- k : Índice interno de la sumatoria.

El autor Bengio, Goodfellow, Courville et al., 2017, en su libro, plantea un esquema similar para representar la convolución de dos dimensiones. Por lo tanto, en la Figura 2.10 se muestra de manera esquemática la Ecuación 2.23, la cual representa el proceso de la convolución unidimensional.

Esta operación permite la detección de patrones locales dentro de la serie temporal, proporcionando robustez ante desplazamientos temporales.

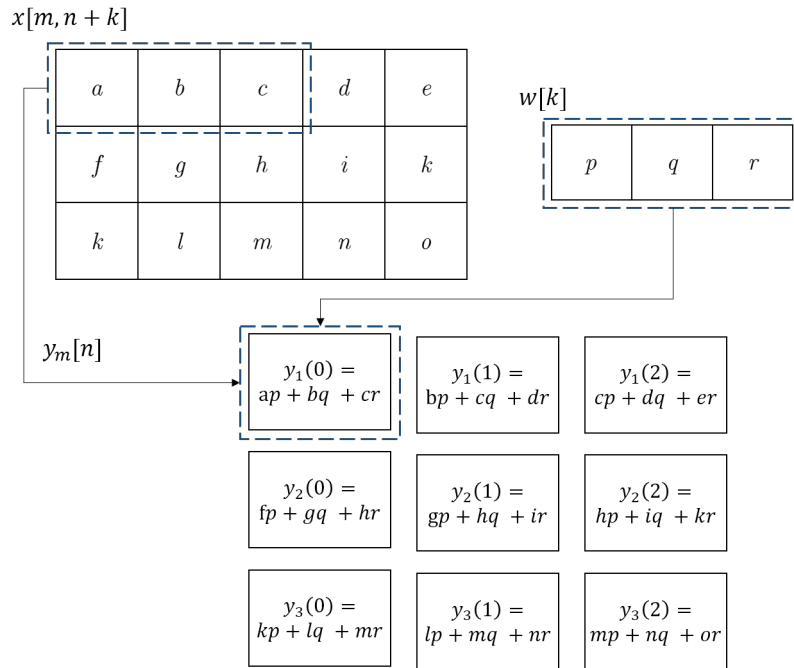


FIGURA 2.10: Diagrama del proceso de convolución unidimensional en una red neuronal.

2.8.2. Arquitectura de una CNN en 1D

Las CNN en 1D están compuestas por varias capas que transforman progresivamente la representación de los datos de entrada:

Capa de Convolución

Es la capa fundamental donde se aplican los filtros para extraer características relevantes de la serie temporal. Los hiperparámetros principales son:

- **Tamaño del filtro:** Define cuántos puntos de datos se analizan en cada paso.
- **Stride (desplazamiento):** Indica cuántos pasos se mueve el filtro en cada operación.
- **Padding:** Puede ser "same" para conservar la longitud de la entrada o "valid" para reducirla.

$$h_y = \frac{h_x - h_w + 2(p)}{\text{stride}} + 1 \quad (2.24)$$

$$w_y = \frac{w_x - w_w + 2(p)}{\text{stride}} + 1 \quad (2.25)$$

Donde:

- h_y : Altura de la salida.
- h_x : Altura de la entrada.
- h_w : Altura del filtro (kernel).
- w_y : Ancho de la salida.
- w_x : Ancho de la entrada.
- w_w : Ancho del filtro (kernel).
- p : Padding (relleno).
- stride: Número de desplazamientos del kernel.

Los hiperparámetros se configuran según lo que se espera obtener en la salida. De este modo, es posible ajustar el tamaño de la salida modificando estos hiperparámetros. En la Ecuación 2.24 se muestra cómo se calcula la altura de la salida de la convolución, mientras que la Ecuación 2.25 muestra cómo se calcula el ancho de la salida, ambas basándose en la señal de entrada y en parámetros como el padding, el kernel y el stride.

2.8.3. Funciones de Activación

Las funciones de activación introducen no linealidad, permitiendo que la red aprenda relaciones complejas (Sanchez-Castro et al., 2020). Entre las más utilizadas están:

- **ReLU (Rectified Linear Unit, por su significado en inglés):**

$$f(x) = \text{máx}(0, x) \quad (2.26)$$

- **Sigmoide:**

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.27)$$

- **Tangente Hiperbólica (tanh):**

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.28)$$

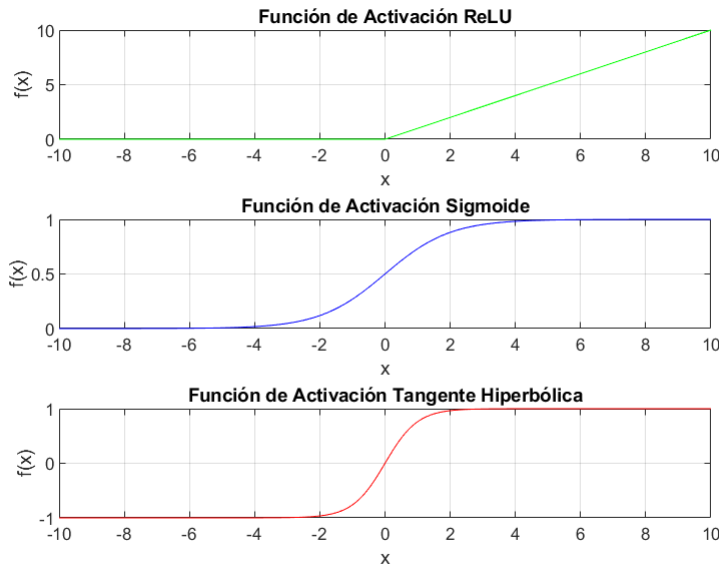


FIGURA 2.11: Comportamiento de las funciones de activación ReLU, sigmoide y tanh.

En la Figura 2.11 se muestra el comportamiento de las funciones de activación ReLU, sigmoide y tanh. En la función ReLU, los valores negativos se convierten en 0, mientras que los valores positivos permanecen iguales. La función sigmoide mapea los valores de x al rango entre 0 y 1, lo que la hace útil para problemas de clasificación binaria. Por su parte, la función tanh mapea los valores de x al rango entre -1 y 1.

2.8.4. Capa de Pooling

La capa de pooling reduce la dimensionalidad de la representación y mejora la capacidad de generalización (Sanchez-Castro et al., 2020). En CNN 1D, las técnicas más comunes son el max pooling y el average pooling.

- **Max pooling:** Selecciona el valor máximo dentro de una ventana deslizante, lo que resalta las características más relevantes y reduce la sensibilidad a pequeñas variaciones.

- **Average pooling:** Calcula el valor promedio dentro de la ventana, preservando más información de la señal pero atenuando las características distintivas.

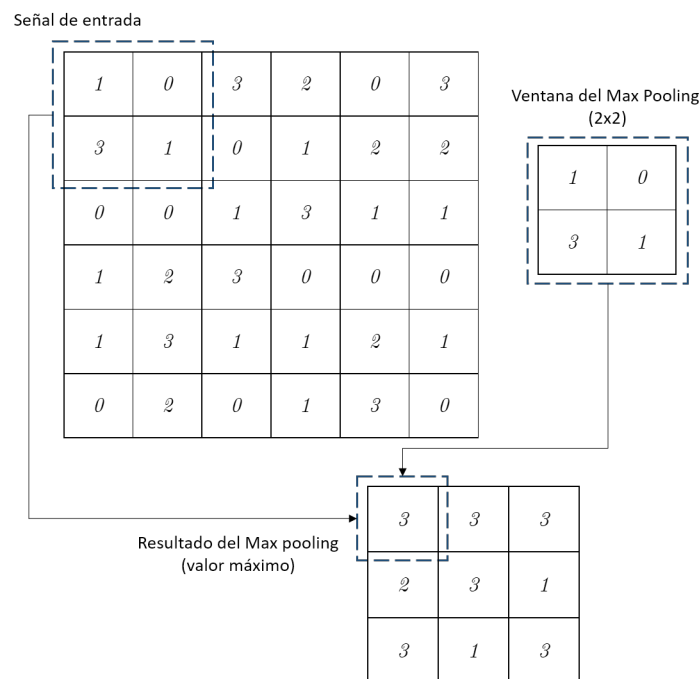


FIGURA 2.12: Operación de Max Pooling de dimensión 2x2 en una matriz de entrada. Cada región de la ventana deslizante selecciona el valor máximo para construir la matriz de salida.

Aunque ambas técnicas son ampliamente utilizadas, el max pooling es preferido en la mayoría de las aplicaciones, ya que ayuda a mantener las características más representativas de la señal al eliminar valores irrelevantes o ruido. Por otro lado, el average pooling tiende a suavizar los datos, lo que puede ser útil en ciertos escenarios donde se busca reducir el impacto de picos.

En la Figura 2.12 se ilustra visualmente el proceso de max pooling en una matriz de entrada. Se observa cómo, dentro de cada ventana deslizante, se selecciona el valor más alto para formar la salida reducida.

2.8.5. Capa Fully Connected

Al final de la arquitectura convolucional, las características extraídas se conectan a capas densamente conectadas (fully connected), terminando en una función de salida adecuada para la tarea (regresión o clasificación).

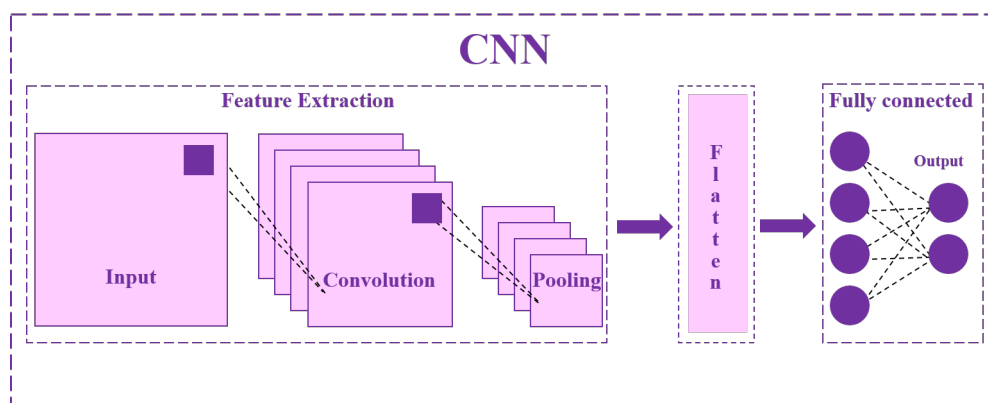


FIGURA 2.13: Esquema de una red neuronal convolucional (CNN) en el que se lleva a cabo la extracción de características.

En la Figura 2.13 se presenta un esquema de una red neuronal convolucional (CNN), donde el proceso de extracción de características comienza con una matriz de entrada a la que se le aplican operaciones de convolución utilizando diferentes filtros para detectar patrones en los datos. Posteriormente, se emplea una capa de pooling para reducir las dimensiones de la representación, preservando la información más relevante. Luego, las características extraídas y comprimidas se les realiza la operación flatten convirtiéndolas en un vector unidimensional. Finalmente, este vector es enviado a una capa fully connected, que procesa la información y genera la salida final, en este caso, correspondiente a una clasificación.

2.9. Preprocesamiento de Señales Inerciales para la Detección de Reductores de Velocidad

La detección de reductores de velocidad en la carretera puede llevarse a cabo mediante diversos métodos, de los cuales los más utilizados son el escaneo láser 3D, las técnicas basadas en visión por computadora y los métodos que emplean sistemas IMU (Sattar, Li y Chapman, 2018). El escaneo láser 3D se utiliza para crear modelos precisos de la superficie de la carretera, mientras que las técnicas de visión por computadora, impulsadas por modelos avanzados de aprendizaje profundo, interpretan imágenes o secuencias de video para identificar irregularidades en la vía. Por otro lado, los métodos basados en una IMU se centran en la medición de vibraciones, utilizando

sensores que detectan y analizan las señales producidas por las variaciones en la superficie de la carretera.

En esta tesis, se opta por el uso de sensores inerciales, debido a sus ventajas en términos de costo, portabilidad y capacidad de realizar mediciones en tiempo real sin la necesidad de infraestructura adicional o acceso visual directo a la carretera. Los sensores inerciales son capaces de proporcionar datos en condiciones de tráfico dinámico, lo que hace que este enfoque sea especialmente útil para el monitoreo continuo de reductores de velocidad, ofreciendo un análisis preciso y eficiente a lo largo de rutas variadas.

Salman y Mian, 2023 en su trabajo realiza la clasificación de reductores de velocidad a partir de las lecturas de sistemas IMU de un dispositivo móvil, el cual fue colocado en el tablero del vehículo. Para recopilar los datos, tomaron las lecturas del acelerómetro y giroscopio del celular en los ejes x , y y z mientras el vehículo cruzaba un reductor de velocidad. Para etiquetar estos datos, los autores realizaron un proceso manual en el que presionaban un botón al inicio del paso por el reductor y lo soltaban al finalizar, marcando así el evento en los datos recopilados.

El autor menciona que enfrentó problemas con el ruido presente en las lecturas de los sensores, por lo que aplicó un promedio móvil simple (PMS) a cada ventana de datos capturada. La fórmula utilizada se muestra en la Ecuación 2.29:

$$\text{PMS} = \frac{\sum_{i=1}^n X_i}{n} \quad (2.29)$$

donde X_i representa los valores de la serie de tiempo y n es el tamaño de la ventana de promedio.

En su investigación, Salman y Mian, 2023 concluye que el tamaño de ventana que proporcionó los mejores resultados fue de 250 muestras, lo que corresponde a una duración de 5 segundos. Finalmente, el autor normalizó los datos en un rango de 0 a 1 para asegurar que todas las características estuvieran en la misma escala. Los modelos entrenados lograron clasificar eventos de “Reductores de velocidad” y “Carretera Normal”.

Por otro lado, el autor Celaya-Padilla et al., 2018 desarrolló un sistema para la detección de reductores de velocidad utilizando un acelerómetro de tres ejes y un GPS montados en el tablero del vehículo. Para la implementación, empleó un Raspberry Pi 3 y un sensor MPU6050 ubicado en la parte externa del vehículo, específicamente en la defensa. Los datos se recopilaron en ventanas de 2 segundos.

A cada ventana se le extrajeron siete características: media (2.30), varianza (2.31), asimetría (2.32), curtosis (2.33), desviación estándar (2.34), valor máximo (2.35) y rango dinámico (2.36).

