Universidad de Valladolid

Escuela Técnica Superior de Ingenieros de Telecomunicación

Trabajo Fin de Grado

Grado en Ingeniería de Tecnologías de Telecomunicación

Evaluación de múltiples arquitecturas de detección de la pose humana mediante aprendizaje profundo en aplicaciones de análisis cinemático respecto al uso de sensores inerciales

Autor:

D. Mario Medrano Paredes

Tutor:

Dr. D. Mario Martínez Zarzuela

Valladolid, julio de 2025

Título:	**Evaluación de múltiples arquitecturas de detección de la pose humana mediante aprendizaje profundo en aplicaciones de análisis cinemático respecto al uso de sensores inerciales**
Autor:	**D. Mario Medrano Paredes**
Tutor:	**Dr. D. Mario Martínez Zarzuela**
Departamento:	**Teoría de la Señal, Comunicaciones e Ingeniería Telemática**

Tribunal

Presidente:	**D. Javier M. Aguiar Pérez**
Vocal:	**D. David González Ortega**
Secretario:	**D. Mario Martínez Zarzuela**
Suplente 1:	**Dña. Míriam Antón Rodríguez**
Suplente 2:	**Dña. M. Ángeles Pérez Juárez**

Fecha:	
Calificación:	

Agradecimientos

En primer lugar, me gustaría agradecer a mi tutor, Mario Martínez Zarzuela, y al resto de miembros del departamento, por su dedicación, sus conocimientos, y por haberme dado la oportunidad de formar parte de este proyecto.

A todos los participantes e investigadores que han ayudado a desarrollar la base de datos necesaria para la realización de este trabajo.

A todos mis amigos y compañeros por haberme acompañado a lo largo de este camino.

Por último, me gustaría dar gracias a mi familia, en especial a mis padres, por su apoyo incondicional, su comprensión y por enseñarme la importancia de la educación y el esfuerzo.

Resumen

El análisis cuantitativo del movimiento humano es una herramienta fundamental en el diagnóstico y la rehabilitación clínica, si bien los sistemas tradicionales de captura de movimiento presentan limitaciones en cuanto a su coste, complejidad de uso y accesibilidad. Este Trabajo de Fin de Grado aborda estas limitaciones desde una doble perspectiva, explorando el potencial del aprendizaje profundo no sólo para la captura cinemática, sino también para aplicaciones médicas, y, en específico, para la segmentación de la marcha mediante alternativas robustas y de bajo coste fuera de laboratorios especializados.

La primera parte del trabajo se centra en la evaluación de cuatro arquitecturas para la Estimación de la Pose Humana (HPE) en 3D a partir de vídeo monocular (MotionAGFormer, MotionBERT, VideoPose3D y NVIDIA BodyTrack). Utilizando el *dataset* VIDIMU, se calcularon los ángulos articulares de múltiples sujetos en diversas actividades cotidianas y se comparó el rendimiento de cada modelo basado en vídeo frente a los datos de referencia obtenidos de sensores inerciales (IMUs) mediante métricas de error y correlación estadística. Los resultados revelaron que, a pesar de que ningún modelo es universalmente superior, MotionAGFormer ofreció el menor RMSE (9.27° ± 4.80°) y la mayor correlación (0.67 ± 0.28) para el conjunto de datos evaluado. Esto verificó la viabilidad de los sistemas visión como alternativa al hardware dedicado para el análisis cinemático en entornos no controlados, destacando los compromisos entre precisión, tiempo de inferencia y eficiencia computacional.

La segunda parte se enfoca en el desarrollo de un modelo de *deep learning* para la segmentación de las fases de la marcha. Para ello, se estimaron las señales de velocidad angular a partir de los cuaterniones de los IMUs y se detectaron los eventos clave (*heel strike, toe off*). Posteriormente, se desarrolló una interfaz gráfica de usuario (GUI) para supervisar el etiquetado de los datos. Finalmente, se entrenó y optimizó una red BiLSTM para clasificar las fases de la marcha a partir de secuencias de ángulos articulares de la rodilla obtenidos mediante IMUs y modelos HPE. El modelo entrenado demostró una capacidad de segmentación prometedora, con F1-Score y accuracy superior a 80% para los datos de los sensores inerciales y superior al 70% para los de vídeo, validando la hipótesis de que es posible aprender patrones temporales complejos a partir de ciclos de marcha.

Palabras clave

Aprendizaje profundo, visión computacional, estimación de la pose humana, segmentación de la marcha, sensores inerciales, análisis cinemático.

Abstract

Quantitative human movement analysis is a fundamental tool in clinical diagnosis and rehabilitation, although traditional motion capture systems present limitations regarding cost, complexity, and accessibility. This thesis addresses these limitations from a dual perspective, exploring the potential of deep learning not only for kinematic capture but also for medical applications, specifically for gait segmentation using robust and low-cost alternatives outside specialized laboratories.

The first part of the work focus on evaluating four state-of-the-art architectures for 3D Human Pose Estimation (HPE) from monocular video (MotionAGFormer, MotionBERT, VideoPose3D, and NVIDIA BodyTrack). Using the VIDIMU dataset, the joint angles of multiple subjects across various daily activities were calculated, and the performance of each video-based model was compared against reference data from inertial measurement units (IMUs) using error and statistical correlation metrics. The results revealed that, although no single model was universally superior, MotionAGFormer offered the lowest RMSE (9.27° ± 4.80°) and the highest correlation (0.67 ± 0.28) for the evaluated dataset. This verified the viability of vision-based systems as an alternative to dedicated hardware for kinematic analysis in uncontrolled environments, highlighting the trade-offs between accuracy, inference time, and computational efficiency.

The second part consists of developing a deep learning model for gait phase segmentation. To this end, angular velocity signals were estimated from IMU quaternions to detect key gait events (heel strike, toe off). Subsequently, a graphical user interface (GUI) was developed to supervise the data labeling. Finally, a BiLSTM network was trained and optimized to classify gait phases from sequences of knee joint angles obtained using IMUs and HPE models. The trained model demonstrated promising segmentation capabilities, with an F1-Score and accuracy greater than 80% for inertial sensor data and greater than 70% for video data, validating the hypothesis that it is possible to learn complex temporal patterns from gait cycles.

Keywords

Deep learning, Computer vision, Human pose estimation, Gait segmentation, Inertial sensors, Kinematic analysis.

Índice general

Índice de figuras

Índice de tablas

1. Introducción

1.1. Contexto y Motivación

El análisis cuantitativo del movimiento y en especial la estimación de la pose humana (*Human Pose Estimation,* HPE), constituyen una herramienta de gran valor en diversas áreas de la salud, abarcando desde el diagnóstico clínico y el seguimiento de la progresión de enfermedades neuromusculares o neurodegenerativas, hasta la planificación y evaluación de programas rehabilitación y la optimización del rendimiento deportivo. En las últimas décadas, se ha experimentado un notable incremento en la demanda de soluciones de telemedicina y rehabilitación a distancia, impulsado por la necesidad de proporcionar atención sanitaria accesible, continua y personalizada fuera de los entornos clínicos tradicionales. Este cambio de paradigma exige el desarrollo de tecnologías innovadoras que permitan la monitorización y evaluación del movimiento de forma remota, fiable y con un bajo coste.

Históricamente, el *gold standard* para la captura del movimiento humano ha consistido en sistemas optoelectrónicos complejos como Vicon, que utiliza cámaras infrarrojas y marcadores pasivos reflectantes colocados sobre diversas partes del cuerpo. Si bien estos sistemas ofrecen una precisión submilimétrica, presentan desventajas significativas que limitan su uso generalizado. Su elevado coste, la necesidad de laboratorios especializados y personal técnico cualificado, y la naturaleza intrusiva de la colocación de marcadores, que puede alterar el patrón de movimiento natural del individuo, limitan su practicidad para seguimientos a largo plazo o en el domicilio del paciente.

Por otro lado, los sensores inerciales (*Inertial Movement Units,* IMUs) se han consolidado como otra de las tecnologías clave para el análisis del movimiento fuera del laboratorio. Estos dispositivos, que integran acelerómetros, giroscopios y en ocasiones magnetómetros, se adhieren al cuerpo para registrar la cinemática de los distintos segmentos corporales. Aunque presentan sus propios desafíos, como el *drift* en la estimación de la posición a largo plazo o la necesidad de una correcta calibración, los IMUs ofrecen datos cuantitativos fiables sobre la orientación y la velocidad angular. Por ello, no solo constituyen una herramienta valiosa para la monitorización ambulatoria, sino que también se emplean frecuentemente como sistemas de referencia o *ground truth* a la hora de validar la precisión de las tecnologías de HPE basadas en visión computacional (*computer vision*), como se realiza en la primera parte de este trabajo.

Una de las aplicaciones más relevantes del análisis del movimiento en el ámbito clínico es la segmentación del ciclo de la marcha en sus fases fundamentales (apoyo y balanceo), así como la identificación de los principales eventos temporales (contacto inicial del talón y despegue de los dedos). La caracterización precisa de estos parámetros espaciotemporales es crucial para el diagnóstico y seguimiento de una amplia gama de patologías que afectan a la locomoción, como por ejemplo la enfermedad de Parkinson, el ictus o la parálisis cerebral, así como para evaluar la eficacia de intervenciones terapéuticas y programas de rehabilitación. Tradicionalmente, la

segmentación se ha realizado mediante la inspección visual por parte de profesionales clínicos o empleando sistemas optométricos combinados con plataformas de fuerza, pero existe un creciente interés en automatizar este proceso mediante sistemas más sencillos, ámbito en el que se centrará la segunda parte de este estudio.