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n X_i \quad (2.30)$$

$$\sigma^2 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n} \quad (2.31)$$

$$\gamma_1 = \frac{\frac{1}{n} \sum_{i=1}^n (X_i - \bar{x})^3}{\left(\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{x})^2 \right)^{3/2}} \quad (2.32)$$

$$K = \frac{\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^4}{\left(\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2 \right)^2} - 3 \quad (2.33)$$

$$\sigma = \sqrt{\sigma^2} \quad (2.34)$$

$$X(n) = \text{máx}\{X_1, \dots, X_n\} \quad (2.35)$$

$$DR = X(n) - \text{mín}\{X_1, \dots, X_n\} \quad (2.36)$$

Donde:

- \bar{x} es la media de los datos.
- σ^2 es la varianza.
- γ_1 es la asimetría.
- K es la curtosis.
- σ es la desviación estándar.
- $X(n)$ es el valor máximo de la muestra.
- DR es el rango dinámico.
- X_i representa cada dato de la muestra.
- n es el número total de muestras.

Tras el preprocesamiento de las ventanas, estas fueron utilizadas como entrada en un algoritmo de aprendizaje automático para la clasificación de reductores de velocidad. El autor obtuvo buenos resultados con una ventana de 2 segundos y una velocidad de conducción de 20 km/h.

Ambos autores utilizaron las lecturas de un acelerómetro y organizaron los datos en ventanas de distintas dimensiones. Aunque aplicaron diferentes métodos de preprocesamiento a la señal antes de ingresarla en un algoritmo de IA, ambos obtuvieron buenos resultados. Sin embargo, ninguno de los sistemas opera en tiempo real, lo que limita su aplicabilidad en escenarios que requieren una detección inmediata.

A diferencia de trabajos previos, el sistema desarrollado en esta tesis organiza las lecturas del acelerómetro en ventanas de datos desplazadas, lo que permite un análisis continuo y eficiente de las señales. A pesar de las dificultades inherentes al uso de sensores inerciales, se implementó un esquema de preprocesamiento eficaz. Este proceso incluye la normalización de las ventanas de datos a una escala de grises, lo que mejora el contraste de las señales y facilita su interpretación por la red neuronal convolucional unidimensional. Gracias a este enfoque, el sistema logra clasificar reductores de velocidad en tiempo real con una exactitud superior al 90 %. Esta capacidad de detección inmediata no solo mejora la exactitud del sistema, sino que también aumenta su aplicabilidad en entornos dinámicos, donde una respuesta rápida y confiable es esencial, eliminando la necesidad de un extenso preprocesamiento y procesamiento, demostrando la viabilidad del método propuesto para futuras aplicaciones en este contexto.

Capítulo 3

Procedimiento de Investigación

El monitoreo de carreteras mediante sensores inerciales ha ganado relevancia en la detección y clasificación de irregularidades viales, proporcionando información clave para la seguridad del tránsito. La capacidad de medir aceleraciones en distintos ejes permite caracterizar eventos específicos, como los reductores de velocidad, en tiempo real. Este enfoque es especialmente útil en aplicaciones que requieren una respuesta rápida y precisa para la toma de decisiones automatizada.

Existen diversos métodos para clasificar irregularidades en las vialidades, entre ellos, los basados en imágenes y los que utilizan sensores inerciales. Los sistemas de visión artificial pueden operar en tiempo real y han demostrado ser exactos y precisos; sin embargo, su exactitud depende de factores externos, como la calidad del reductor de velocidad. Si este está despintado, deteriorado o carece de señalización adecuada, su detección se ve afectada significativamente. En contraste, los métodos basados en sensores inerciales dependen únicamente de las aceleraciones registradas, lo que los hace independientes de condiciones ambientales como la iluminación o el estado del reductor de velocidad.

Si bien existen diversas investigaciones que utilizan sensores inerciales para el análisis estructural de reductores de velocidad, el procesamiento de datos varía entre autores. Aunque la mayoría organiza las señales de serie de tiempo en ventanas, el método específico de segmentación, preprocesamiento y procesamiento de los datos influye en los resultados obtenidos. Dado que la estructura y tratamiento de los datos afectan directamente el desempeño del algoritmo, cada enfoque puede generar diferentes niveles de precisión y exactitud. Además, a pesar de los estudios previos sobre el uso de las IMU para detectar reductores de velocidad, no se ha identificado un sistema que opere en tiempo real, al menos dentro de la literatura revisada en esta investigación. La duración de cada ventana de datos utilizada en la clasificación varía entre 2 y 5 segundos, dependiendo del enfoque de cada autor. En el caso de la detección de reductores de velocidad y otras irregularidades en las

vialidades, una detección con un desfase de 2 segundos o más resulta inadecuada para su aplicación práctica. Esto se debe a que, si se desea mapear e indicar con precisión la ubicación de un reductor de velocidad utilizando GPS, un desfase temporal puede llevar a marcar la ubicación en un punto erróneo, lo que compromete la utilidad del sistema. Por ello, se definió que, para que un sistema pueda considerarse en tiempo real en este contexto, debe realizar una clasificación en menos de 500 milisegundos. Este umbral garantiza una respuesta lo suficientemente rápida para aplicaciones dinámicas, como la navegación en tiempo real o el mapeo preciso de irregularidades en las carreteras.

En este contexto, el presente capítulo detalla el procedimiento de investigación seguido en esta tesis de maestría. Se inicia con la descripción del método de adquisición de datos del acelerómetro integrado en el MyRIO, basado en tecnología MEMS capacitiva. Posteriormente, se explica el proceso de ventaneo de datos, en el cual las señales de aceleración se segmentan en ventanas temporales para su organización, preprocesamiento y optimización como entrada a una Red Neuronal Convolutiva de una dimensión (CNN-1D). Finalmente, se aborda la implementación del modelo de clasificación en tiempo real para la identificación de reductores de velocidad, con el objetivo de superar una exactitud del 90%. A lo largo del capítulo, se presentará detalladamente el procedimiento experimental seguido para el desarrollo y validación del sistema propuesto.

3.1. Selección de Hardware y Software para la Adquisición de Datos

En esta subsección se describe el proceso de selección del hardware y software utilizados para la adquisición de datos en el proyecto. Dado que el objetivo principal es capturar y procesar señales de aceleración de manera precisa y confiable, se evaluaron diferentes opciones de sensores y plataformas de adquisición. La selección se basó en criterios como la precisión, la estabilidad, la facilidad de integración y la capacidad de operar en tiempo real.

El software utilizado para la programación de este proyecto fue LabVIEW, empleado para la adquisición de datos, y Python, utilizado para implementar una Red Neuronal Convolutiva de una dimensión destinada a la detección de reductores de velocidad. Ambos programas fueron ideales, ya que

LabVIEW se destaca en la adquisición de datos y Python en el desarrollo de inteligencia artificial. Además, su compatibilidad permite una comunicación eficiente y fluida durante la ejecución en tiempo real.

El hardware utilizado en este proyecto fue seleccionado cuidadosamente para garantizar la adquisición, procesamiento y clasificación de datos de manera eficiente y confiable. A continuación, se describe cada uno de los componentes principales:

- **MyRIO-1900:** Este dispositivo fue utilizado como interfaz de comunicación entre la computadora y el sensor inercial. Su capacidad de procesamiento en tiempo real, sumada a su versatilidad para la adquisición de señales, su diseño compacto y robusto, lo hacen ideal para entornos dinámicos como el interior de un vehículo en movimiento. Además, el MyRIO-1900 es compatible de manera nativa con LabVIEW, lo que facilita su programación y configuración. Esta integración, junto con el hecho de que tanto el MyRIO-1900 como LabVIEW son productos de National Instruments, asegura una comunicación eficiente y un manejo óptimo de los datos adquiridos.
- **Laptop DELL Latitude 7480:** Se utilizó esta computadora portátil para el procesamiento de los datos adquiridos y la ejecución de los algoritmos de clasificación.
- **Honda Civic Coupe 2012:** Este vehículo fue empleado como plataforma móvil para la adquisición de datos mientras se encontraba en marcha. Su uso permitió simular condiciones reales de conducción, lo que fue fundamental para capturar señales de aceleración representativas de las irregularidades viales, como los reductores de velocidad.
- **Inversor de Corriente Portátil 300W, 12V a 110V:** Dado que el MyRIO-1900 requiere una fuente de alimentación estable, se utilizó este inversor de corriente para convertir la energía de 12V del vehículo a 110V, permitiendo así energizar el dispositivo de manera continua durante las pruebas en movimiento.

3.1.1. Selección y Evaluación del Acelerómetro

En este proyecto, uno de los componentes de hardware más críticos a seleccionar fue el acelerómetro para la adquisición de las aceleraciones. Por

facilidades de comunicación y disponibilidad, se consideraron dos opciones: el acelerómetro integrado en el sensor MPU-9255 y el acelerómetro del dispositivo MyRIO, ambos basados en tecnología MEMS capacitiva.

Para determinar cuál de estos sensores era más adecuado para el proyecto, se realizaron pruebas comparativas que evaluaron su desempeño en términos de estabilidad y precisión. Estas pruebas incluyeron análisis estadísticos de las señales capturadas, como el cálculo de la media, la desviación estándar y la varianza, con el objetivo de comparar su rendimiento y seleccionar el sensor que mejor se adaptara a los requerimientos del sistema.

El análisis se realizó en modo estático, ya que de esta manera se puede determinar cuánto ruido se detecta en el sensor si se mantiene estable. Para poder realizar un análisis estadístico entre los dos sensores propuestos, primero es necesario desarrollar un programa que adquiera las aceleraciones de ambos sensores y las almacene en un archivo Excel. Esto permitirá realizar una comparación adecuada entre los dos. Dado que cada sensor tiene un protocolo de comunicación distinto, se utilizaron diferentes configuraciones para poder leer las aceleraciones.

Protocolo de Comunicación para la MPU-9255

La MPU 9255, desarrollado por InvenSense, es un sensor inercial de nueve grados de libertad que integra un acelerómetro, un giroscopio y un magnetómetro. Este sensor puede leer señales y configurar parámetros mediante comunicación I2C, lo que facilita su integración con diversos microcontroladores, como Arduino y MyRIO.

A continuación se muestra el algoritmo 1 para estimar la aceleración a partir de los datos crudos del sensor MPU-9255, primero se leen los registros correspondientes a los valores más y menos significativos de cada eje (x , y , z). Luego, los bytes se concatenan para formar valores de 16 bits, considerando el complemento a dos para manejar números negativos. Posteriormente, los valores se normalizan dividiéndolos entre 16384, asumiendo un rango de $\pm 2g$. Finalmente, se convierten a unidades de metros sobre segundo cuadrado (m/s^2) multiplicándolos por la constante gravitacional $9.80665m/s^2$.

Algoritmo 1 Estimación de aceleración a partir de datos crudos de una MPU-9255.

Inicio

Leer datos crudos del acelerómetro

$$x_{MSB} = \text{read}(x_0)$$

$$x_{LSB} = \text{read}(x_0)$$

$$y_{MSB} = \text{read}(y_0)$$

$$y_{LSB} = \text{read}(y_0)$$

$$z_{MSB} = \text{read}(z_0)$$

$$z_{LSB} = \text{read}(z_0)$$

Unir los bytes para formar valores de 16 bits

$$x_i = x_{MSB} \times 256 + x_{LSB}$$

$$y_i = y_{MSB} \times 256 + y_{LSB}$$

$$z_i = z_{MSB} \times 256 + z_{LSB}$$

Convertir a unidades de "g" (suponiendo un rango de $\pm 2g$)

$$x_g = x_i / 16384$$

$$y_g = y_i / 16384$$

$$z_g = z_i / 16384$$

Convertir a unidades de m/s^2

$$x_{m/s^2} = x_g \times 9.80665$$

$$y_{m/s^2} = y_g \times 9.80665$$

$$z_{m/s^2} = z_g \times 9.80665$$

Fin

Protocolo de Comunicación para el Acelerómetro del MyRIO

Cada sistema IMU emplea su propio protocolo de comunicación y procesamiento de señales para obtener aceleraciones en m/s^2 . El NI MyRIO-1900 integra un acelerómetro triaxial con una tasa de muestreo de 800 Hz, cuyas mediciones se expresan inicialmente en g y luego se convierten a m/s^2 . A diferencia del MPU-9255, la comunicación con este acelerómetro es directa, ya que LabVIEW proporciona funciones específicas que acceden a sus lecturas sin necesidad de configuración adicional, mostrándolas directamente en g . Esta integración simplifica la adquisición de datos, como se muestra en el algoritmo 2.

Algoritmo 2 Lectura del acelerómetro del NI MyRIO-1900.

Inicio

Leer valores de aceleración en g : x_g, y_g, z_g

Convertir a unidades de m/s^2 usando la Ecuación:

$$x_{ms} = x_g \times 9.80665$$

$$y_{ms} = y_g \times 9.80665$$

$$z_{ms} = z_g \times 9.80665$$

Fin

Programa en LabVIEW para la Adquisición y Recopilación de Datos de Ambos Sensores Inerciales

Una vez identificado el protocolo de comunicación de cada sensor, se desarrolló un programa en LabVIEW para la adquisición de las aceleraciones. Estas lecturas se almacenan en un archivo para su posterior análisis.

El programa permite la comunicación con ambos sensores inerciales a través del MyRIO, el cual se conecta a la computadora mediante Wi-Fi. Las aceleraciones se visualizan en tiempo real mediante gráficos e indicadores, y simultáneamente se guardan en un archivo para su procesamiento posterior. La Figura 3.1 muestra el diagrama de flujo del programa, el cual configura la comunicación I2C con el sensor MPU 9255 y establece la comunicación simultánea con el acelerómetro integrado en el MyRIO. Los datos, adquiridos en formato crudo, se procesan para convertirlos a m/s^2 , como se describe en los algoritmos 1 y 2, y se visualizan en tiempo real en LabVIEW. Al finalizar la captura, los datos se almacenan en un archivo CSV, se detiene la lectura de los sensores y se cierra el programa.

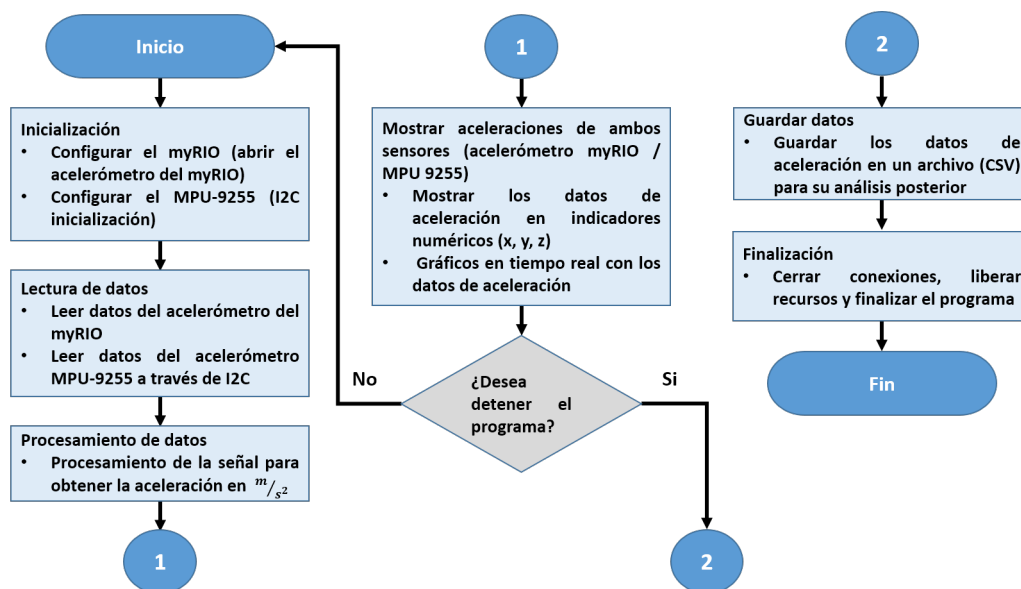


FIGURA 3.1: Diagrama de flujo del programa en LabVIEW para la adquisición y recopilación de datos de ambos sensores inerciales.

3.1.2. Evaluación del Acelerómetro del MyRIO vs el MPU-9255

Una vez establecida la comunicación con ambos sensores, se realizaron pruebas manteniendo los dos acelerómetros en reposo durante varios minutos. Durante este período, se registraron los datos de aceleración en m/s^2 de ambos sensores para llevar a cabo un análisis estadístico y determinar cuál de los dos ofrece una mejor estabilidad en nuestro sistema. Dado que el eje z es el más afectado por la gravedad, se graficó la distribución normal de los datos en dicho eje para cada sensor. La Figura 3.2 muestra que la distribución de los datos en el eje z del sensor MPU 9255 es más amplia en comparación con la del acelerómetro del MyRIO, lo que sugiere que la aceleración en el eje z es más estable en el MyRIO que en el MPU 9255.

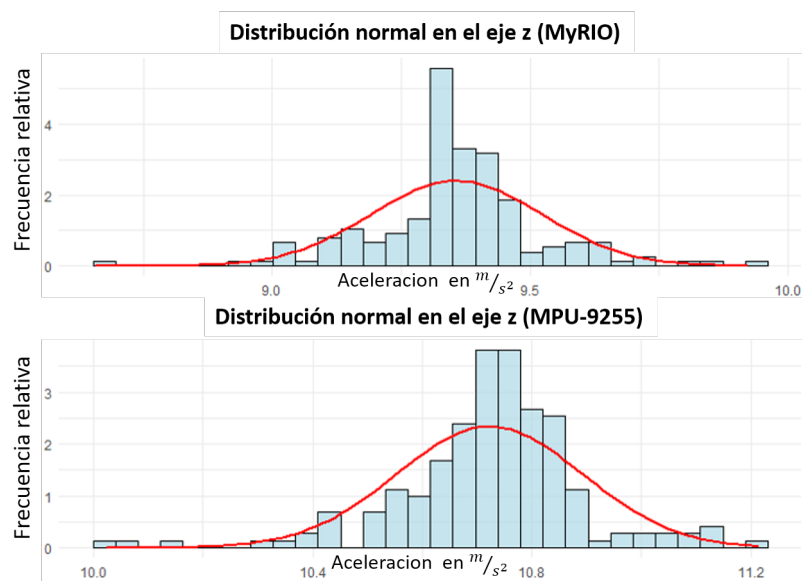


FIGURA 3.2: Distribución normal de los datos del eje z: comparación entre el acelerómetro MyRIO y el MPU 9255.

A partir de las aceleraciones medidas en los ejes x , y y z de cada sensor, se calcularon la media, la desviación estándar y el error cuadrático medio (RMS, por sus siglas en inglés). Los resultados obtenidos para cada eje se presentan en la Tabla 3.1. El análisis indica que el acelerómetro del MyRIO presenta una mayor estabilidad en estado estático en comparación con el MPU 9255, lo que se refleja en una media más cercana al valor esperado, así como en una menor desviación estándar y un menor error cuadrático medio.

Con base en el análisis estadístico de las aceleraciones obtenidas de ambos sensores en estado estático, decidimos utilizar el acelerómetro del MyRIO para nuestro sistema.

TABLA 3.1: Comparación de la media, desviación estándar y error cuadrático medio para las aceleraciones medidas en los ejes x , y y z de los sensores MyRIO y MPU-9255.

Medida estadística	MyRIO			MPU-9255		
	x	y	z	x	y	z
Eje						
Media	0.3531	0.4125	9.2571	0.779	0.0405	10.7229
Desviación estándar	0.5857	0.5748	0.166	0.6535	0.5659	0.1706
RMSE	0.0019	0.0019	0.4678	0.003	0.0028	0.6199

3.2. Organización y Estructuración de los Datos

La presente tesis busca determinar la entrada adecuada para una red neuronal convolucional unidimensional, con el fin de identificar patrones en series de tiempo en tiempo real. A partir de estos patrones, es posible detectar y clasificar distintos eventos dentro de la serie temporal.

En este caso, la aplicación se centra en la detección de reductores de velocidad. Identificarlos mediante series de tiempo es un desafío, ya que un reductor de velocidad no ocurre en un solo instante, sino que abarca una secuencia de datos a lo largo del tiempo.

El principal reto en otros sistemas de detección de reductores de velocidad es su implementación en tiempo real. El método más efectivo para identificar el patrón correspondiente a esta clase es el uso de ventanas de datos, técnica aplicada por diversos autores. En estos enfoques, las aceleraciones capturadas durante el evento se etiquetan como "Reductor", mientras que todas las demás se clasifican como "Nada".

Generalmente, la segmentación se realiza como se muestra en la Figura 3.3. En ella, W_2 representa la ventana que contiene las aceleraciones correspondientes a la clase "Reductor", mientras que las aceleraciones previas y posteriores se etiquetan como "Nada". Aunque este método permite identificar correctamente el evento, impide su detección en tiempo real. Si el suceso dura 2 segundos, la clasificación tendrá un desfase del mismo tiempo.

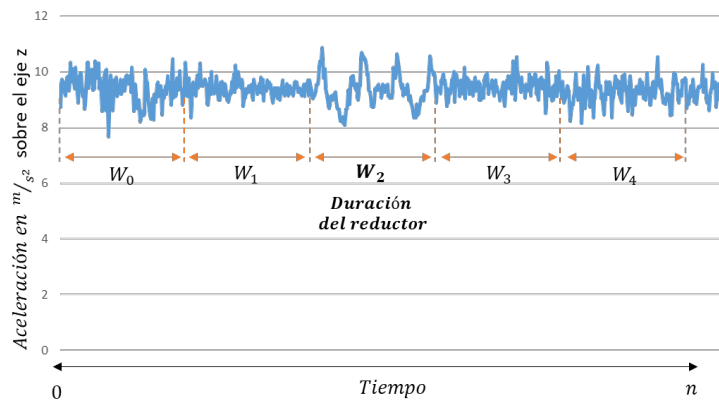


FIGURA 3.3: Segmentación de la señal de aceleración en el eje z , donde W_2 representa la ventana correspondiente al *Reductor de velocidad*. Las demás ventanas contienen datos previos y posteriores, etiquetados como *Nada*.

Basado en técnicas conocidas, el método propuesto en este proyecto agrupa las aceleraciones en ventanas deslizantes. Este enfoque permite analizar

la información antes y durante el paso sobre un reductor de velocidad, facilitando su identificación desde el inicio del cruce, sin necesidad de esperar a atravesarlo por completo. La clasificación se realiza con baja latencia, permitiendo una respuesta en tiempo real para nuestra aplicación.

En la Figura 3.4 se observa la segmentación de ventanas, donde algunas se traslapan debido a que solo se desplazan ciertos intervalos de tiempo. Esto permite que cada ventana contenga información tanto del pasado como del presente, lo que facilita la detección progresiva del reductor de velocidad. Una CNN-1D, entrenada con estos datos, identifica patrones en la señal y determina si el vehículo está comenzando a cruzar un reductor de velocidad, si lo está atravesando o si ya lo ha pasado.

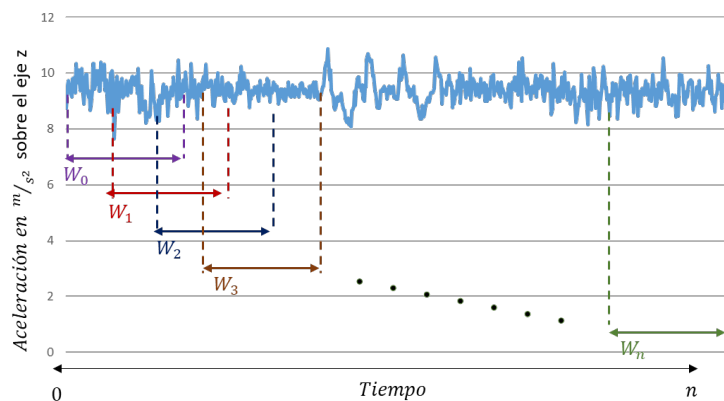


FIGURA 3.4: Segmentación con ventanas deslizantes, donde cada W_i captura información previa y actual para mejorar la detección del reductor de velocidad en tiempo real.

El tiempo de detección de los reductores de velocidad dependerá en gran medida del tamaño de las ventanas utilizadas. Asimismo, la exactitud de la CNN se verá afectada principalmente por este parámetro. Por lo tanto, es crucial encontrar un tamaño de ventana que contenga suficiente información para permitir que la CNN identifique patrones en la señal, pero que, al mismo tiempo, posibilite una detección rápida para su implementación en tiempo real. Además, el desplazamiento u *offset* utilizado en el recorrido entre ventanas también juega un papel clave, lo que a su vez afecta la continuidad y precisión del análisis. Por lo tanto, antes de definir el tamaño de la ventana, se llevó a cabo una experimentación en la que se transitó por distintos reductores de velocidad a diversas velocidades. El objetivo fue determinar cuántas muestras, adquiridas a una frecuencia de 800 Hz, corresponden en promedio a la duración del paso sobre un reductor de velocidad. Los resultados obtenidos fueron variables, ya que dependen tanto del tamaño del reductor como

TABLA 3.2: Duración promedio del cruce sobre un reductor de velocidad con una frecuencia de muestreo de 800 Hz.

Reductor de velocidad	Número de muestras
1	134
2	159
3	127
4	145
5	127
6	121
7	107
8	146
9	152
10	124
11	132
12	157
13	119
14	141
15	149
16	144
17	114
18	108
19	112
20	144

de la velocidad del vehículo. No obstante, las diferencias entre ellos no son significativas. Dichos resultados se presentan en la Tabla 3.2.

A partir de los datos obtenidos, se dedujo que se requieren aproximadamente 145 intervalos de tiempo (deltas de tiempo) para capturar completamente el paso por un reductor de velocidad. Esta cantidad varía según el tamaño del reductor y la velocidad del vehículo. Con esta información, se determinó que las dimensiones ideales de la ventana serían de 3×308 , ya que con 308 intervalos de tiempo es posible capturar todo el proceso del paso por el reductor, incluyendo los datos previos y posteriores al evento.

En esta ventana de 3×308 , cada fila representa la aceleración en uno de los tres ejes (x, y, z), y cada columna corresponde a un intervalo de tiempo.

En la Figura 3.5 se puede observar el deslizamiento de ventanas sobre el dataset de 756 deltas de tiempo, se generan 65 ventanas de tamaño 3×308 , avanzando 7 deltas de tiempo por iteración hasta completar el recorrido.

Aunque casi todas las ventanas contienen datos del paso sobre un reductor de velocidad, solo se etiquetan como “Reductor” aquellas en las que las aceleraciones corresponden al momento exacto del suceso. Si bien algunas

ventanas aún pueden incluir aceleraciones previas tras haber pasado el reductor de velocidad, no se etiquetan como tal, ya que corresponden al movimiento del vehículo después de haber cruzado completamente el reductor de velocidad y continuar en la carretera.

Es importante destacar que, para el entrenamiento del algoritmo de inteligencia artificial, se requieren conjuntos de 756 muestras, sobre las cuales se aplica el deslizamiento de ventanas, generando siempre las mismas 65 ventanas. No obstante, la cantidad de ventanas etiquetadas como “Reductor” varía, ya que cada reductor de velocidad puede contener un número diferente de deltas de tiempo, lo que afecta la distribución de las etiquetas en el conjunto de datos.

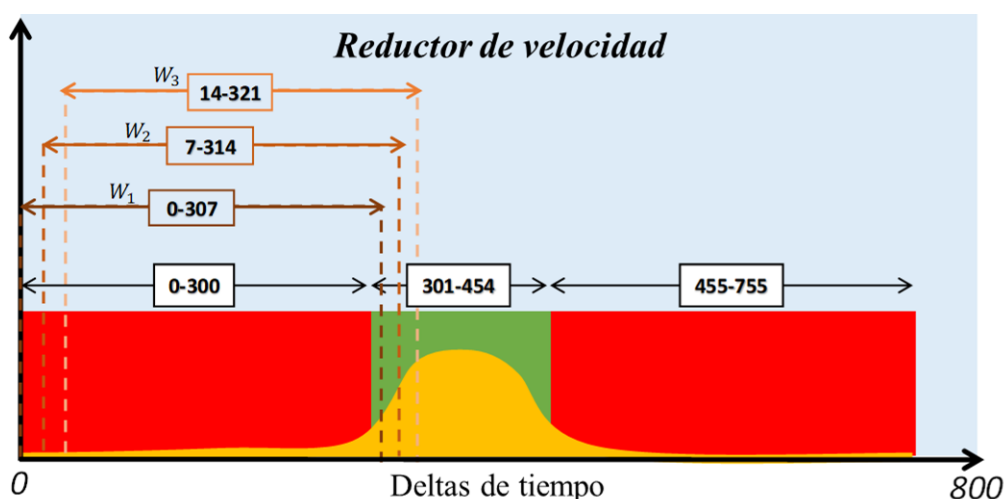


FIGURA 3.5: Segmentación con ventanas deslizantes, donde cada W_i captura información previa y actual para mejorar la detección del reductor de velocidad en tiempo real.

La Tabla 3.3 muestra un ejemplo del dataset generado para representar el desplazamiento completo sobre un reductor de velocidad con una duración de 154 deltas de tiempo.

Para cubrir todo el recorrido, se tomaron 756 deltas de tiempo y se aplicó un deslizamiento de ventanas, obteniendo 65 ventanas en total. En este ejemplo, las ventanas de la 1 a la 22 se etiquetan como “Reductor”, mientras que de la 23 a la 65 se etiquetan como “Nada”.

Este proceso de segmentación y etiquetado se repite para cada reductor de velocidad utilizado en el entrenamiento del algoritmo.

Dado que las redes neuronales convolucionales y el lenguaje de programación utilizado manejan los datos en forma de arreglos, adoptaremos esta

TABLA 3.3: Ejemplo de segmentación y etiquetado de ventanas para un reductor de velocidad.