El crecimiento exponencial del aprendizaje profundo (*Deep Learning*) ha abierto una vía novedosa para superar las barreras inherentes a los métodos clásicos expuestos anteriormente mediante dos aproximaciones complementarias. En primer lugar, los modelos de visión artificial permiten localizar las articulaciones y reconstruir con gran precisión la pose humana en 3D a partir de vídeo monocular, prescindiendo de hardware específico o marcadores, y reduciendo todo el proceso a una simple grabación con una cámara convencional. En segundo lugar, las redes neuronales clásicas y sus variantes modernas permiten segmentar las fases de la marcha e incluso detectar patrones patológicos o anomalías específicas. Esta aproximación ofrece una alternativa prometedora y accesible en entornos menos controlados, abriendo la puerta a escenarios de rehabilitación y análisis cinemático automatizados y adaptados a cada paciente.

Este Trabajo de Fin de Grado se enmarca en este contexto con un doble propósito. Por un lado, se aborda la necesidad de evaluar la precisión de diversas arquitecturas de HPE 3D, utilizando los datos de IMUs como referencia, para determinar su idoneidad en sistemas que requieran del cálculo de ángulos articulares. Cabe destacar que, durante la realización del trabajo, se ha escrito y enviado un artículo científico sobre este tópico a la revista *Artificial Intelligence Review* de Springer, y actualmente se encuentra en revisión (Medrano-Paredes et al., 2025). Por otro lado se explora una aplicación directa de estos análisis cinemáticos mediante el desarrollo de una red BiLSTM (*Bidirectional Long Short Term Memory*) para la segmentación de las fases de la marcha humana. La motivación subyacente a este proyecto es por tanto la democratización del análisis biomecánico, acercándolo al domicilio y mejorando las capacidades de diagnóstico, seguimiento y rehabilitación y, por extensión, la calidad de vida de los pacientes.

1.2. Estado del arte

A pesar de los avances significativos en los últimos años, sigue existiendo una notable brecha en la literatura en cuanto a la existencia de una comparación directa y sistemática entre los métodos de estimación de la pose basados en aprendizaje profundo para la medición de ángulos articulares fuera del laboratorio. Conjuntos de datos de pose humana como Human3.6M (Ionescu et al., 2014), MPI-INF-3DHP (Mehta et al., 2017), Microsoft COCO y 3DPW siguen siendo fundamentales para evaluar los métodos de HPE basados en vídeo. Estos *datasets* ilustran diversos escenarios, incluyendo oclusiones, interacciones entre múltiples personas y distintos planos de grabación. Los primeros trabajos, como DeepPose (Toshev & Szegedy, 2014) o DeeperCut (Insafutdinov et al., 2016), fueron pioneros en la estimación de la pose humana basada en vídeo utilizando redes neuronales profundas y convolucionales. Si bien efectivos, estos métodos a menudo omitían factores como las deficiencias en la consistencia temporal, la ambigüedad de la profundidad, o los errores por ocultamientos.

Revisiones exhaustivas, como las de (Andriluka et al., 2018; Ben Gamra & Akhloufi, 2021; Y. Chen et al., 2020; Pang et al., 2022; Wang et al., 2021; C. E. Zheng et al., 2018), analizan sistemáticamente múltiples métodos y arquitecturas para HPE tanto 2D como 3D, enfatizando desafíos como la oclusión y la incertidumbre en datos de vídeo monocular. Sin embargo, los *benchmarks* existentes a menudo priorizan entornos de laboratorio, descuidando los desafíos del mundo real como la iluminación dinámica y los ángulos de cámara variables, inherentes a la rehabilitación en entornos del día a día. Además, solo unos pocos abordan los compromisos en precisión, tiempos de inferencia y simplicidad de uso entre modelos, así como las aplicaciones específicas según el tipo de arquitectura (K. Chen et al., 2018; Ienaga et al., 2022). En cuanto a los sistemas de *ground truth*, la mayoría de las revisiones existentes (Lee et al., 2024; Vafadar et al., 2022) comparan las arquitecturas de visión por computador con configuraciones basadas en marcadores optoelectrónicos múltiples con varias cámaras infrarrojas como Vicon u OptiTrack, que no cumplen con los requisitos de accesibilidad y presupuesto para la rehabilitación en el hogar.

En los últimos años han surgido varios métodos nuevos de HPE en respuesta a la rápida aparición de nuevos desarrollos en el campo del aprendizaje automático, siendo los Transformers los más populares. Por ejemplo, ViTPose (Xu et al., 2022) introdujo una aproximación simplificada que aprovecha los Transformers de visión no jerárquicos para la estimación de la pose sin depender de extracciones de características adicionales basadas en CNN. El diseño del modelo enfatiza la escalabilidad y la flexibilidad de entrenamiento. Además, (C. Zheng et al., 2021) propone un modelo totalmente basado en Transformers que estima directamente las poses humanas 3D a partir de secuencias de poses 2D, lo que lleva a un rendimiento del estado del arte en Human3.6M y MPI-INF-3DHP. Por otro lado, (Jiang et al., 2023) presenta ZeDO, un pipeline de optimización basado en difusión de tipo *zero-shot* que refina iterativamente las poses minimizando los errores de reproyección 2D utilizando un modelo de difusión preentrenado. A diferencia de los métodos convencionales basados en aprendizaje que tienen dificultades con los cambios de

dominio al promediar poses, ZeDO estima cada pose caso por caso sin depender de datos de entrenamiento 2D-3D emparejados.

En paralelo, el análisis y la segmentación de la marcha han sido también objeto de múltiples investigaciones. (Grimmer et al., 2019) estudia el uso de la velocidad angular de varios segmentos de la extremidad inferior (biológicos y virtuales) para la detección de las fases de apoyo y balanceo, concluyendo que, si bien el concepto es viable, se requieren conjuntos de reglas adicionales para una detección fiable con datos de IMUs. Por su parte, (Salminen et al., 2024) presenta un método basado en la velocidad angular acumulada mediolateral de la tibia (CSAV), medida con sensores inerciales, para dividir el ciclo de la marcha en siete fases mediante la detección precisa de eventos como el despegue del talón (HR), pies adyacentes (FA) y tibia vertical (TBV), mostrando una robustez superior a métodos existentes para la detección del despegue de los dedos (TO) bajo diversas condiciones de velocidad y calzado. En cuanto al uso de otras tecnologías, (Khokhlova et al., 2019) utilizaron datos de orientación articular del sensor Kinect v.2 para calcular parámetros cinemáticos y desarrollaron una red neuronal para la clasificación no supervisada de marcha normal y patológica (cojera y rigidez de rodilla), demostrando la viabilidad de sensores de profundidad para este tipo de análisis.

La investigación en segmentación de la marcha también ha explorado diferentes tipos de herramientas más allá de las metodologías clásicas. Las redes LSTM, por su capacidad para modelar dependencias temporales en datos secuenciales, se han convertido en la herramienta preferida para esta tarea. A este respecto, (Shi et al., 2023) desarrollaron un sistema de análisis de la marcha portátil basado en sensores inerciales montados en el pie, utilizando una red LSTM para segmentar cuatro fases típicas sin necesidad de umbrales, alcanzando precisiones superiores al 95%. De manera similar, (Su & Gutierrez-Farewik, 2020) propone un enfoque basado en el mismo tipo de red recurrente para predecir tanto las trayectorias de los segmentos de las extremidades inferiores (velocidad angular de muslo, tibia y pie) como cinco fases de la marcha (respuesta a la carga, apoyo medio, apoyo terminal, pre-balanceo y balanceo) a partir de datos de IMUs, demostrando una correlación superior a 0.98 en la predicción de trayectorias y una notable precisión en la predicción de fases, especialmente la de balanceo.

1.3. Hipótesis

La hipótesis que se plantea en este Trabajo de Fin de Grado sostiene que los avances recientes en aprendizaje profundo permiten, por un lado, obtener una estimación tridimensional de la pose humana a partir de vídeo monocular de forma accesible (sin necesidad de sistemas ópticos profesionales ni sensores corporales), alcanzando una precisión suficiente para equipararse a los IMUs en el cálculo de parámetros cinemáticos clínicamente relevantes. Por otro lado, se propone que es posible utilizar las señales de velocidad angular derivadas de dichos sensores inerciales para etiquetar de forma consistente los eventos de la marcha. De esta manera, una vez se dispone de ángulos articulares estimados a partir de vídeo y correctamente sincronizados, es posible entrenar

un modelo de aprendizaje profundo capaz de segmentar automáticamente los ciclos de marcha y clasificar cada instante en las fases entre eventos de marcha únicamente a partir de esos ángulos.