Número de ventana	Deltas de tiempo	Etiqueta
1	0 - 307	Reductor
2	7 - 314	Reductor
.	.	
.	.	
.	.	
22	147 - 454	Reductor
23	154 - 461	Nada
24	161 - 468	Nada
.	.	
.	.	
.	.	
65	448 - 755	Nada

terminología. El conjunto de datos para entrenar una CNN-1D está compuesto por arreglos 2D (ventanas) que contienen las aceleraciones desplazadas en el tiempo de un reductor de velocidad. Dicha técnica de deslizamiento de ventanas sobre la señal se muestra en la Figura 3.5.

Inicialmente, se capturan las aceleraciones durante 756 deltas de tiempo, cantidad necesaria para cubrir completamente el paso sobre un reductor de velocidad. Estos datos se organizan como se muestra en la Figura 3.6, en una matriz con 756 filas, cada una correspondiente a un delta de tiempo, y tres columnas que representan las aceleraciones en los ejes x , y y z .

El conjunto de datos de entrenamiento, con una dimensión de 3×756 , se transpone para que las filas representen los tres ejes de aceleración (x , y , z) y las columnas correspondan a los deltas de tiempo. Para generar los arreglos bidimensionales, se aplica un deslizamiento de ventanas sobre este conjunto de datos. Cada ventana abarca 308 deltas de tiempo, lo que da como resultado un total de 65 ventanas.

Finalmente, al agrupar las 65 ventanas, se obtiene un arreglo de $65 \times 3 \times 308$, que representa el desplazamiento completo sobre el reductor de velocidad. En la Figura 3.6 se muestra la estructura de dicho arreglo, que contiene cada una de las ventanas que utilizará la CNN-1D para su entrenamiento. A cada arreglo de dos dimensiones que se utiliza como entrada para entrenar el algoritmo de inteligencia artificial, se le aplica una convolución unidimensional, de acuerdo con el método presentado en la subsección 2.8.1.

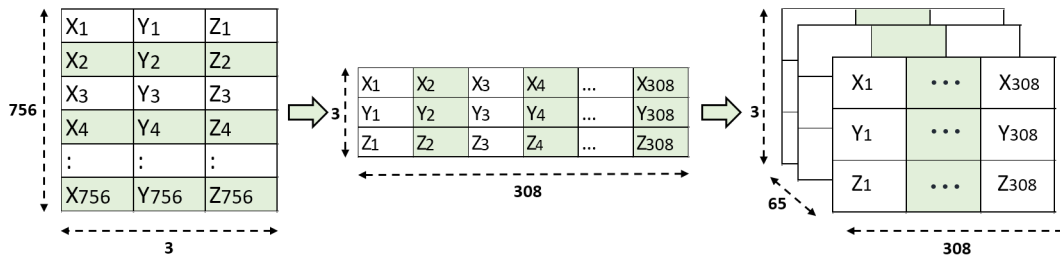


FIGURA 3.6: Estructura del arreglo utilizado como entrada para el entrenamiento de la CNN-1D.

3.3. Preprocesamiento de los Datos

La estructura de los datos es fundamental para lograr una buena exactitud en el modelo, pero el preprocesamiento también juega un papel clave, ya que impacta directamente en el desempeño de la CNN-1D.

Dado que el objetivo es trabajar en tiempo real, se priorizaron operaciones de bajo costo computacional para optimizar la respuesta del algoritmo y garantizar resultados en tiempo real.

El único preprocesamiento necesario fue la normalización de los datos. Para determinar la técnica más adecuada para la CNN-1D, se evaluaron tres enfoques diferentes:

- **Normalización en el rango [-1,1]**

Este método centra los valores alrededor de cero, lo cual es útil cuando es importante distinguir entre valores positivos y negativos.

$$A_i = \frac{A_m}{19.6132 \text{ m/s}^2} \tag{3.1}$$

donde:

- A_i : Matriz de aceleraciones normalizada en el rango [-1,1].
- A_m : Matriz de aceleraciones en m/s^2 .
- 19.6132 m/s^2 : Rango máximo de aceleración capturada.

- **Normalización en el rango [0,1]**

Esta técnica es común en redes neuronales, especialmente cuando la última capa utiliza la función de activación sigmoide en problemas de clasificación binaria, ya que el resultado será un valor entre 0 y 1 que representa la probabilidad de pertenecer a una clase.

$$A_k = \left(\frac{A_m}{19.6132 \text{ m/s}^2} + 1 \right) / 2 \quad (3.2)$$

donde:

- A_k : Matriz de aceleraciones normalizada en el rango [0,1].
- A_m : Matriz de aceleraciones en m/s^2 .
- 19.6132 m/s^2 : Rango máximo de aceleración capturada.
- **Normalización en escala de grises [0,255]**

Esta normalización se utiliza comúnmente en el procesamiento de imágenes, ya que ayuda a resaltar ciertas características y mejorar el contraste.

$$A_n = A_k * 255 \quad (3.3)$$

donde:

- A_n : Matriz de aceleraciones normalizada en el rango [0,255].
- A_k : Matriz de aceleraciones normalizada en el rango [0,1].
- 255: Valor máximo de un píxel en escala de grises.

Durante el entrenamiento, las tres técnicas mostraron buenos resultados. Sin embargo, en pruebas de campo, los desempeños variaron significativamente. Estos resultados se analizan con detalle en el capítulo 4.

Para este proyecto, se determinó que la normalización más adecuada corresponde a la conversión a escala de grises en el rango [0, 255]. La experimentación que respalda esta elección se presenta en la sección 4.2.

3.4. Conjunto de Datos de Entrenamiento

Para entrenar la red neuronal, se generaron distintos conjuntos de datos (*datasets*). Como se mencionó previamente, por cada reductor de velocidad se obtienen 65 ventanas, de las cuales aproximadamente 22 corresponden a la etiqueta "Reductor", mientras que las restantes se etiquetan como "Nada".

Dado que se requería un conjunto de datos con al menos cien reductores de velocidad, pero la disponibilidad de estos era limitada, principalmente por la logística, se optó por utilizar la técnica de *data augmentation* en los datos existentes para ampliar el *dataset*. Se generaron distintas combinaciones

con el objetivo de evaluar su impacto en el desempeño de la CNN-1D y determinar cuál configuración era la más adecuada.

En la Tabla 3.4 se presentan los seis conjuntos de datos utilizados en los experimentos, indicando la cantidad de ventanas en cada conjunto y la distribución de las etiquetas “Reductor” y “Nada”. La proporción de cada clase tuvo un impacto en el rendimiento final de la CNN-1D, especialmente en las pruebas en tiempo real. Los resultados obtenidos se analizarán en el Capítulo 4. Cabe destacar que el conjunto de datos seleccionado para entrenar el modelo de la CNN-1D en este proyecto fue el Dataset 6.

TABLA 3.4: Distribución de etiquetas en los conjuntos de datos de entrenamiento.

Conjunto de datos de	Etiquetas	
	Reductor	Nada
Dataset 1	219	6609
Dataset 2	219	2911
Dataset 3	1971	4663
Dataset 4	876	3290
Dataset 5	1752	8432
Dataset 6	12933	4599

3.5. Diseño de la Red Neuronal Convolutiva Unidimensional en Python

Una vez creado y normalizado el conjunto de datos (array tridimensional), se carga un archivo que contiene la etiqueta correspondiente para cada índice del array. Dicho archivo clasifica los datos en dos categorías: “Reductor” y “Nada”. Estas etiquetas se convierten a un formato binario, asignando el valor 1 a la clase “Reductor” y 0 a la clase “Nada”.

Posteriormente, los datos se dividen en conjuntos de entrenamiento y prueba utilizando la función *train_test_split*, reservando el 80% para entrenamiento y el 20% para prueba. Este paso es fundamental antes de definir los parámetros de la CNN-1D.

Una vez preparados los datos para el algoritmo de aprendizaje automático, se procede a configurar la arquitectura de la red neuronal convolutiva de una dimensión. La Figura 3.7 muestra la estructura del modelo. La primera capa es una convolución 1D con 32 filtros, un tamaño de kernel de 3, función de activación ReLU y *padding='same'* para mantener el tamaño de la

entrada. El modelo recibe datos con forma $(1,3,308)$, donde 1 representa el canal, 3 los renglones y 308 las columnas. A continuación, la capa *Flatten()* transforma la salida en un vector unidimensional, que se introduce en una capa densa con 32 neuronas y activación ReLU. Finalmente, una capa densa con una única neurona y activación sigmoide genera una salida entre 0 y 1 para la clasificación binaria.

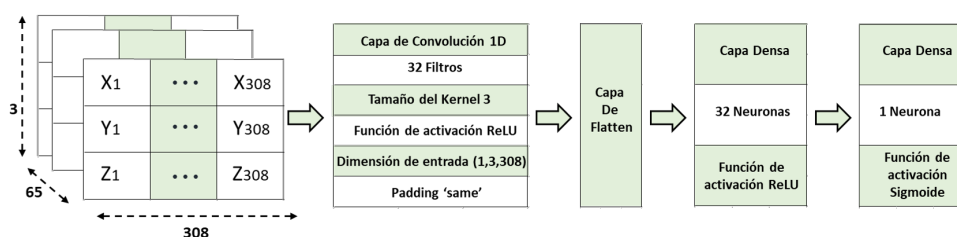


FIGURA 3.7: Diagrama del modelo de red neuronal convolucional 1D (CNN-1D) con una capa convolucional, una capa densa intermedia y una capa de salida para clasificación binaria.

3.5.1. Entrenamiento y Compilación de la CNN-1D en Python

Después de definir la estructura del modelo, es necesario compilarlo antes de proceder con el entrenamiento. Una vez entrenado, se guardan tanto los pesos como la arquitectura del modelo, lo que permite reutilizarlo en una función de Python sin necesidad de volver a entrenarlo. En la Figura 3.8 se muestra el código en Python para este proceso. Primero, el modelo se compila utilizando el optimizador Adam y la función de pérdida `binary_crossentropy`, adecuada para clasificación binaria. Luego, se genera un resumen de la arquitectura con `model.summary()`. A continuación, el modelo se entrena con los datos de entrenamiento (`X_train`, `y_train`) durante 10 épocas, con lotes de tamaño 10 y un 20 % de los datos reservados para validación. Posteriormente, la arquitectura del modelo se guarda en un archivo JSON (`modelo.json`), mientras que los pesos se almacenan en un archivo H5 (`modelo.h5`). Finalmente, el modelo se evalúa con los datos de prueba (`X_test`, `y_test`), y se imprimen la pérdida y la precisión obtenidas.

El Algoritmo 3 presenta el pseudocódigo que describe el proceso de carga y uso del modelo de la red neuronal convolucional unidimensional (CNN-1D) preentrenado, con el objetivo de realizar predicciones a partir de una matriz de datos de entrada (un array bidimensional).

En primer lugar, se importan las bibliotecas necesarias: `numpy` para el manejo de arreglos numéricos y `tensorflow` para trabajar con modelos de

```
# Compilar el modelo
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Resumen de la arquitectura
model.summary()

# Entrenar el modelo
history = model.fit(X_train, y_train, epochs=10, batch_size=10, validation_split=0.2)

# Guardar la arquitectura del modelo en formato JSON
model_json = model.to_json()
with open('modelo.json', 'w') as json_file:
    json_file.write(model_json)
# Guardar los pesos de la red neuronal
model.save_weights('modelo.h5')

# Validación
loss, accuracy = model.evaluate(X_test, y_test)

# Imprimir la pérdida y la exactitud del modelo
print(f'Pérdida: {loss}')
print(f'Accuracy: {accuracy}')
```

FIGURA 3.8: Código en Python para la compilación, entrenamiento y almacenamiento de un modelo de red neuronal convolucional 1D.

aprendizaje profundo. A continuación, la matriz de entrada se convierte a un arreglo de tipo float64 para garantizar la compatibilidad con el modelo. Posteriormente, se carga la arquitectura del modelo desde un archivo JSON y se cargan los pesos preentrenados desde un archivo en formato H5.

Una vez cargado el modelo, se realiza la predicción utilizando los datos de entrada. El resultado de la predicción se convierte a float64 antes de ser devuelto como salida, ya que, por defecto, las redes neuronales en Python retornan un valor de tipo float32. Es necesario forzar esta conversión a float64 para enviarlo a LabVIEW, dado que este espera recibir un dato en ese formato. La función retorna un valor numérico en el rango de 0 a 1, el cual es generado por la función de activación sigmoide en la última capa de la red neuronal. Este valor representa la probabilidad de que el array bidimensional de entrada pertenezca a la clase “Reductor”, donde 1 indica una alta probabilidad y 0 una baja probabilidad.

Esta estructura de la red se utiliza para cada uno de los conjuntos de datos generados, permitiendo entrenar y guardar distintos modelos. De esta manera, se pueden comparar sus desempeños y determinar cuál es el más eficiente, según el archivo de entrada utilizado para el entrenamiento del algoritmo.

Algoritmo 3 Función en Python para realizar predicciones con un modelo CNN-1D.

```
1: function HACER_PREDICCION(datos_matriz)
2:   Importar numpy como np
3:   Importar tensorflow como tf
4:   dato ← Convertir datos_matriz a un array de tipo float64
5:   Abrir archivo JSON del modelo
6:   leer_json ← Leer contenido del archivo JSON
7:   Cerrar archivo JSON
8:   modelo ← Cargar modelo desde JSON
9:   Cargar pesos del modelo desde archivo H5
10:  x ← Realizar predicción con modelo.predict(dato)
11:  prediccion ← Convertir x a tipo float64
12:  return prediccion
13: end function
```

3.6. Desarrollo de Programas en LabVIEW

Aunque el algoritmo de inteligencia artificial está desarrollado en Python, gran parte de la programación se realiza en LabVIEW. Esto se debe a que LabVIEW gestiona la comunicación con el MyRIO, adquiriendo los datos en tiempo real. Además, se emplea una interfaz de usuario en LabVIEW para visualizar la clasificación y las aceleraciones detectadas.

3.6.1. Programa en LabVIEW para la Adquisición de Señales

Se desarrolló un programa en LabVIEW para establecer comunicación con el acelerómetro del MyRIO, utilizando una versión de 32 bits debido a la compatibilidad del dispositivo. Para optimizar la adquisición de señales, se implementó la estructura Productor-Consumidor, lo que permite ejecutar en paralelo la adquisición y el preprocesamiento de datos. Esta estructura mejora el rendimiento del programa al optimizar recursos, minimizar la pérdida de datos mediante colas y mantener una interfaz de usuario más fluida.

La Figura 3.9 muestra la estructura del productor en LabVIEW, donde se configura la comunicación con el acelerómetro del MyRIO. En este bloque, se adquieren las aceleraciones en los ejes x , y y z , las cuales primero pasan por un subVI que las convierte de g a m/s^2 , ya que el MyRIO las proporciona en unidades g . El contenido de este subVI se muestra en la Figura 3.10. Una vez convertidas, las aceleraciones se almacenan en una cola para evitar la pérdida de datos y posteriormente son enviadas al ciclo del consumidor.

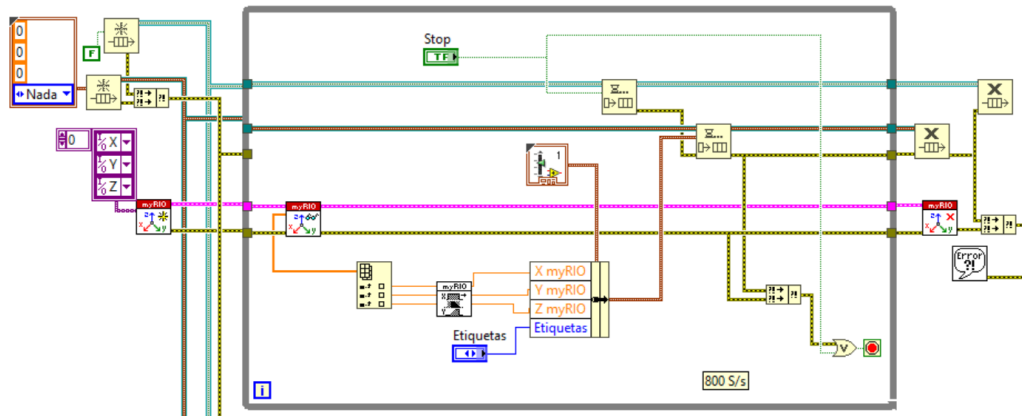


FIGURA 3.9: Estructura del productor en LabVIEW, donde se adquieren las señales del acelerómetro del MyRIO y se convierten a unidades de m/s^2 .

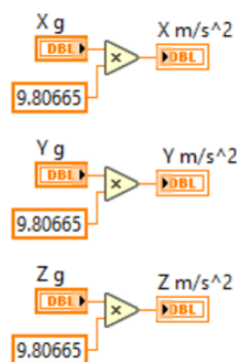


FIGURA 3.10: SubVI encargado de la conversión de aceleraciones de g a m/s^2 en LabVIEW.

Las aceleraciones leídas se visualizan en tiempo real mediante gráficos e indicadores. Además, pueden guardarse en un archivo al presionar un botón, asignándoles una etiqueta como “Reductor” o “Nada”. Simultáneamente, las aceleraciones almacenadas en las colas se envían a otro programa mediante variables compartidas.

En la Figura 3.11 se muestra la sección del consumidor en el programa de LabVIEW, donde se lleva a cabo este proceso. El esquema Productor-Consumidor permite procesar los datos en paralelo, evitando bloqueos en la adquisición y enviando en tiempo real las aceleraciones a otro programa de LabVIEW sin perder ningún dato, gracias al uso de colas.

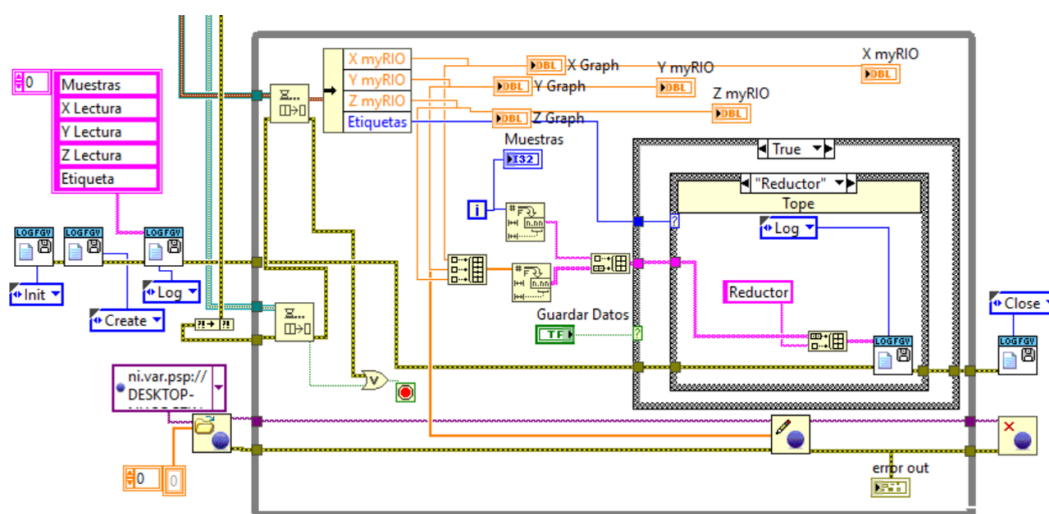


FIGURA 3.11: Estructura del productor en LabVIEW, donde las aceleraciones se visualizan, se guardan con su etiqueta y se envían en tiempo real a otro programa mediante variables compartidas.

En el panel frontal (interfaz de usuario) del programa en LabVIEW utilizado para la adquisición de señales (Figura 3.12), se pueden visualizar los siguientes elementos:

- Una gráfica independiente para cada aceleración medida en los ejes x , y y z .
- Un indicador numérico que muestra el valor en tiempo real de la aceleración en cada eje.
- Un botón para guardar los datos adquiridos.
- Un selector para asignar una etiqueta identificativa al archivo de las aceleraciones.

- Un botón de detención para finalizar la ejecución del programa.

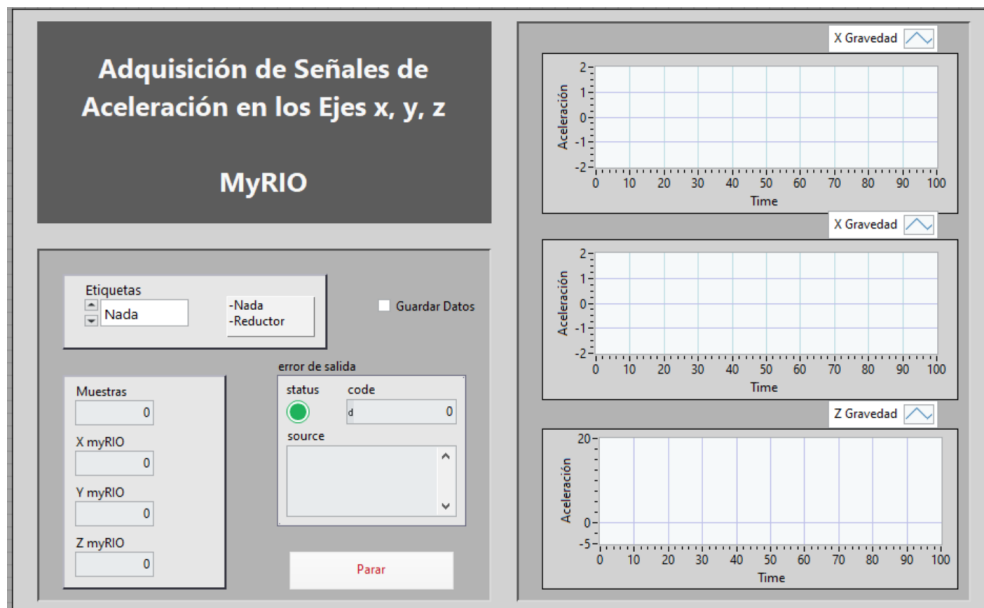


FIGURA 3.12: Panel frontal en LabVIEW para adquisición de señales de aceleración en los ejes x , y , z . Incluye gráficas por eje, indicadores numéricos, botones de guardado y parada.

3.6.2. Clasificación de Reductores de Velocidad en LabVIEW Mediante una CNN-1D en Python

Para establecer la comunicación con Python, se desarrolló un programa adicional en LabVIEW de 64 bits. Este programa interactúa con un nodo Python mediante llamadas a una función especializada, cuya tarea es predecir si la ventana actual de lecturas de aceleración corresponde al evento de un reductor de velocidad.

Al igual que el programa anterior, este implementa una arquitectura de *productor-consumidor*, a la que se añade un tercer ciclo paralelo dedicado exclusivamente a la clasificación. La estructura del sistema se compone de:

- **Ciclo productor:** Mostrado en la Figura 3.13, se encarga de:
 - Inicializar el sistema de colas.
 - Acceder a la variable compartida creada por el programa de adquisición de datos.
 - Leer continuamente las aceleraciones (convertidas a m/s^2) en los tres ejes (x , y , z).

- Alimentar las colas con los datos en tiempo real para evitar pérdida de información.

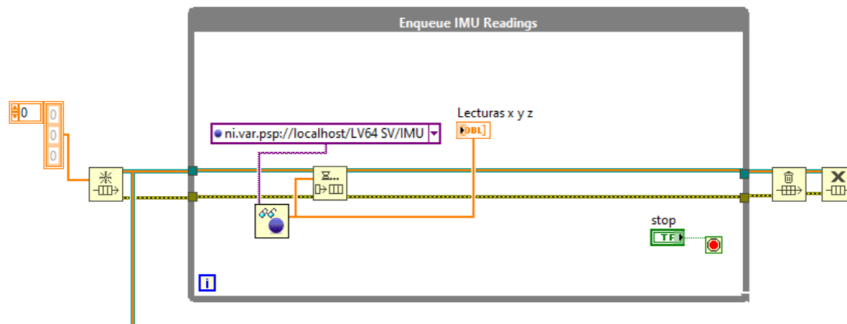


FIGURA 3.13: Diagrama del ciclo productor en LabVIEW. Muestra la inicialización de colas, lectura de la variable compartida y envío de datos de aceleración en tiempo real.

El ciclo consumidor (Figura 3.14) recibe las aceleraciones mediante colas y las organiza en ventanas de datos para su posterior clasificación. El proceso sigue este flujo:

1. Formación de la primera ventana:

- El sistema espera hasta acumular 308 muestras en la cola (equivalente a 0.385 segundos de datos).
- Estas muestras se estructuran en una matriz de dimensiones 3×308 (ejes x , y , z por estampa de tiempo).

2. Procesamiento continuo:

- Cada nueva ventana conserva 301 muestras históricas (98 % de los datos anteriores).
- Se añaden las 7 muestras más recientes (actualización del 2 %).
- Este esquema permite:
 - Mantener contexto temporal (301 muestras pasadas).
 - Incorporar nueva información (7 muestras actuales).
 - Reducir el intervalo entre ventanas a 8.75 ms.

3. Envío al clasificador:

- Cada ventana actualizada se transmite inmediatamente al siguiente ciclo.

- El proceso se repite indefinidamente hasta la interrupción del programa.

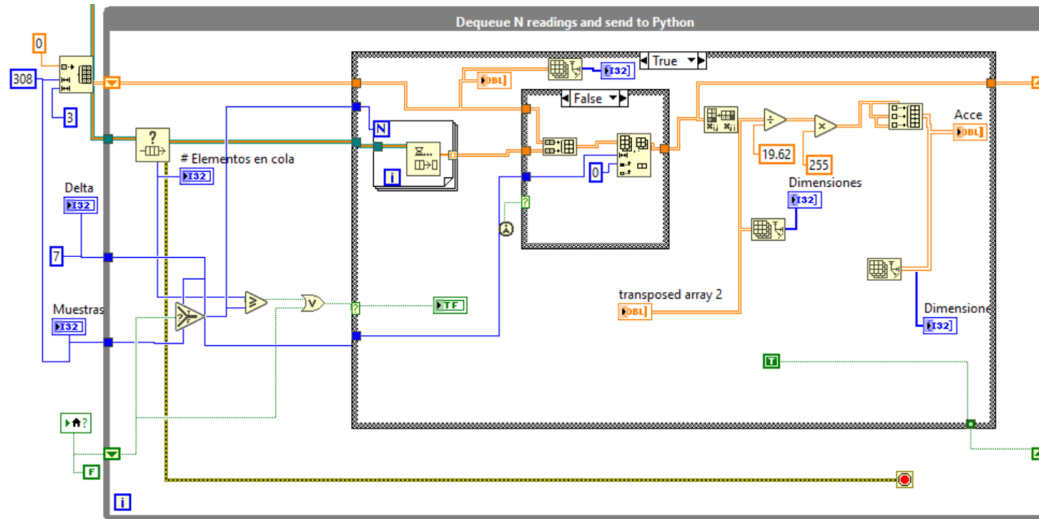


FIGURA 3.14: Arquitectura consumidor en LabVIEW: formación de ventanas temporales traslapadas (3×308) para alimentar la CNN-1D, con actualización cada 8.75 ms.

El ciclo de clasificación, representado en la Figura 3.15, implementa un sistema híbrido que combina LabVIEW y Python para procesar ventanas de aceleración de dimensiones 3×308 mediante una CNN-1D, siguiendo el flujo descrito en el Algoritmo 3. La arquitectura de este ciclo abarca cuatro etapas principales: (1) la recepción de la matriz de aceleración, (2) la invocación del modelo Python según el Algoritmo 3, (3) la toma de una decisión binaria basada en un umbral predefinido, y (4) la visualización de los resultados obtenidos.

La configuración inicial del sistema incluye la carga del entorno Python, especificando la ruta del script `function_topes.py`, y la inicialización única del modelo CNN-1D para garantizar un rendimiento eficiente durante la ejecución. El flujo de procesamiento comienza con la entrada de datos, donde la matriz `Acce` (3×308) es recibida desde el ciclo consumidor y convertida al tipo `float64`, tal como se detalla en la línea 4 del Algoritmo 3. A continuación, se realiza la predicción mediante la función `hacer_prediccion()`, que carga el modelo preentrenado en formato JSON/H5 (líneas 5–9 del algoritmo) y ejecuta el cálculo utilizando `model.predict()` (línea 10). Finalmente, el sistema toma una decisión binaria basada en el valor de probabilidad obtenido: si $P \geq 0.5$, se clasifica como `True` (indicando la presencia de un reductor),

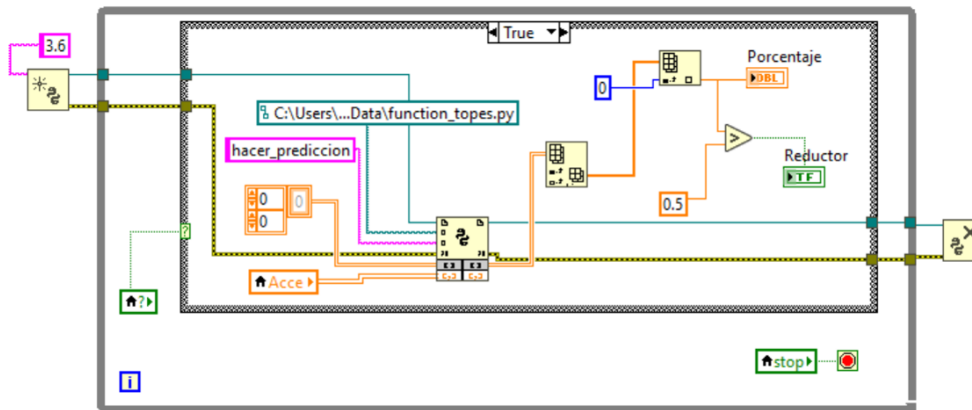


FIGURA 3.15: Arquitectura del ciclo de clasificación. (1) Recepción de matriz 3×308 , (2) Invocación del modelo Python (Algoritmo 3), (3) Decisión binaria basada en umbral, y (4) Visualización de resultados.

mientras que si $P < 0.5$, se devuelve False (sin detección). Este proceso permite una clasificación automatizada y eficiente, integrando las capacidades de LabVIEW para la adquisición de datos y Python para el procesamiento mediante redes neuronales.