De comprobarse ambas premisas, se demostraría la viabilidad de un flujo completo, robusto y de bajo coste que combina visión artificial para la captura de datos y redes neuronales recurrentes para la interpretación de la señal, ofreciendo una alternativa no invasiva y accesible a los sistemas clásicos de captura de movimiento para el análisis cinemático a distancia.

1.4. Objetivos

Este trabajo tiene como objetivo general evaluar la viabilidad y precisión de las arquitecturas de visión artificial para la estimación de la pose humana en tres dimensiones a partir de vídeo monocular, comparando su rendimiento con el de los sensores inerciales mediante la medida de los ángulos articulares, así como desarrollar un modelo de aprendizaje profundo para segmentar de forma automática las fases de la marcha humana en base a dichos datos angulares. Para lograrlo, se proponen los siguientes objetivos específicos:

i) Realizar una revisión rigurosa de la bibliografía acerca de la estimación de pose, la segmentación de la marcha, y técnicas de aprendizaje profundo aplicadas a la biomecánica y al análisis cinemático.
ii) Seleccionar e implementar diversas arquitecturas de visión computacional para estimación de pose humana 3D y comparar su precisión frente a los ángulos articulares calculados mediante sensores inerciales, utilizando para ello un flujo de procesado (filtrado, suavizado, sincronización, etc.) y el cálculo de métricas descriptivas para el conjunto de datos disponible.
iii) Discutir las fortalezas, limitaciones e implicaciones de cada arquitectura evaluada, así como los casos de uso más recomendables para cada una de ellas.
iv) Implementar un flujo de procesado para calcular señales de velocidad angular a partir de los cuaterniones extraídos mediante los sensores inerciales. Realizar un filtrado inicial, estandarizar la señal corrigiendo los signos y detectar los eventos de la marcha.
v) Desarrollar una interfaz gráfica que permita revisar y corregir de manera interactiva las etiquetas de fase de marcha, garantizando un conjunto de entrenamiento libre de errores.
vi) Diseñar, entrenar y validar un modelo de aprendizaje profundo que clasifique cada instante del ciclo empleando ángulos articulares como datos de entrada, incorporando para ello técnicas de regularización y validación cruzada.
vii) Evaluar cuantitativamente el rendimiento del sistema completo mediante métricas estadísticas, matrices de confusión y curvas de aprendizaje.
viii) Analizar los resultados obtenidos, identificar las limitaciones del enfoque propuesto y proponer líneas de mejora y trabajos futuros que consoliden la viabilidad de soluciones accesibles para análisis cinemático humano.

1.5. Estructura del documento

Este manuscrito se organiza en cinco capítulos que siguen la lógica del proceso de investigación llevado a cabo. El Capítulo 1 introduce el Trabajo de Fin de Grado, presentando su contexto y motivación, así como el estado del arte, la hipótesis de partida y los objetivos perseguidos.

El Capítulo 2 expone el marco teórico, revisando los fundamentos científicos necesarios y abordando los principios conceptuales del aprendizaje profundo, la estimación de pose humana, y las bases biomecánicas y clínicas de la segmentación de la marcha.

El Capítulo 3 describe la evaluación de arquitecturas de estimación de pose, explicando los materiales y base de datos empleados, la metodología seguida para el cálculo y la comparación de ángulos articulares respecto a los sensores inerciales. Incluye asimismo los resultados cuantitativos del estudio y su discusión.

El Capítulo 4 detalla el desarrollo de un flujo para la segmentación de la marcha, junto con el preprocesado de señales, el algoritmo y la interfaz gráfica de corrección para el etiquetado de fases. Además, se profundiza en el diseño, entrenamiento y validación del modelo BiLSTM y los resultados obtenidos en la clasificación automática de las fases de la marcha a partir de ángulos articulares.

El capítulo 0 integra la consecución de los objetivos planteados, extrae las conclusiones derivadas de las dos partes del estudio, analiza las limitaciones de los sistemas desarrollados y propone posibles mejoras y líneas de investigación futuras para desarrollar soluciones efectivas de rehabilitación telemática.

Al final del documento se incluye la lista de referencias bibliográficas que sustentan los contenidos citados a lo largo de todo el texto.

2. Marco teórico

2.1. Deep Learning

2.1.1. Introducción

Durante las últimas décadas, el aprendizaje profundo (*deep learning*) se ha consolidado como la corriente dominante dentro del aprendizaje automático (*machine learning*), fenómeno que ha sido impulsado por tres factores convergentes: la disponibilidad de grandes conjuntos de datos de entrenamiento, el aumento exponencial de la capacidad de cómputo (especialmente mediante unidades de procesamiento gráfico, GPUs) y la optimización de arquitecturas neuronales capaces de aprender representaciones jerárquicas cada vez más complejas. El resultado ha sido un salto cualitativo en el rendimiento de tareas tradicionalmente complejas para los sistemas algorítmicos, como el reconocimiento y la síntesis de voz, la programación asistida por inteligencia artificial, la visión computacional y, de forma especialmente relevante para este Trabajo de Fin de Grado, la estimación de la pose humana y la segmentación de datos médicos.

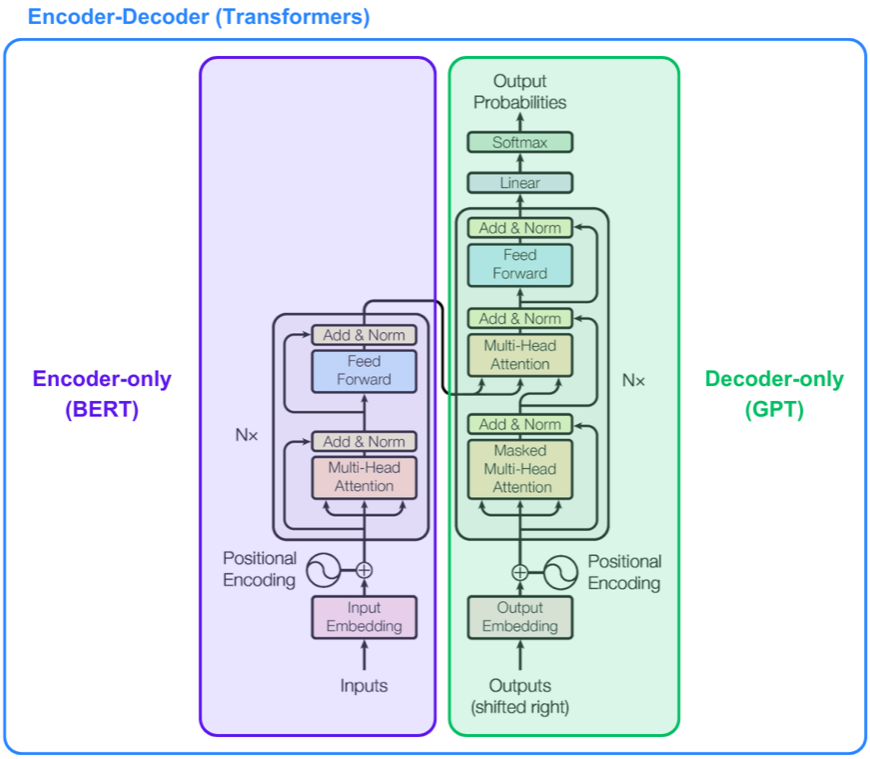


Figura 2.1. Arquitectura de Transformers vs BERT vs GPT. Diseñada a partir de (Vaswani et al., 2023).

El origen del aprendizaje profundo se remonta al perceptrón ideado por Rosenblatt a finales de los años cincuenta (Rosenblatt, 1958), seguido del neocognitrón de Kunihiko Fukushima en 1980 (Fukushima, 1980). No obstante, el hito fundacional suele situarse en 2006, cuando Hinton

y Hopfield reactivaron el campo con la propuesta de los *autoencoders* apilados y el pre-entrenamiento no supervisado, lo que posteriormente les llevaría a ser galardonados en 2024 con el premio Nobel de Física por sus contribuciones fundamentales al desarrollo de redes neuronales artificiales (*Press Release*, s. f.). Sin embargo, la verdadera eclosión llegó en 2012 con AlexNet (Krizhevsky et al., 2012), que demostró la superioridad de las Redes Neuronales Convolucionales (CNNs) en el ILSVRC. Desde entonces, la investigación ha avanzado hacia arquitecturas cada vez más profundas y especializadas (ResNet, EfficientNet), al aprendizaje autoregresivo de secuencias mediante Redes Recurrentes (RNNs, LSTMs) y, más recientemente, hacia los *Graph Neural Networks* (GNNs) (Scarselli et al., 2009) y la arquitectura de *Transformers* basada en mecanismos de atención (Vaswani et al., 2023). Todo ello ha dado origen a modelos cuyo uso se ha generalizado de forma acelerada en la actualidad, como por ejemplo los *Generative Pre-trained Transformers* (GPTs) (Radford et al., s. f.) y los *Bidirectional Encoder Representations from Transformers* (BERT) (Devlin et al., 2019).

En el aprendizaje automático clásico la eficacia del modelo dependía en gran medida de la calidad de las características diseñadas manualmente por expertos del dominio. Estas características se alimentaban después a clasificadores o regresores relativamente simples (p. ej., máquinas de soporte vectorial o árboles de decisión). En contraste, el aprendizaje profundo automatiza la extracción de representaciones a múltiples escalas mediante redes neuronales artificiales con numerosos niveles de abstracción. Cada capa aprende transformaciones no lineales de los datos de entrada, de forma que las capas más cercanas a la salida capturan conceptos de alto nivel adecuados para la tarea objetivo. Este paradigma *end-to-end* reduce la necesidad de ingeniería manual, pero introduce nuevos retos vinculados al tamaño de los modelos, la interpretabilidad y la demanda de datos y recursos computacionales.

Por otro lado, la visión artificial, también denominada visión por computador (*computer vision*), es un campo interdisciplinario que persigue dotar a los ordenadores de la capacidad de adquirir, procesar y comprender información visual de manera análoga a la percepción humana y en algunos aspectos, superior. Sus fundamentos combinan óptica y electrónica para la captura de imágenes, álgebra lineal y teoría de la señal para el preprocesado (filtrado, corrección geométrica, normalización de color), y estadística, geometría proyectiva y aprendizaje automático para transformar los píxeles en representaciones semánticas como detección de bordes, extracción de características, segmentación, reconocimiento de objetos o estimación de la profundidad.

En la actualidad, el desarrollo del *deep learning* ha sustituido las cadenas de procesamiento manual por arquitecturas neuronales que aprenden de extremo a extremo, ampliando el espacio de aplicaciones potenciales, especialmente en el ámbito de la salud. La estimación de la pose humana para tareas de rehabilitación telemática es un ejemplo de ello, y será el principal foco en la Parte I de este trabajo. Una descripción detallada de las arquitecturas y modelos de visión empleados se presenta en la Sección 2.2.3.

2.1.2. Tipos y arquitecturas de aprendizaje profundo

El término arquitectura hace referencia al patrón interno de conexiones, flujo de datos y funciones de activación que conforman una red neuronal. El aprendizaje profundo abarca una amplia gama de arquitecturas neuronales, cada una con fortalezas particulares para diferentes tipos de datos y tareas. Antes de describir las familias de redes más empleadas en la actualidad, conviene situarlas en el contexto de los paradigmas de aprendizaje:

- **Aprendizaje supervisado (*Supervised Learning*).** El modelo aprende a partir de datos etiquetados, donde cada instancia de entrada está asociada con una salida o etiqueta conocida. El objetivo es aprender una función que mapee las entradas a las salidas para poder predecir la salida de nuevas entradas no vistas. Algunos ejemplos típicos son las tareas de clasificación y regresión.
- **Aprendizaje no supervisado (*Unsupervised Learning*).** En este caso, el modelo recibe datos sin etiquetar y debe encontrar patrones, estructuras o representaciones inherentes en los datos por sí mismo.
- **Aprendizaje por refuerzo (*Reinforcement Learning*).** Este paradigma se basa en agentes que aprenden a tomar secuencias de decisiones en un entorno para maximizar una recompensa acumulada. El agente aprende a través de la interacción con el entorno, recibiendo retroalimentación en forma de premios o penalizaciones por sus acciones.
- **Aprendizaje semi-supervisado (*Semi-Supervised Learning*).** Combina una pequeña cantidad de datos etiquetados con una gran cantidad de datos no etiquetados.
- **Aprendizaje auto-supervisado (*Self-Supervised Learning*).** Es una forma de aprendizaje no supervisado donde las etiquetas se generan automáticamente a partir de los propios datos de entrada.

Convolutional Neural Networks (CNNs)

Las Redes Neuronales Convolucionales surgieron para explotar la correlación espacial local presente en datos con estructura de rejilla (imágenes, audio espectrogramas, etc.). Su bloque fundamental es la convolución discreta o capa convolucional entre el tensor de entrada y un conjunto de filtros o *kernels*. Cada filtro comparte pesos a lo largo de toda la imagen, proporcionando invariancia traslacional y reduciendo drásticamente el número de parámetros frente a una capa *fully connected* tradicional. Tras varias capas, las salidas se suelen combinar mediante funciones de activación no lineales (ReLU, GELU) y submuestreo o agrupación (*max-pooling*, *stride*), extrayendo progresivamente características de mayor abstracción. Al final, suelen ubicarse una o más capas completamente conectadas que toman los mapas de características de alto nivel y realizan la clasificación final o la regresión.

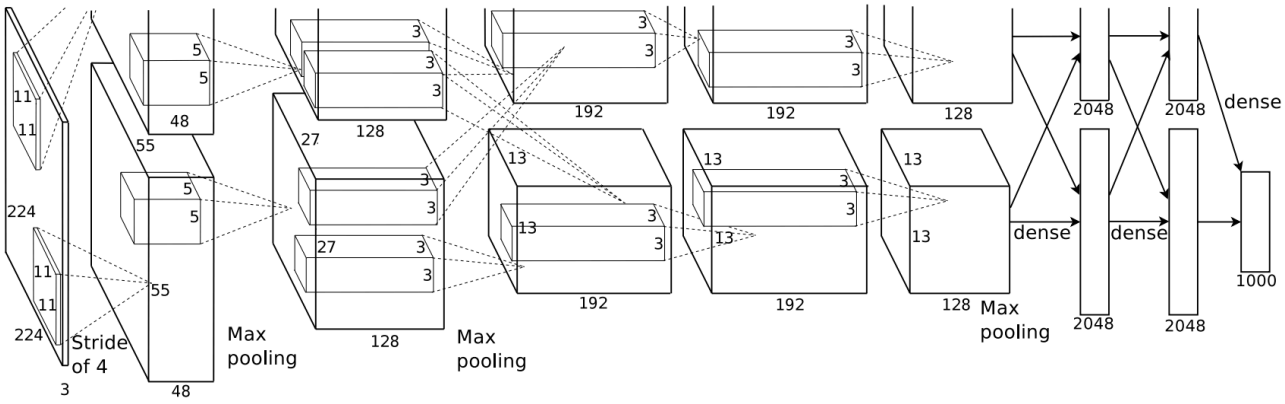


Figura 2.2. Arquitectura de la CNN AlexNet (Krizhevsky et al., 2012).

Recurrent Neural Networks **(RNNs)**

Cuando los datos presentan dependencia temporal, resulta más natural un modelo que mantenga un estado a lo largo de la serie. Las Redes Neuronales Recurrentes introducen un bucle interno que, en cada instante t, recibe la entrada x_t y el estado oculto anterior h_{t-1} para producir un nuevo estado $h_t = f(W_{hh}h_{t-1} + W_{xh}x_t + b_h)$, donde W son matrices de pesos, b es un sesgo y f es una función de activación. Esto les permite mantener una "memoria" o estado interno que captura información de los elementos procesados previamente en la secuencia.

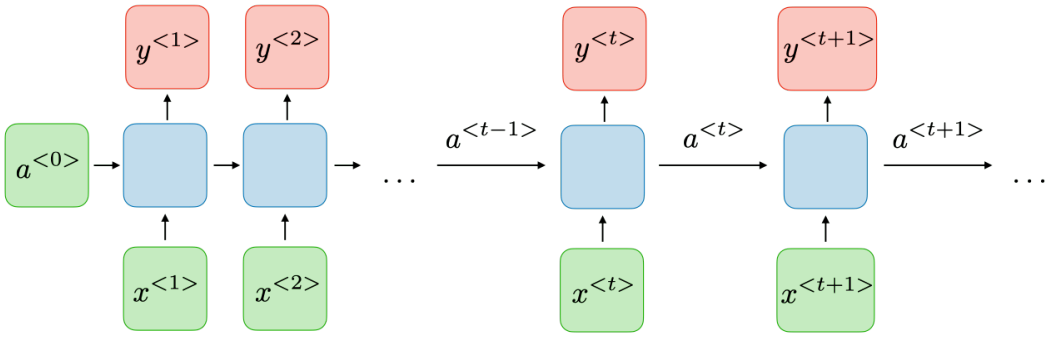


*Figura 2.3. Arquitectura de una RNN tradicional (*CS 230 - Recurrent Neural Networks Cheatsheet*, s. f.).*

Teóricamente, esta recurrencia puede modelar dependencias arbitrariamente largas. No obstante, en la práctica, la propagación del gradiente a través de muchos pasos da lugar al problema del gradiente desvaneciente (*vanishing gradient*) o explosivo (*exploding gradient*). Durante la retropropagación a través del tiempo, los gradientes pueden multiplicarse por los mismos pesos recurrentes en cada paso temporal. Si estos pesos son pequeños, los gradientes pueden disminuir exponencialmente, impidiendo que la red aprenda relaciones distantes. Si los pesos son grandes, los gradientes pueden crecer exponencialmente, llevando a actualizaciones inestables. Para solventarlo se propusieron arquitecturas más sofisticadas, siendo algunas de las más conocidas:

- ***Long Short-Term Memory* (LSTM)**. Introducen una estructura de celda más compleja con tres puertas (*gates*). Las puertas de olvido deciden que información del estado de la celda anterior debe descartarse. Las de entrada deciden que nueva información se almacenará, y las de salida deciden que parte del estado de la celda actual se utilizará para generar el estado oculto y, por tanto, la salida.
- ***Bidirectional LSTM* (BiLSTM)**. Una BiLSTM consiste en dos LSTMs que procesan la secuencia de entrada en direcciones opuestas: una de adelante hacia atrás (pasado a futuro) y otra de atrás hacia adelante (futuro a pasado). Las salidas de los estados ocultos de ambas LSTMs en cada instante de tiempo se concatenan para formar la representación final en ese instante. Esto permite que la predicción tenga acceso tanto al contexto pasado como al futuro de la secuencia. Al disponer de ambos contextos en cada paso, es especialmente útil para clasificación de series temporales estacionarias, como la delimitación de eventos de marcha humana, donde conocer la dinámica inmediatamente posterior mejora la detección de transición entre fases.