La Figura 3.16 muestra la interfaz de usuario del programa desarrollado en LabVIEW, donde se visualizan en tiempo real:

- Las ventanas de aceleración en sus distintas etapas de preprocesamiento.
- El porcentaje de probabilidad estimado por la CNN-1D.
- Un LED que se activa cuando la función sigmoide (Ecuación 3.4) supera el umbral de 0.5, indicando la detección de un reductor de velocidad.

$$p[S(x)] = \begin{cases} 1, & \text{sí } S(x) \geq 0.5 \\ 0, & \text{otro caso} \end{cases} \quad (3.4)$$

Donde $p[S(x)]$ es el valor booleano que indica si se trata de un reductor de velocidad, y $S(x)$ representa el valor obtenido por la función sigmoide.

El diseño es intuitivo y proporciona retroalimentación visual inmediata. El LED se enciende únicamente ante la detección de reductores, minimizando falsas alarmas por otras irregularidades del camino. Además, la interfaz

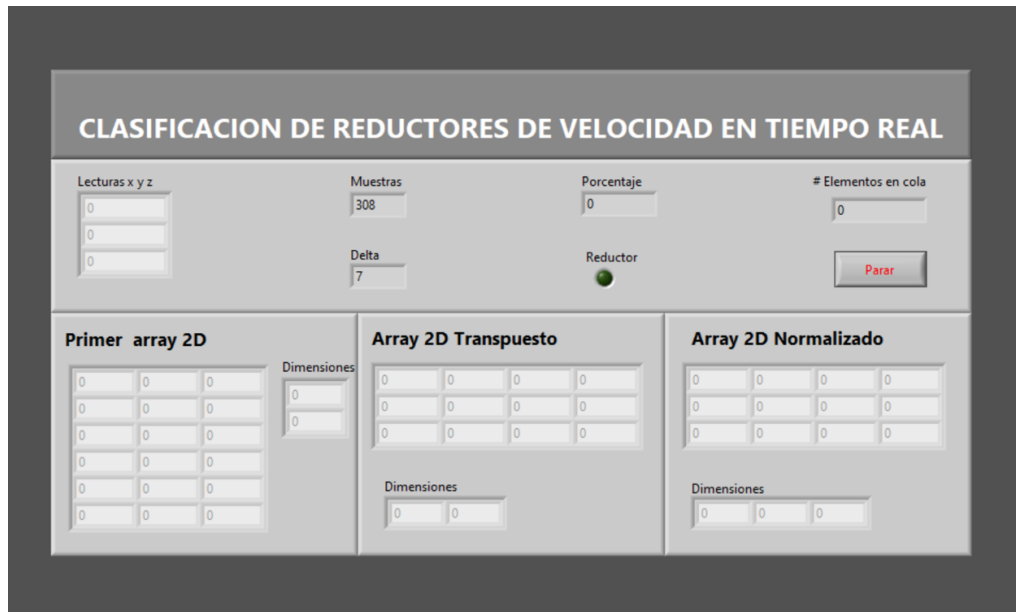


FIGURA 3.16: Interfaz de usuario: visualización de datos en tiempo real y resultado de la clasificación.

permite verificar la coherencia del proceso en tiempo real al mostrar la matriz de aceleración enviada al nodo Python y el porcentaje de probabilidad calculado.

Para evaluar la efectividad de la CNN-1D y la metodología propuesta, se realizaron pruebas en campo, analizando el desempeño del sistema en condiciones reales. La discusión de estos resultados se presenta en el Capítulo 4.

3.6.3. Optimización del Umbral de Decisión

Durante el desarrollo del proyecto, se realizó un análisis teórico para determinar el umbral de decisión óptimo utilizando la curva ROC. Este análisis se basó en el modelo entrenado, pero no se aplicó experimentalmente. El índice de Youden, que maximiza la diferencia entre la tasa de verdaderos positivos (TPR) y la tasa de falsos positivos (FPR), fue utilizado para evaluar el umbral más adecuado para la clasificación.

El índice de Youden se define como:

$$\theta_{\text{óptimo}} = \underset{\theta}{\operatorname{argmax}} (TPR(\theta) - FPR(\theta)) \quad (3.5)$$

donde:

- $TPR(\theta)$: Tasa de verdaderos positivos, o sensibilidad.

- $FPR(\theta)$: Tasa de falsos positivos, equivalente a $1 -$ especificidad.
- θ : Umbral evaluado en el rango $[0,1]$.

A pesar de que se generó una gráfica ROC (Figura 3.17) que mostró un valor óptimo de $\theta = 0.97$, indicando que el modelo requeriría una alta certeza para clasificar un evento como reductor, este análisis quedó en el ámbito teórico y no se implementó experimentalmente en el proyecto. Sin embargo, esta información es importante y se puede considerar en trabajos futuros.

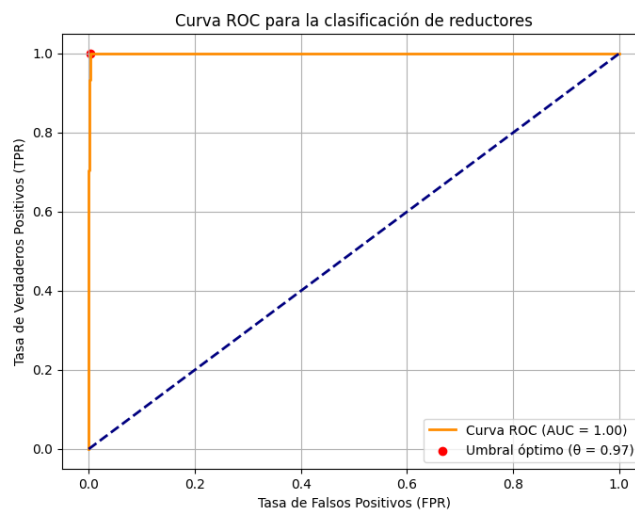


FIGURA 3.17: Curva ROC del Modelo 6. El punto óptimo (rojo) en $\theta = 0.97$ maximiza la diferencia entre sensibilidad y especificidad.

Capítulo 4

Experimentos y Análisis de Resultados

Dado que el objetivo principal consistía en encontrar la configuración óptima para que una red neuronal convolucional unidimensional (1D-CNN) lograra clasificar en tiempo real el paso por reductores de velocidad, se realizaron diversos experimentos cuyos resultados se detallan en este capítulo.

4.1. Experimentación con Diferentes Modelos

Con el objetivo de encontrar una arquitectura adecuada para el problema planteado, se desarrollaron y evaluaron seis modelos distintos. Estos se agrupan en dos etapas: la primera corresponde a modelos exploratorios, en los que se probaron diferentes configuraciones sin obtener resultados satisfactorios. Sin embargo, su análisis permitió identificar áreas de mejora. La segunda etapa se enfoca en modelos optimizados, diseñados a partir de los aprendizajes obtenidos en la fase inicial, y cuyos resultados fueron significativamente mejores.

4.1.1. Modelos Iniciales y Análisis de sus Limitaciones

En esta subsección se presentan los primeros cuatro modelos implementados. Aunque sus resultados no fueron óptimos, su análisis permitió identificar las principales limitaciones y errores de diseño, lo cual resultó fundamental para el desarrollo de modelos más eficientes.

Modelo 1

- **Tamaño del dataset:** 6,609 ventanas
- **Distribución de clases:**
 - *Reductor:* 219 ventanas (3.3136 %)
 - *Nada:* 6,390 ventanas (96.6863 %)

- **Rendimiento:** Durante las pruebas en tiempo real, el modelo clasificó correctamente los reductores cuando el vehículo circulaba hasta 30 km/h. A velocidades mayores, mostró dificultades en la detección.

Modelo 2

- **Tamaño del dataset:** 2,911 ventanas
- **Distribución de clases:**
 - *Reductor:* 219 ventanas (7.5231 %)
 - *Nada:* 2,692 ventanas (92.4768 %)
- **Rendimiento:** Inicialmente, el modelo mostró resultados prometedores; sin embargo, clasificaba erróneamente cualquier evento de frenado como Reductor. Esto se debía a que, al aproximarse a un reductor de velocidad, es común frenar, lo que generó un sesgo en los resultados.

Modelo 3

- **Tamaño del dataset:** 4,663 ventanas
- **Distribución de clases:**
 - *Reductor:* 1,971 ventanas (42.2689 %)
 - *Nada:* 2,692 ventanas (57.731 %)
- **Rendimiento:** Mostró un alto índice de falsos positivos, clasificando como *Reductor* situaciones de conducción normal, lo que impide su uso en aplicaciones reales.

Modelo 4

- **Tamaño del dataset:** 3,290 ventanas
- **Distribución de clases:**
 - *Reductor:* 876 ventanas (26.6261 %)
 - *Nada:* 2,414 ventanas (73.3738 %)
- **Rendimiento:** Detectó correctamente reductores, pero generó clasificaciones erróneas durante giros, paso sobre alcantarillas y cambios en la inclinación del vehículo.

4.1.2. Modelos Mejorados con Base en Resultados Previos

Esta subsección describe los dos modelos desarrollados con base en los hallazgos de la etapa anterior. El análisis de los primeros modelos mostró que el sistema tendía a asociar los eventos de frenado con la presencia de reductores, así como con giros, inclinaciones del vehículo o el paso por irregularidades como alcantarillas. Para corregir este sesgo, se recolectaron nuevos datos que incluían muestras explícitas de frenado y otros eventos, etiquetados como “Nada”.

Modelo 5

- **Tamaño del dataset:** 3,290 ventanas
- **Distribución de clases:**
 - *Reductor*: 876 ventanas (26.6261 %)
 - *Nada*: 2,414 ventanas (73.3738 %)
- **Rendimiento:** Mejoró significativamente en la discriminación entre frenados y reductores, aunque aún presentaba falsos positivos durante giros y paso sobre obstáculos.

Modelo 6

- **Tamaño del dataset:** 12,933 ventanas
- **Distribución de clases:**
 - *Reductor*: 4,599 ventanas (35.5601 %)
 - *Nada*: 8,334 ventanas (64.4398 %)
- **Rendimiento:** Demostró clasificación robusta, identificando correctamente los reductores independientemente de giros, inclinaciones, cambios de velocidad u otros eventos. No generó falsos positivos en las condiciones de prueba.

El Modelo 6 demostró ser el más preciso y robusto para la aplicación en tiempo real, por lo que se seleccionó para entrenar la CNN-1D.

4.2. Evaluación del Impacto de la Normalización en cada Modelo

Además de evaluar el modelo con diversos conjuntos de datos, se aplicaron diferentes técnicas de normalización para preparar las entradas del algoritmo de inteligencia artificial: se utilizó la normalización en el rango de 0 a 1 para estandarizar los valores en una escala positiva, la normalización de -1 a 1 para preservar la simetría de los datos alrededor del cero, y la normalización de 0 a 255, común en aplicaciones de procesamiento de imágenes. Los resultados comparativos de estos enfoques se presentan en la Tabla 4.1, permitiendo analizar el impacto de cada método en el rendimiento del modelo.

TABLA 4.1: Comparación del desempeño de los diferentes modelos con diferentes técnicas de normalización.

Modelo	Normalización	Exactitud	Perdida
1	[0, 255]	0.9727	0.1545
	[-1, 1]	0.8997	0.2356
	[0, 1]	0.8834	0.2276
2	[0, 255]	0.9708	0.108
	[-1, 1]	0.8345	0.1918
	[0, 1]	0.8779	0.1576
3	[0, 255]	0.9882	0.079
	[-1, 1]	0.8508	0.334
	[0, 1]	0.8819	0.1062
4	[0, 255]	0.9863	0.0921
	[-1, 1]	0.8999	0.2298
	[0, 1]	0.8989	0.2091
5	[0, 255]	0.994	0.00097
	[-1, 1]	0.9016	0.1005
	[0, 1]	0.9147	0.1173
6	[0, 255]	0.9961	0.0204
	[-1, 1]	0.9153	0.035
	[0, 1]	0.9267	0.0045

Un aspecto clave identificado en los experimentos fue el impacto del pre-procesamiento en el desempeño del modelo. Aunque los datos de entrada eran señales temporales de aceleración, la normalización en el rango [0, 255], común en el procesamiento de imágenes, resultó ser la más efectiva en todos los modelos evaluados. Este resultado es especialmente interesante, ya que, incluso al emplear una CNN-1D diseñada para datos secuenciales, dicho pre-procesamiento superó de forma consistente a otras técnicas de normalización

como $[0,1]$ o $[-1,1]$. Esto sugiere que la escala de grises preserva mejor las características dinámicas de las aceleraciones para el aprendizaje automático.

Durante las pruebas en tiempo real, el Modelo 6 confirmó esta tendencia al alcanzar la mayor precisión con esta normalización, evidenciando su impacto tanto en el entrenamiento como en la generalización. Este desempeño podría explicarse por tres factores clave: (1) su capacidad para manejar la variabilidad en señales dinámicas, especialmente en frenados bruscos o giros repentinos, ya que el conjunto de datos incluía información explícita de estos eventos; (2) su sinergia con arquitecturas convolucionales, donde la escala ampliada ($[0, 255]$) potencia la activación de filtros en capas iniciales; y (3) el balance de clases en el conjunto de datos, que, junto con un volumen adecuado de muestras, facilitó la captura de patrones diversos.

Este estudio muestra cómo técnicas de preprocesamiento del dominio visual pueden adaptarse con éxito al análisis de series temporales, incluso en arquitecturas especializadas como CNN-1D. Así, se refuerza la importancia de no solo optimizar la calidad y diversidad de los datos, sino también explorar estrategias de normalización que, aun provenientes de otros campos, pueden aportar ventajas inesperadas en el procesamiento de señales.

4.3. Análisis de Robustez del Modelo 6

El Modelo 6 demostró un rendimiento superior durante todas las fases de experimentación. Para su desarrollo, se utilizó una partición estándar del 80 % de los datos para entrenamiento y 20 % para validación. Los resultados obtenidos confirman la efectividad de la arquitectura propuesta.

4.3.1. Proceso de Entrenamiento

La Figura 4.1 muestra la evolución de la función de pérdida y exactitud durante las 14 épocas de entrenamiento. Se observa:

- Convergencia estable en ambas curvas (entrenamiento y validación)
- Pérdida final: 0.0204
- Exactitud final: 0.9961

La disminución monótona de la pérdida y el aumento consistente de la exactitud indican un entrenamiento exitoso sin sobreajuste.

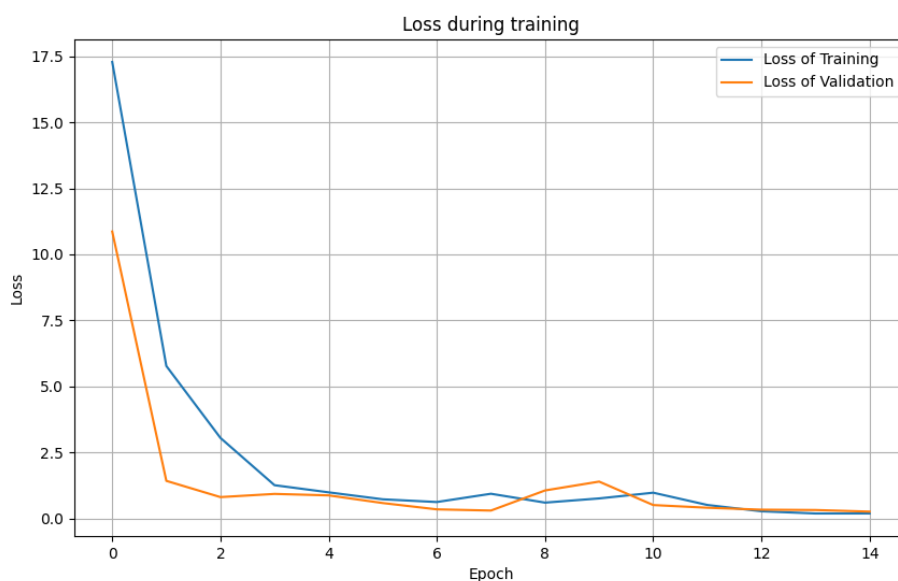


FIGURA 4.1: Curvas de pérdida para entrenamiento y validación del Modelo 6.

4.3.2. Evaluación del Desempeño

Las métricas en el conjunto de prueba (Tabla 4.2) revelan un excelente rendimiento:

TABLA 4.2: Métricas de evaluación del Modelo 6.

Métrica	Valor
Precisión	0.9894
Sensibilidad (Recall)	1.0000
F1-Score	0.9947
AUC-ROC	0.9996

La matriz de confusión (Figura 4.2) detalla:

- 0 falsos negativos: Detección perfecta de reductores
- Solo 10 falsos positivos (0.39 % de error)
- 2,577 clasificaciones correctas (99.61 % de acierto)

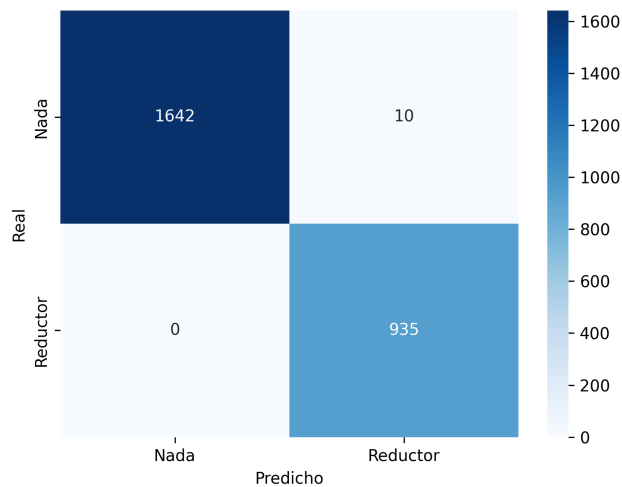


FIGURA 4.2: Matriz de confusión para el conjunto de prueba (n=2,587).

4.4. Análisis de las Ventanas de Entrada en la CNN-1D para Detección de Reductores de Velocidad

Esta sección demuestra la efectividad del método propuesto para procesar señales de aceleración mediante ventanas temporales en una CNN-1D, logrando una detección robusta de reductores de velocidad en tiempo real.

4.4.1. Configuración de las Ventanas de Entrada

La arquitectura recibe ventanas de dimensión 3×308 , donde:

- Cada fila corresponde a un eje de aceleración (x, y, z)
- Cada columna representa un intervalo de tiempo (Δt)
- Los valores están normalizados en el rango [0, 255]

4.4.2. Visualización Temporal de las Señales

La Figura 4.3 muestra una ventana al inicio del paso por un reductor, donde solo los últimos 7 puntos (2.27 % de la ventana) corresponden al evento real, mientras que los 301 puntos restantes capturan el estado previo.

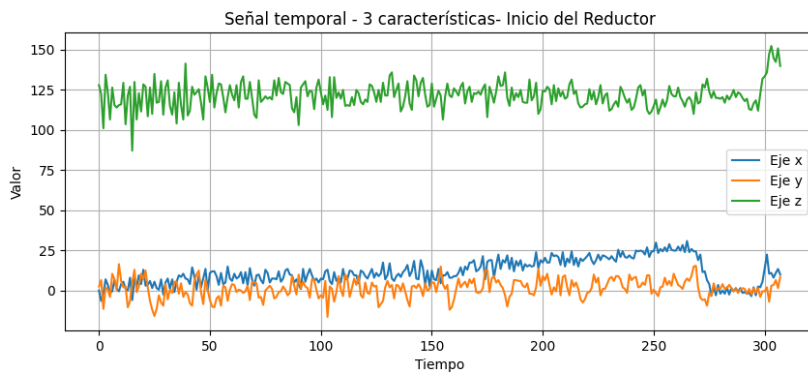


FIGURA 4.3: Señal temporal al inicio de un reductor. Los últimos 7 muestreos contienen los primeros indicios del evento.

En contraste, la Figura 4.4 presenta una ventana donde aproximadamente 140 puntos (45 % de la ventana) corresponden al paso por el reductor..

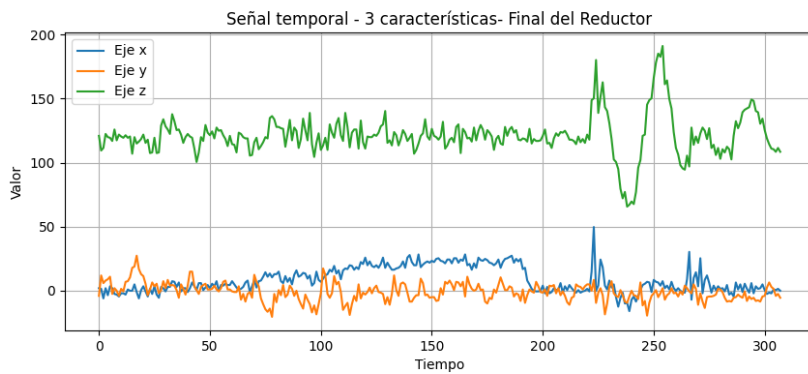


FIGURA 4.4: Señal temporal al finalizar un reductor. Donde aproximadamente 140 puntos pertenecen al evento de pasar el reductor de velocidad.

La Figura 4.5 ilustra una ventana etiquetada como *Nada*, donde no ocurre interacción con reductores. La normalización [0,255] provee un rango dinámico amplio que facilita la discriminación de patrones.

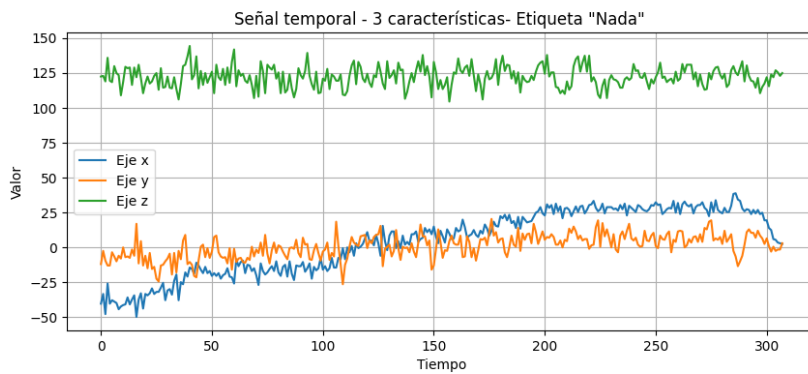


FIGURA 4.5: Señal temporal de referencia sin presencia de reductores. La amplitud constante muestra el patrón característico de conducción normal.

4.4.3. Representación en Escala de Grises

Las Figuras 4.6-4.8 muestran las mismas ventanas como imágenes en escala de grises. Aunque visualmente indistinguibles para humanos, la CNN-1D identifica patrones mediante:

- Filtros convolucionales que extraen características locales
- Patrones espacio-temporales en las tres dimensiones
- Diferencias sutiles en texturas y gradientes

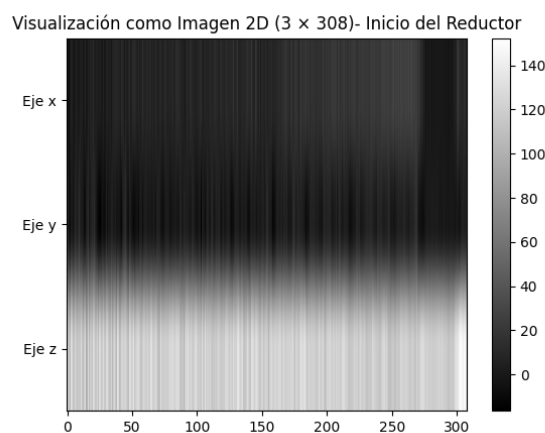


FIGURA 4.6: Representación en escala de grises de la Figura 4.3. Esta señal corresponde al evento de comenzar a pasar el reductor de velocidad.

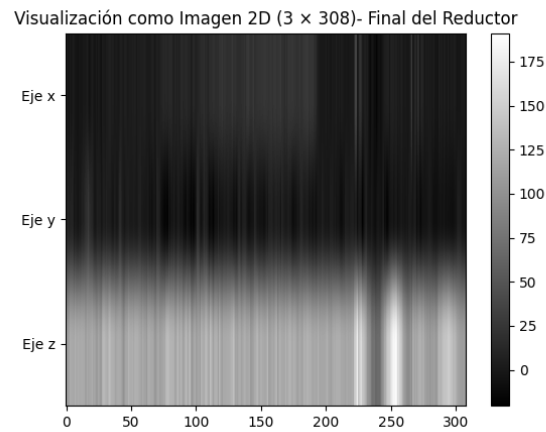


FIGURA 4.7: Representación en escala de grises de la Figura 4.4. Esta señal corresponde al evento de estar terminando de pasar el reductor de velocidad.

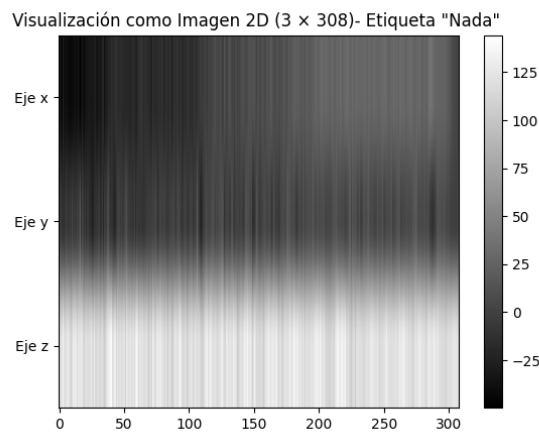


FIGURA 4.8: Representación en escala de grises de la Figura 4.5. Esta señal corresponde al evento denominado *Nada*, donde la ventana no contiene ninguna aceleración asociada al paso por un reductor de velocidad.

Al terminar el análisis, se puede afirmar que la configuración de las ventanas de entrada con dimensiones 3×308 captura de manera efectiva los eventos relevantes. La normalización de los datos en el rango $[0, 255]$ contribuye significativamente a la optimización del aprendizaje del modelo. Además, la CNN-1D, mediante sus filtros convolucionales, es capaz de extraer patrones que, aunque visualmente indistinguibles para el ojo humano, son fundamentales para la detección de reductores de velocidad en tiempo real.

4.5. Análisis del Tiempo de Ventaneo en Clasificación en Tiempo Real

El sistema propuesto supera las limitaciones de los enfoques tradicionales al implementar un esquema de ventaneo dinámico con actualización parcial, permitiendo la clasificación *en tiempo real* de reductores de velocidad. A diferencia de otros métodos utilizados — que procesan ventanas completas sin solapamiento — nuestra arquitectura reduce el intervalo entre clasificaciones a **8.75 ms**.

Mecanismo de Actualización Parcial: El sistema opera con una frecuencia de muestreo de 800 Hz ($T_{\text{muestra}}=1.25\text{ms}$). El tiempo de ventaneo se calcula como:

1. **Inicialización:** 0.385 s para llenar la primera ventana.

$$308, \text{muestras} \times 1.25, \text{ms/muestra} = 385, \text{ms} = 0.385, \text{s} \quad (4.1)$$

2. **Actualización continua:** 7 muestras nuevas (8.75 ms) desplazan el 2 % de la ventana, preservando el 98 % de los datos históricos. Este solapamiento del 98 % garantiza continuidad temporal mientras minimiza latencia.

$$7 \times 1.25 \text{ ms} = 8.75 \text{ ms} \quad (4.2)$$

Implicaciones en Tiempo Real: El intervalo de 8.75 ms entre clasificaciones cumple con los requisitos para considerarse tiempo real en esta aplicación, donde se requiere una respuesta en menos de 0.5 segundos para una ubicación exacta del reductor. Esta mejora es significativa al compararla con los trabajos previos:

- **Salman y Mian, 2023:** Desfase de 5 segundos en la clasificación
- **Celaya-Padilla et al., 2018:** Latencia de 2 segundos
- **Propuesta actual:** 8.75 ms (mejora de $570\times$ y $228\times$ respectivamente)

La capacidad del sistema para realizar clasificaciones en menos de 10 ms representa un avance importante hacia el procesamiento en tiempo real, lo cual es clave para futuras integraciones con sistemas de georreferenciación. Este comportamiento se logra mediante una estrategia de ventaneo dinámico, que reutiliza el 98 % de los datos previos, y una arquitectura paralela

implementada en LabVIEW. A diferencia de otros enfoques revisados, que pueden presentar desfases del orden de varios segundos, esta propuesta reduce significativamente dicho intervalo, permitiendo una posible correlación más precisa entre eventos dinámicos y coordenadas geográficas en desarrollos posteriores.

Aunque el presente sistema no incorpora GPS, su desempeño sugiere que podría ser empleado en aplicaciones como la generación de mapas viales, la validación en campo de reductores de velocidad y la detección de elementos no registrados en bases de datos oficiales. Sin afirmar una superioridad generalizada frente al estado del arte, los resultados obtenidos evidencian un desempeño competitivo y alineado con los requisitos de clasificación en tiempo real.