Mecanismos de atención y *Transformers*

Los mecanismos de atención permiten a un modelo ponderar dinámicamente la importancia de diferentes partes de la secuencia de entrada (o de estados ocultos anteriores) al generar una salida o tomar una decisión. En lugar de comprimir toda la información de la secuencia en un único vector de contexto fijo (como en las arquitecturas RNN más simples *encoder-decoder*), las arquitecturas de atención calculan pesos que indican qué partes de la entrada son más relevantes para la tarea actual. Esto ha demostrado ser muy eficaz para manejar secuencias largas y capturar dependencias complejas.

Los *transformers* llevan este principio un paso más allá, eliminando por completo la recurrencia y basando todo el cómputo en autoatención (*self-attention)* multicabeza, que permite a cada elemento de la secuencia atender a todos los demás elementos para calcular su representación y calcula en paralelo la dependencia entre todos los pares de posiciones de la secuencia. Con la ayuda del *positional encoding* para introducir noción de orden, los *transformers* predominan en aplicaciones de Procesamiento del Lenguaje Natural (NLP) y se están trasladando a visión por computador y problemas de series temporales.

Aunque las arquitecturas basadas en *transformers* son muy potentes y son útiles en la detección de pose de la Parte I de este trabajo, para tareas de segmentación de series temporales con secuencias de longitud moderada (como el análisis de la marcha a partir de ángulos articulares), las LSTMs siguen siendo arquitecturas muy competitivas y eficientes, motivo por el que se ha optado por emplearlas en la Parte II.

2.1.3. Técnicas de entrenamiento y evaluación

El rendimiento de cualquier modelo de aprendizaje automático depende tanto de su arquitectura como del proceso de entrenamiento y evaluación. En esta subsección se describen, de manera sistemática, las técnicas y conceptos que constituyen su ciclo completo de desarrollo: preprocesado de datos, optimización, regularización, validación, ajuste de hiperparámetros y cálculo de métricas de evaluación.

Preprocesado y normalización de datos de entrada

Antes de iniciar el entrenamiento conviene escalar todas las variables de entrada mediante el uso del *Standard Scaler* para que presenten una media cercana a 0 y una desviación estándar unitaria (*z-score*). Esto se logra restando la media y dividiendo por la desviación estándar de cada característica, calculadas a partir del conjunto de entrenamiento. Esta técnica evita que las neuronas asociadas a características de gran magnitud dominen el gradiente y acelera la convergencia de algoritmos como Adam o SGD. Todo esto ayuda a que los algoritmos de descenso de gradiente converjan más rápidamente y de manera más estable, ya que evita que las características con rangos de valores más grandes dominen la actualización de los pesos.

Algoritmos de optimización

La optimización constituye el núcleo del proceso de entrenamiento. El aprendizaje de los pesos, sesgos o parámetros del modelo se formaliza como un problema de minimización de la función de pérdida mediante descenso por gradiente u otros métodos. Algunos de los algoritmos de optimización más comunes son SGD (*Stochastic Gradient Descent*), Adam, RMSprop, o Adagrad.

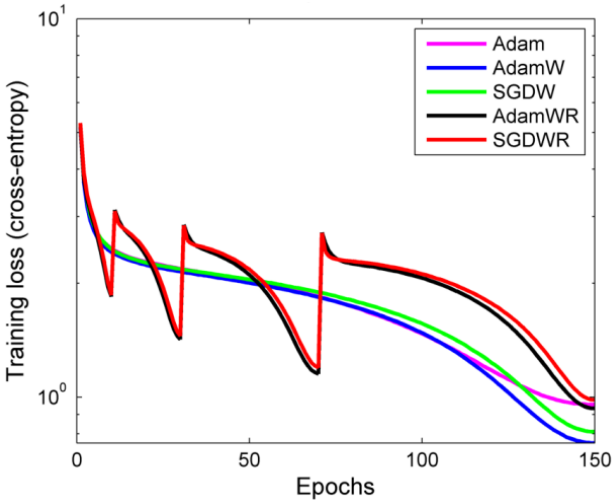


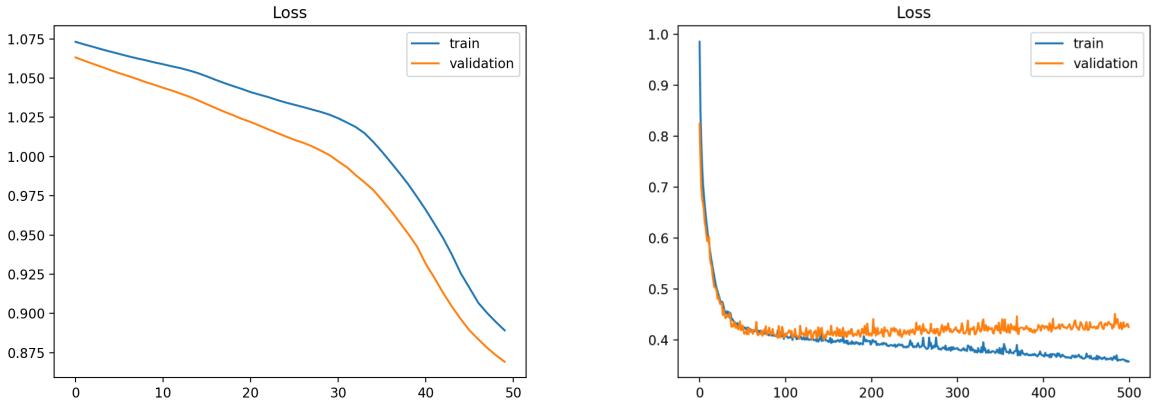
Figura 2.4. Curvas de pérdida durante el entrenamiento para distintos optimizadores (Loshchilov & Hutter, 2019).

En este proyecto haremos uso del optimizador AdamW (*Adaptative Moment Estimation with Weight Decay*) (Loshchilov & Hutter, 2019), una variante de Adam que desacopla la regularización por decaimiento o penalización de pesos L2 tras la actualización del gradiente. Su regla de actualización combina adaptabilidad de pasos (RMSProp) con momentos de primer orden, mientras aplica *weight decay* de forma explícita para controlar la complejidad del modelo, lo que ha demostrado mejorar la generalización en la mayoría de los casos.

Overfitting vs underfitting

Antes de introducir las técnicas de regularización utilizadas, conviene definir dos de los problemas más habituales a la hora de entrenar y modelar redes neuronales.

- **El *overfitting* (sobreajuste)** aparece cuando el modelo captura demasiado bien los datos y el ruido idiosincrático del conjunto de entrenamiento, fallando al generalizar con información nunca antes vista. Se observa cuando el error en el conjunto de entrenamiento es bajo, pero el error en el conjunto de validación es significativamente más alto. Para mitigarlo se pueden aplicar diversos métodos de regularización.
- **El *underfitting* (subajuste)** ocurre cuando la red es demasiado simple o está insuficientemente entrenada, por lo que falla al capturar los patrones subyacentes en los datos. Se manifiesta por un alto error tanto en el conjunto de entrenamiento como en el de validación, y para corregirlo se deben ajustar los hiperparámetros o aumentar la capacidad del modelo.



(a) *(b)*

Figura 2.5. (a) Funciones de pérdida de un modelo sobreajustado. (b) Funciones de pérdida de un modelo subajustado. (Brownlee, 2019).

Regularización

La regularización comprende un conjunto de técnicas diseñadas para prevenir el fenómeno de sobreajuste explicado previamente. Algunas de las técnicas más conocidas, y que se han implementado en la Parte II de este trabajo, son las siguientes:

- ***Weight Decay*** **(decaimiento de pesos o regularización L2)**. Añade un término de penalización a la función de pérdida que es proporcional a la suma de los cuadrados de los pesos del modelo. Al minimizar esta función de pérdida modificada, el optimizador tiende a mantener los pesos del modelo pequeños, lo que promueve modelos más simples y menos propensos al sobreajuste. Un modelo con pesos grandes es más sensible a pequeñas variaciones en la entrada, lo que puede indicar memorización del ruido.
- ***Dropout*****.** Durante el entrenamiento, se "apaga" aleatoriamente (se pone a cero) un subconjunto de neuronas en una capa con una probabilidad determinada en cada iteración. Esto fuerza a la red a aprender representaciones más robustas y distribuidas, ya que no puede depender excesivamente de ninguna neurona o conjunto pequeño de neuronas. Es similar a entrenar múltiples redes neuronales más pequeñas de forma implícita y luego promediar sus predicciones en tiempo de inferencia. No obstante, en la práctica (en inferencia) se escalan los pesos de las neuronas activas por la probabilidad de no ser descartadas durante el entrenamiento.
- ***Early Stopping*** **(parada temprana).** Esta técnica monitoriza el rendimiento del modelo en un conjunto de validación separado durante el entrenamiento. El entrenamiento se detiene cuando el rendimiento en el conjunto de validación comienza a empeorar para una determinada métrica durante un número predefinido de épocas consecutivas (parámetro de "paciencia)", incluso si la pérdida en el conjunto de entrenamiento sigue disminuyendo. Esto evita que el modelo continúe entrenándose hasta el punto de sobreajustar los datos de entrenamiento.

Función de pérdida

La función de pérdida (*loss function*) cuantifica la discrepancia entre las predicciones del modelo y los valores verdaderos para un determinado problema. La elección de la función de pérdida depende de la naturaleza de la tarea llevada a cabo. Algunas de las más comunes son las siguientes:

- ***Cross-Entropy Loss*** **(entropía cruzada).** Es comúnmente utilizada para tareas de clasificación. Mide la diferencia entre dos distribuciones de probabilidad: la distribución verdadera de las etiquetas y la distribución de las predicciones del modelo. Para problemas de clasificación multiclase, se utiliza típicamente la entropía cruzada categórica.
- ***Weighted Loss*** **(pérdida ponderada).** En situaciones con clases desbalanceadas (donde algunas clases tienen mucha más representación que otras en el conjunto de entrenamiento),

una función de pérdida estándar puede llevar a que el modelo presente sesgo hacia la clase mayoritaria, de modo que en el entrenamiento se minimiza la pérdida porque se etiquetan la mayoría de muestras con la etiqueta de la clase más frecuente, pero el rendimiento no mejora. La pérdida ponderada asigna pesos diferentes a los errores cometidos en cada clase, penalizando más los errores en las clases minoritarias. Esto ayuda a que el modelo preste más atención a estas clases menos frecuentes y mejore su rendimiento global, combatiendo el problema del desbalanceo de clases. Como se explica en la Parte II, esta función fue empleada para solventar la sobrerrepresentación de la clase *Turn* durante la segmentación de la marcha.

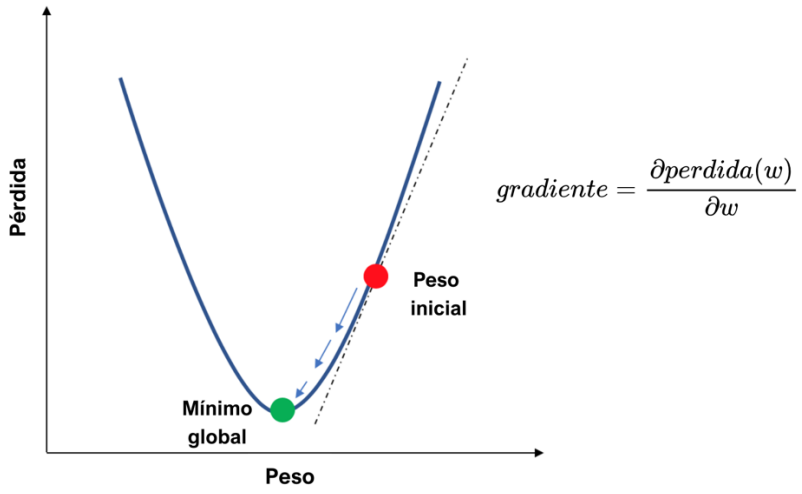


Figura 2.6. Función de pérdida ponderada.

Estrategias de validación

Para evaluar objetivamente el rendimiento de un modelo y tomar decisiones informadas sobre su arquitectura e hiperparámetros, es fundamental separar los datos disponibles en conjuntos de entrenamiento, validación y prueba. Un *split* simple (entrenamiento/validación/prueba) puede inducir varianza si el conjunto es limitado o los sujetos presentan gran heterogeneidad. La validación cruzada (*cross validation*) K-Fold es una técnica robusta para estimar el rendimiento del modelo. El conjunto de entrenamiento se divide en k subconjuntos (*folds*). El modelo se entrena k veces, utilizando en cada iteración un *fold* diferente como conjunto de validación y los $k-1$ subconjuntos restantes como conjunto de entrenamiento. Las métricas de rendimiento se promedian sobre las k iteraciones para obtener una estimación más estable del rendimiento del modelo. Este proceso se ilustra de manera visual en la Figura 2.7.

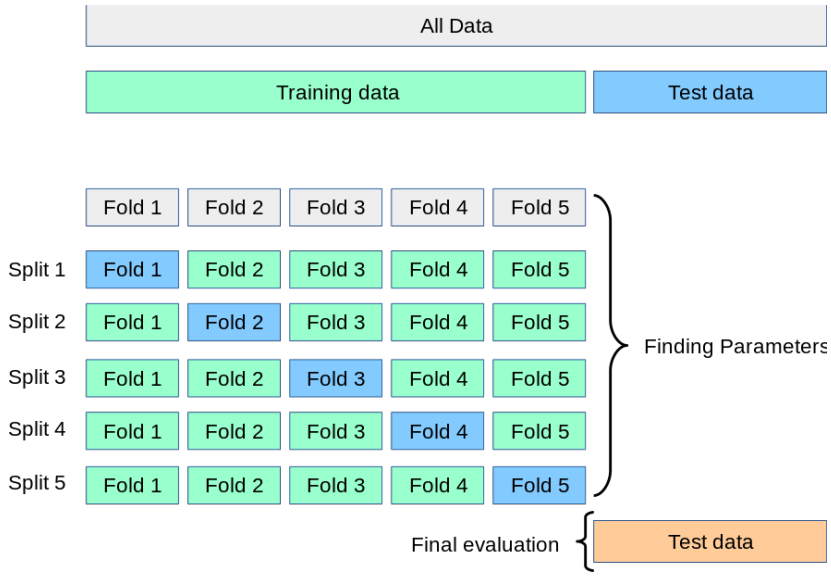


*Figura 2.7. Diagrama de funcionamiento de la validación cruzada (*3.1. Cross-Validation*, s. f.).*

Cabe mencionar que en este trabajo se hizo uso de la técnica GroupKFold para asegurar que todas las muestras de un mismo sujeto estén en el mismo conjunto dentro de un *fold*. Esto es crucial en datos biomédicos para evitar la fuga de datos y obtener una estimación más realista de la generalización del modelo a sujetos no vistos

Ajuste de hiperparámetros

Los hiperparámetros son parámetros que no se aprenden directamente durante el entrenamiento, sino que son fijados previamente al diseñar la red (p. ej. tasa de aprendizaje, número de capas, número de neuronas por capa, o coeficiente de *dropout*). Su optimización implica probar diferentes combinaciones de valores y seleccionar aquellos que maximizan el rendimiento en el conjunto de validación manualmente o mediante estrategias automatizadas. Técnicas como la búsqueda en cuadrícula (*grid search*), la búsqueda aleatoria (*random search*) o métodos más avanzados basados en optimización bayesiana pueden emplearse para este fin.

Métricas de rendimiento

En el aprendizaje profundo, las métricas de rendimiento son esenciales para cuantificar cómo de bien un modelo realiza la tarea para la que fue diseñado. Algunas de estas métricas difieren según el tipo de tarea a realizar. En este apartado nos centraremos en aquellas relativas a tareas de clasificación, por ser el problema abordado en la Parte II.

- ***Loss* (Pérdida).** Es una media del error que indica cuánto se ajusta la red a los datos no vistos, tanto durante la etapa de entrenamiento (*train loss)* como en la de validación (*validation loss*).

- ***Accuracy* (Exactitud o precisión).** Es la proporción de predicciones correctas sobre el total de predicciones. Aunque intuitiva, puede ser engañosa en conjuntos de datos desbalanceados. Al igual que la pérdida, se mide tanto en entrenamiento (*training accuracy*) como en validación (*validation accuracy*).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

- ***F1-Score* (Puntuación F1).** Es la media armónica de la precisión o pureza (*precision*) y la exhaustividad (*recall*). Es una métrica robusta, que mide el compromiso entre las dos anteriores. Para la optimización de parámetros y los resultados finales, se emplea el F1 macro ponderado, que es la media de las puntuaciones F1 de cada una de las clases.

$$Precision = \frac{TP}{TP + FP} \tag{2.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.3}$$

$$F1 = \frac{2\,Precision\,Recall}{Precision + Recall} \tag{2.4}$$

$$F1_{\text{macro}} = \frac{1}{K}\sum_{i=1}^{K} F1_i \tag{2.5}$$

- **Matrices de confusión**. Se trata de una tabla de doble entrada que detalla predicciones vs. etiquetas reales, revelando sesgos sistemáticos y permitiéndonos visualizar el rendimiento de un algoritmo de clasificación. Las filas representan las clases verdaderas y las columnas las clases predichas. Permite identificar qué tipos de errores comete el modelo, es decir, qué clases se confunden entre sí.
- **Curvas de entrenamiento y validación.** Son gráficas que muestran la pérdida y la exactitud a lo largo de las épocas de entrenamiento y validación. Ayudan a diagnosticar el subajuste (pérdida alta y paralela en ambas) o el sobreajuste (buenas gráficas de entrenamiento pero con divergencia creciente entre ambas).