Capítulo 5

Conclusiones

El análisis experimental realizado ha demostrado que la configuración ideal para el sistema de detección de reductores de velocidad se basa en ventanas temporales de 3×308 muestras (correspondientes a los tres ejes del acelerómetro), con normalización en el rango [0-255]. Esta estructura ha mostrado una mayor precisión comparada con otros métodos de normalización, alcanzando un 99.61 % de exactitud en la clasificación. Además, el esquema de actualización parcial con solapamiento del 98 % y procesamiento de nuevos paquetes de 7 muestras cada 8.75 ms ha permitido mantener la operación en tiempo real sin comprometer el contexto temporal necesario para la detección precisa.

El Modelo 6, producto final de esta investigación, demostró un desempeño excepcional en las pruebas de validación, alcanzando cero falsos negativos y solo 10 falsos positivos en un conjunto de 2,587 muestras, lo que representa un error de apenas 0.39 %. Su robustez se evidenció al mantener un alto nivel de precisión bajo condiciones adversas, tales como frenadas bruscas, giros cerrados y superficies irregulares, superando ampliamente el umbral del 90 % de exactitud establecido como objetivo. Además, la arquitectura CNN-1D implementada mostró una notable eficiencia computacional, con latencias consistentemente inferiores a 10 ms, lo que supone una mejora de 228 veces en comparación con soluciones convencionales reportadas en la literatura. Por ejemplo, el método de Salman y Mian, 2023 presenta un desfase en la clasificación de reductores de velocidad de hasta 5 segundos, mientras que el de Celaya-Padilla et al., 2018 alcanza un retraso de 2 segundos.

5.1. Conclusiones Generales

Este trabajo ha cumplido satisfactoriamente con el objetivo principal de determinar la estructura adecuada de entradas para una red neuronal convolucional que procesa datos de aceleración de una IMU, desarrollando un sistema capaz de clasificar reductores de velocidad con alta precisión en tiempo real. Los resultados obtenidos validan la efectividad de la metodología

propuesta, que combina una segmentación temporal cuidadosa de las señales inerciales con técnicas avanzadas de preprocesamiento y una arquitectura de red neuronal configurada adecuadamente.

La investigación ha demostrado que las redes neuronales convolucionales unidimensionales (CNN-1D) representan una solución idónea para el procesamiento de señales inerciales en aplicaciones de movilidad inteligente. Este enfoque establece un nuevo referente en cuanto a técnicas de preprocesamiento para datos de una IMU. El sistema desarrollado no solo superó los parámetros de desempeño establecidos inicialmente, sino que también demostró un notable potencial para su implementación en escenarios prácticos, tales como el mapeo automatizado de infraestructura vial, sistemas avanzados de asistencia al conductor y programas de mantenimiento predictivo de vías. Estos hallazgos abren nuevas oportunidades para futuras investigaciones, las cuales podrían integrar este desarrollo con sistemas de posicionamiento global (GPS) y extender su aplicación a la detección de otros tipos de obstáculos viales.

5.2. Cumplimiento de Objetivos

A continuación se presentan los objetivos cumplidos durante esta investigación:

5.2.1. Recolección de Datos de Aceleración Desde una IMU

Se configuró con éxito un sistema de adquisición de datos que capturó las aceleraciones en los tres ejes (A_x , A_y , A_z) con una frecuencia de muestreo de 800 Hz, garantizando mediciones precisas en tiempo real. Estos datos fueron fundamentales para detectar cambios característicos producidos por reductores de velocidad.

5.2.2. Diseño e Implementación del Preprocesamiento de Datos

Se desarrolló una secuencia de preprocesamiento que incluyó normalización en el rango [0-255] y segmentación en ventanas temporales de 308 muestras. Esta metodología resultó ser superior a otras técnicas de normalización, mejorando significativamente la exactitud de la clasificación. La estrategia

de ventaneo dinámico con actualización parcial (7 muestras cada 8.75 ms) optimizó el balance entre contexto temporal y latencia computacional.

5.2.3. Identificación de Características Relevantes para la CNN

El análisis de los datos preprocesados reveló que los patrones temporales en los tres ejes de aceleración, especialmente las variaciones bruscas en Az (eje vertical), constituyen las características más discriminativas para identificar reductores de velocidad. Estas características fueron preservadas efectivamente mediante la normalización en escala de grises [0-255].

5.2.4. Metodología para Estructurar las Entradas Adecuadas para la CNN

Tras evaluar múltiples configuraciones, se determinó que la estructura óptima para las entradas de la CNN consistía en matrices de 3×308 (tres ejes \times tiempo), organizadas como imágenes en escala de grises. Esta representación permitió a la red extraer patrones espaciales y temporales con mayor eficacia que otros formatos probados.

5.2.5. Entrenamiento y Validación del Modelo CNN

El proceso iterativo de entrenamiento produjo seis modelos, culminando en el Modelo 6, el cual alcanzó una exactitud de 99.61 %, superando el 90 % requerido. La validación cruzada demostró su robustez, con cero falsos negativos y solo un 0.39 % de falsos positivos, incluso ante maniobras complejas como frenadas bruscas o giros cerrados.

5.2.6. Integración del Sistema en Tiempo Real

La implementación final logró clasificación continua con latencia garantizada menor a 10 ms, cumpliendo con los requisitos de tiempo real. Las pruebas en condiciones reales de circulación verificaron que el sistema mantiene su precisión ante variaciones de velocidad y diferentes tipos de reductores.

5.3. Consideraciones Finales

El sistema desarrollado ha logrado integrar exitosamente la adquisición de datos, el entrenamiento del modelo y la ejecución en tiempo real dentro de

un mismo flujo funcional. Esta solución representa un avance hacia sistemas de monitoreo inteligente, con aplicaciones en vehículos y plataformas móviles. La capacidad de detección basada en IA, junto con sensores inerciales de bajo costo, ofrece una alternativa eficiente frente a soluciones tradicionales basadas en visión por computadora o hardware especializado.

Las principales contribuciones incluyen:

- Un protocolo estandarizado para adquisición y preprocesamiento de datos de una IMU en aplicaciones viales.
- La demostración práctica de que representaciones tipo imagen (escala de grises) optimizan el desempeño de CNN-1D con señales temporales.
- Un sistema de clasificación en tiempo real con latencias 228 veces menores que soluciones previas.

5.4. Recomendaciones y Trabajo a Futuro

A partir de los resultados obtenidos y las limitaciones identificadas durante el desarrollo del sistema, se proponen las siguientes recomendaciones y líneas de trabajo a futuro:

- **Ampliación del conjunto de datos:** Se recomienda recolectar un conjunto más amplio y diverso de datos, incluyendo diferentes tipos de reductores de velocidad, velocidades de tránsito, superficies de rodamiento y condiciones climáticas.
- **Uso de sensores adicionales:** Integrar sensores complementarios como giroscopios y GPS podría mejorar la precisión y confiabilidad de la clasificación.
- **Evaluación en escenarios reales:** Probar el sistema en diferentes contextos urbanos y rurales permitirá una evaluación más completa de su desempeño y robustez.
- **Migración a plataformas embebidas:** Implementar el sistema en hardware más compacto y autónomo, como microcontroladores con soporte para IA, puede ampliar su aplicabilidad.

item **Integración con mapeo de ubicaciones:** Se sugiere integrar el sistema con plataformas de mapeo para marcar la ubicación de los reductores de velocidad detectados. Esta funcionalidad también podría extenderse a otras irregularidades viales, permitiendo mapear y visualizar en tiempo real los eventos ocurridos, lo que facilitaría la implementación de aplicaciones como el mapeo de infraestructura vial y sistemas de asistencia al conductor.

Bibliografía

- Baltrėnas, Hab P, Tomas Januševičius y Aleksandras Chlebnikovas (2017). «Research into the impact of speed bumps on particulate matter air pollution». En: *Measurement* 100, págs. 62-67.
- Belloch Díaz, Orlando (2019). «Sistema de detección del estado de la calzada basado en sensores de vibración y sistemas de cómputo de bajo coste». B.S. thesis.
- Bengio, Yoshua, Ian Goodfellow, Aaron Courville et al. (2017). *Deep learning*. Vol. 1. MIT press Cambridge, MA, USA.
- Carden, E Peter y Paul Fanning (2004). «Vibration based condition monitoring: a review». En: *Structural health monitoring* 3.4, págs. 355-377.
- Castro Toscano, Moisés Jesús (2019). «Determinación de trayectorias con mediciones inerciales para el seguimiento de posición y orientación». En.
- Celaya-Padilla, Jose M et al. (2018). «Speed bump detection using accelerometer features: A genetic algorithm approach». En: *Sensors* 18.2, pág. 443.
- Chapa, Jose M et al. (2020). «Accelerometer systems as tools for health and welfare assessment in cattle and pigs—a review». En: *Behavioural Processes* 181, pág. 104262.
- Chen, Hua-Peng y Nenad Bicanic (2000). «Assessment of damage in continuum structures based on incomplete modal information». En: *Computers & Structures* 74.5, págs. 559-570.
- Cherrez Illescas, Francisco Esteban (2022). «Desarrollo de una mesa vibratoria para simulación de sismos en estructuras». B.S. thesis. Universidad del Azuay.
- Cifuentes, Lizette Vanessa Tello, Johannio Marulanda y Peter Thomson (2021). «Detección de grietas en el pavimento usando técnicas de procesamiento de imágenes y redes neuronales artificiales». En: *Encuentro Internacional de Educación en Ingeniería*.
- Fajardo Viloría, Blanca (2023). «Utilización de acelerómetros para determinar de forma automatizada el comportamiento de cabras lecheras». En.
- Flores Castro, Juan Gerardo (2023). «Diseño de acelerómetros capacitivos con brazos basados en arreglos flexibles». En.

- Galeano, Iván Andrés Serna, Jorge Andrés Herrera Maldonado et al. (2019). «Detección de baches en vías urbanas a partir de imágenes de alta resolución espacial, mediante técnicas de GEOBIA». En: *Revista de Topografía AZIMUT* 10.1.
- García-Ramírez, Yasmany et al. (2022). «Detection of potholes and their severity using the Video VBOX Lite and smartphones». En.
- Goenaga, Boris y Luis Fuentes (2020). «Efecto de los reductores de velocidad en la condición del pavimento». En: *Ingeniería de pavimentos.*, pág. 606.
- Güemes, Alfredo (2013). «SHM technologies and applications in aircraft structures». En: *Proceedings of the 5th International Symposium on NDT in Aerospace, Singapore*. Vol. 1315, pág. 4.
- Hou, Rongrong y Yong Xia (2021). «Review on the new development of vibration-based damage identification for civil engineering structures: 2010–2019». En: *Journal of Sound and Vibration* 491, pág. 115741.
- Izquierdo, Alberto et al. (2022). «Scalable and low-cost MEMS-based Structural Health Monitoring System». En: *CMMost 2021: 6th International Conference on Mechanical Models in Structural Engineering*. Universidad de Valladolid, págs. 2-16.
- Junior, Ronny Francis Ribeiro et al. (2022). «Fault detection and diagnosis in electric motors using 1d convolutional neural networks with multi-channel vibration signals». En: *Measurement* 190, pág. 110759.
- Mehtab, Sidra y Jaydip Sen (2022). «Analysis and forecasting of financial time series using CNN and LSTM-based deep learning models». En: *Advances in Distributed Computing and Machine Learning: Proceedings of ICADCML 2021*. Springer, págs. 405-423.
- Menegazzo, Jeferson y Aldo von Wangenheim (2022). «Speed Bump Detection Through Inertial Sensors and Deep Learning in a Multi-contextual Analysis». En: *SN Computer Science* 4.1, pág. 18.
- Moaveni, Babak et al. (2013). «Finite-element model updating for assessment of progressive damage in a 3-story infilled RC frame». En: *Journal of Structural Engineering* 139.10, págs. 1665-1674.
- Nikolaev, PN, A Espinoza Valles y MS Shcherbakov (2023). «Calibration of On-board Magnetometers of the SamSat-ION Nanosatellite». En: *2023 30th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS)*. IEEE, págs. 1-4.
- Peralta-López, José-Eleazar et al. (2023). «Speed bump and pothole detection using deep neural network with images captured through zed camera». En: *Applied Sciences* 13.14, pág. 8349.

- Pernia-Márquez, Daniel (2004). «Introducción a la medición de vibración». En: *Facultad de Ingeniería. Universidad de los Andes. Venezuela*.
- Quispe, Ricardo et al. (2022). «Assessment of the structural health monitoring in Edgardo Rebagliati Martins hospital with limited number of accelerometers in lima city». En: *Tecnia* 32.2, págs. 76-88.
- Salman, Amir y Adnan Noor Mian (2023). «Deep learning based speed bumps detection and characterization using smartphone sensors». En: *Pervasive and Mobile Computing* 92, pág. 101805.
- Samatas, Gerasimos G y Theodore P Pachidis (2022). «Inertial measurement units (imus) in mobile robots over the last five years: A review». En: *Designs* 6.1, pág. 17.
- Sanchez-Castro, Jonathan J et al. (2020). «A lean convolutional neural network for vehicle classification». En: *2020 IEEE 29th International Symposium on Industrial Electronics (ISIE)*. IEEE, págs. 1365-1369.
- Sattar, Shahram, Songnian Li y Michael Chapman (2018). «Road surface monitoring using smartphone sensors: A review». En: *Sensors* 18.11, pág. 3845.
- Shcherbachev, Andrew V et al. (2024). «Development of a Digital System for Dynamic Health Monitoring». En: *2024 Conference of Young Researchers in Electrical and Electronic Engineering (ElCon)*. IEEE, págs. 982-986.
- Sreevidya, PV et al. (2018). «Development of two axes magnetometer for navigation applications». En: *Journal of Magnetism and Magnetic Materials* 448, págs. 298-302.
- Stewart, James (2008). «Cálculo de varias variables». En: *Trascendentes tempranas* 6.
- Ugale, Hrishikesh et al. (2021). «IoT system for sensing condition of roads using IMU sensors». En: *2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC)*. IEEE, págs. 344-349.
- Varanis, Marcus et al. (2018). «MEMS accelerometers for mechanical vibrations analysis: A comprehensive review with applications». En: *Journal of the Brazilian Society of Mechanical Sciences and Engineering* 40, págs. 1-18.
- Wu, Tianqiong et al. (2023). «Research status and development trend of piezoelectric accelerometer». En: *Crystals* 13.9, pág. 1363.
- Zarei, Asghar, Hossein Beheshti y Babak Mohammadzadeh Asl (2022). «Detection of sleep apnea using deep neural networks and single-lead ECG signals». En: *Biomedical Signal Processing and Control* 71, pág. 103125.
- Zhao, He y Zheyao Wang (2011). «Motion measurement using inertial sensors, ultrasonic sensors, and magnetometers with extended kalman filter for data fusion». En: *IEEE Sensors Journal* 12.5, págs. 943-953.