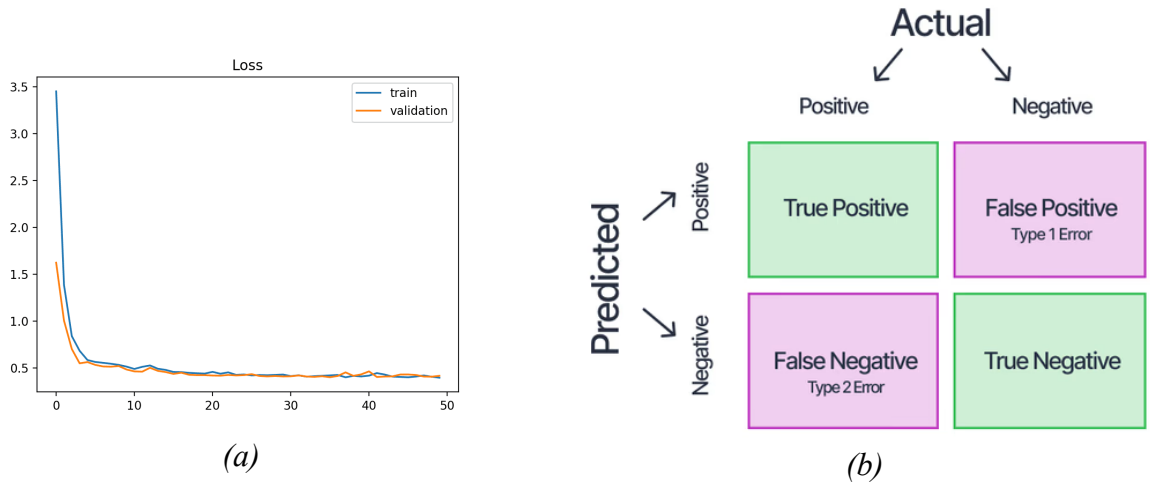


*Figura 2.8. (a) Curvas de pérdida para un modelo con buen ajuste. (b) Diagrama de la matriz de confusión (*Confusion Matrix*, s. f.).*

2.2. Estimación de la pose humana

2.2.1. Introducción

La Estimación de la Pose Humana (HPE, por sus siglas en inglés *Human Pose Estimation*) es el proceso que consiste en localizar los puntos más relevantes en las articulaciones del cuerpo humano (*keypoints*) y, a partir de ellos, inferir la configuración espacial o postura del cuerpo en imágenes o secuencias de vídeo. El objetivo es obtener una representación esquelética del sujeto, que puede ser bidimensional o tridimensional, especificando las coordenadas de cada articulación en un espacio definido.

Tradicionalmente, la HPE ha dependido de sistemas ópticos de captura de movimiento basados en marcadores reflectantes y cámaras infrarrojas, así como de la extracción manual de características y el uso de modelos probabilísticos o de partes deformables. Sin embargo, a pesar de que proporcionan una precisión submilimétrica, estos enfoques presentaban limitaciones en cuanto a su coste económico, y la necesidad de un entorno controlado y un hardware específico.

Las primeras aproximaciones basadas en imágenes recurrieron a técnicas de visión por computador clásicas: modelos y árboles pictóricos, descriptores HOG, y ensamblados de clasificadores. No obstante, el advenimiento del aprendizaje profundo, y en particular de las Redes Neuronales Convolucionales (CNNs), ha supuesto un punto de inflexión en este campo. Las arquitecturas de *deep learning* pueden aprender jerarquías de características directamente de los datos de entrada, alcanzando niveles de generalización significativamente superiores con un bajo coste y de forma completamente accesible. En la actualidad los mejores resultados se obtienen con modelos híbridos que combinan redes convolucionales profundas, atención multicabeza y refinamiento temporal mediante RNN o módulos LSTM.

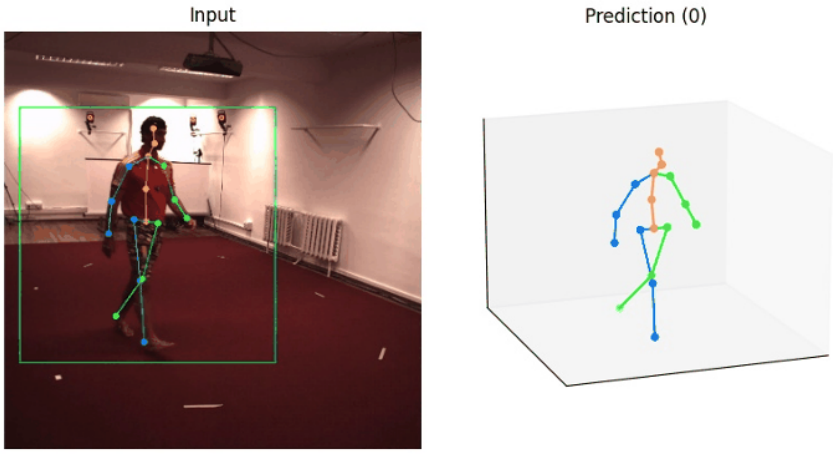


Figura 2.9. Ejemplo de estimación de pose 3D mediante visión artificial.

La relevancia de la HPE en entornos clínicos y de rehabilitación es considerable, ya que proporciona una metodología no invasiva y sin marcadores para el análisis cuantitativo del movimiento humano. Esto contrasta con los sistemas optoelectrónicos que requieren la colocación precisa de elementos reflectantes sobre el paciente, o incluso con los sensores inerciales que, aunque portátiles, implican adherir dispositivos al cuerpo. La HPE basada en visión permite la evaluación en entornos más naturales y menos restrictivos, facilitando la monitorización a largo plazo y la telerehabilitación al requerir únicamente una o varias cámaras convencionales.

Una vez que se ha estimado la pose humana, ya sea en 2D o en 3D, se abre un abanico de aplicaciones analíticas. A partir de las coordenadas de los *keypoints* es posible calcular parámetros biomecánicos de gran interés clínico, tales como ángulos articulares, longitudes de los segmentos corporales, velocidades y aceleraciones de los centros articulares. Estos datos cinemáticos son la base para análisis más complejos como la segmentación de las fases de la marcha (identificando eventos clave como el contacto inicial del talón o el despegue de los dedos), la evaluación del rango de movimiento articular (ROM), la detección de patrones de movimiento compensatorios o anómalos, el seguimiento objetivo de la recuperación funcional de un paciente, o incluso la cuantificación del riesgo de caídas. La HPE 3D, en particular, es de gran interés para el análisis cinemático completo, ya que proporciona la información espacial necesaria para un cálculo preciso de ángulos y movimientos en los tres planos del espacio, superando las limitaciones de la ambigüedad inherente a las proyecciones 2D.

La integración con IMUs proporciona ventajas complementarias: reduce ambigüedades de profundidad, mejora la robustez frente a oclusiones y posibilita la monitorización en entornos sin línea de visión directa. De esta manera la combinación de visión por computador y sensores portátiles ilustra el estado del arte actual en análisis cinemático de bajo coste para aplicaciones clínicas y de rehabilitación.

2.2.2. Análisis instrumentado y cinemática inversa

El análisis instrumentado de la detección de pose humana se fundamenta en la adquisición de señales cinemáticas y cinéticas que describen la interacción entre el cuerpo y el entorno. A grandes rasgos, se puede distinguir entre métodos que emplean marcadores o sensores situados en el cuerpo del paciente (*marker-based*) como por ejemplo Vicon o sensores inerciales y métodos que no requieren de *wearables* o hardware específico en el cuerpo para funcionar (*markerless*), como por ejemplo aquellos basados en vídeo (Kanko et al., 2021). Dentro de estos últimos, podemos distinguir a su vez entre sistemas de detección 2D y 3D, si bien es común la reconstrucción de poses tridimensionales a partir de secuencias de imágenes en dos dimensiones (Park et al., 2016). Este trabajo se centrará en la estimación de pose 3D, dado que gracias a los últimos avances en visión computacional, esta opción es actualmente la más robusta y permite una mayor precisión a la hora de extraer información con relevancia clínica.

Existen diversas alternativas que recogen diversos parámetros para realizar este tipo de análisis. Cada tecnología aporta una resolución temporal o espacial diferente y condiciona los métodos posteriores de segmentación del ciclo de la marcha u otras aplicaciones. Los métodos más empleados en entornos profesionales se describen a continuación.

Plataformas de fuerza (*force plates)*

Las plataformas de fuerza son transductores piezoeléctricos o galgas extensiométricas que registran las tres componentes de la fuerza de reacción del suelo y el punto de aplicación. El instante exacto en que la componente vertical cruza umbrales próximos a cero permite localizar con gran precisión los eventos de marcha de contacto inicial del talón y el despegue del pie. Además, la fuerza resultante es imprescindible para la obtención de momentos articulares mediante dinámica inversa (Liu et al., 2010). Aunque precisas para la cinética (proporcionan información sobre las fuerzas verticales, anteroposteriores, mediolaterales, y el centro de presiones), no ofrecen información directa sobre el movimiento de los segmentos corporales.

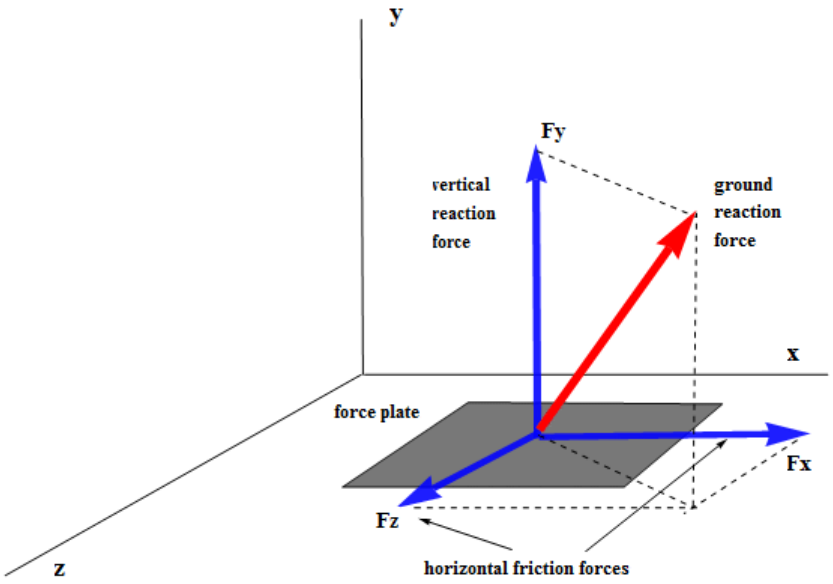


Figura 2.10. Ejemplo de las fuerzas en una force plate.

Sistemas ópticos y optométricos de captura de movimiento

Las soluciones que emplean un sistema multicámara por infrarrojos, como OptiTrack, Qualisys y especialmente Vicon, son considerados el *gold standard* en el análisis de movimiento. Estas herramientas identifican marcadores reflectantes pasivos adheridos a la piel del sujeto en puntos anatómicos clave y reconstruyen sus trayectorias tridimensionales con errores inferiores a 1 mm dentro del volumen calibrado (Windolf et al., 2008). La alta frecuencia de muestreo,

típicamente 100-250 Hz, facilita el cálculo de velocidades y aceleraciones de segmentos corporales y su posterior introducción en algoritmos de cinemática inversa.

Sus principales limitaciones son su elevado coste, la dependencia de un laboratorio dedicado y la sensibilidad al ocultamiento de marcadores. Dado que uno de los objetivos de este trabajo ha sido explorar alternativas accesibles, de bajo coste y funcionales en entornos del día a día, se optó por emplear sensores inerciales que constituyeron el *ground truth* en la captura de movimiento. Vicon fue un sistema descartado desde el inicio para la realización de este proyecto, tanto por el hecho de que no se disponía de un *setup* apropiado para realizar las adquisiciones con este sistema, como por el incumplimiento de los requisitos mencionados anteriormente.

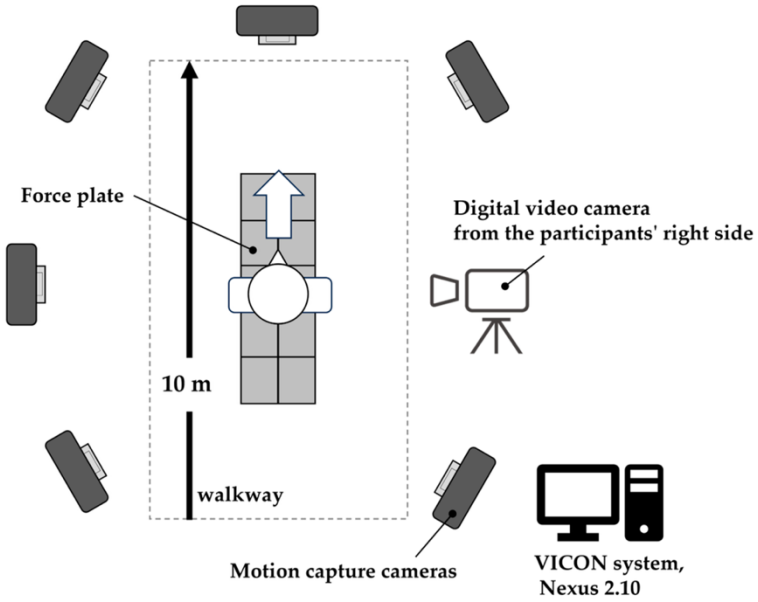


Figura 2.11. Setup de captura de movimiento típico empleando Vicon y force plates.

Sensores inerciales (IMUs)

Las Unidades de Medida Inercial (*Inertial Measurement Units*, IMUs) son dispositivos electrónicos que disponen de múltiples sensores para medir y reportar la fuerza específica, la velocidad angular y, en ocasiones, la orientación de un cuerpo al que están sujetos. Normalmente integran acelerómetros para medir la aceleración lineal, giroscopios para medir la velocidad angular y a veces magnetómetros para mejorar la estimación de la orientación absoluta al proporcionar una referencia respecto al campo magnético terrestre. Mediante fusión de sensores por filtros de Kalman o filtros complementarios se estima la orientación de cada segmento en forma de cuaterniones, que suponen una medida precisa de la orientación espacial. A partir de esto, puede obtenerse la cinemática del segmento corporal al que se adhieren.

Su colocación directa sobre segmentos anatómicos, la ausencia de cámaras y su reducido tamaño convierten a los IMUs en la opción preferente para estudios fuera del laboratorio y para el seguimiento longitudinal de pacientes (Caramia et al., 2025). Por tanto, los IMUs ofrecen una alternativa accesible y de bajo coste respecto a los sistemas Vicon utilizados tradicionalmente en entornos controlados. Debido a su capacidad para proporcionar datos cinemáticos precisos, especialmente en cuanto a cinemática y velocidad angular, fueron usados como *ground truth* para validar las tecnologías de estimación de pose basadas en visión por computador a lo largo de la primera parte de este trabajo. En específico, se emplearon 5 sensores inerciales que habían sido fabricados previamente de manera customizada y que ofrecieron datos sobre la posición tridimensional de segmentos corporales determinados en forma de cuaterniones.

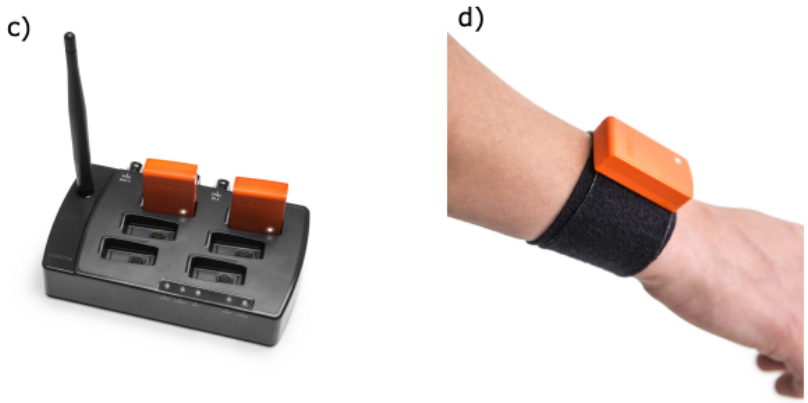


Figura 2.12. Kit de desarrollo con IMUs Xsens MTw Awinda.

Cinemática inversa

La cinemática inversa es un procedimiento de optimización que determina los ángulos articulares y las posiciones y orientaciones de los segmentos corporales a partir de medidas externas. Este proceso consiste en minimizar la función de error que cuantifica la distancia entre la posición de los marcadores experimentales y la posición de los marcadores virtuales ligados a un modelo esquelético del cuerpo humano. El problema se resuelve *frame* a *frame* de forma independiente, por lo que no acumula error temporal. Las técnicas más extendidas emplean mínimos cuadrados ponderados y restricciones de rango articular para mantener la plausibilidad anatómica.

Cabe mencionar que esta técnica es especialmente relevante para el análisis de la marcha, ya que permite estudiar los patrones y realizar evaluaciones clínicas a partir de los datos de estimación de pose de un paciente (Yu et al., 2020). Para realizar la cinemática inversa y calcular los ángulos articulares, se optó por OpenSim, un software médico de código abierto desarrollado

por el *National Center for Simulation in Rehabiliation Research* (NCSRR) de Stanford, y que permite crear y analizar modelos musculoesqueléticos para simular el movimiento humano.

El flujo de procesamiento con dicha herramienta es el siguiente:

1. Escalado de un modelo genérico a la antropometría del sujeto mediante distancias inter-marcador y medidas corporales. La elección del modelo musculoesquelético depende del tipo de análisis que se quiera realizar. Por ejemplo, el modelo Gait2392 está enfocado las extremidades inferiores del cuerpo humano, por lo que es especialmente útil en aplicaciones de análisis de la marcha.
2. Asignación de los marcadores experimentales al modelo virtual.
3. Resolución de la cinemática inversa con el *solver* interno que implementa optimización por gradiente.
4. Exportación de los ángulos articulares y velocidades angulares para análisis posteriores.
5. Generación de gráficas y análisis estadístico de los resultados obtenidos.

De este modo, se pudo reconstruir el movimiento articular a partir de datos de captura de movimiento y calcular ángulos y trayectorias de segmentos corporales con facilidad.

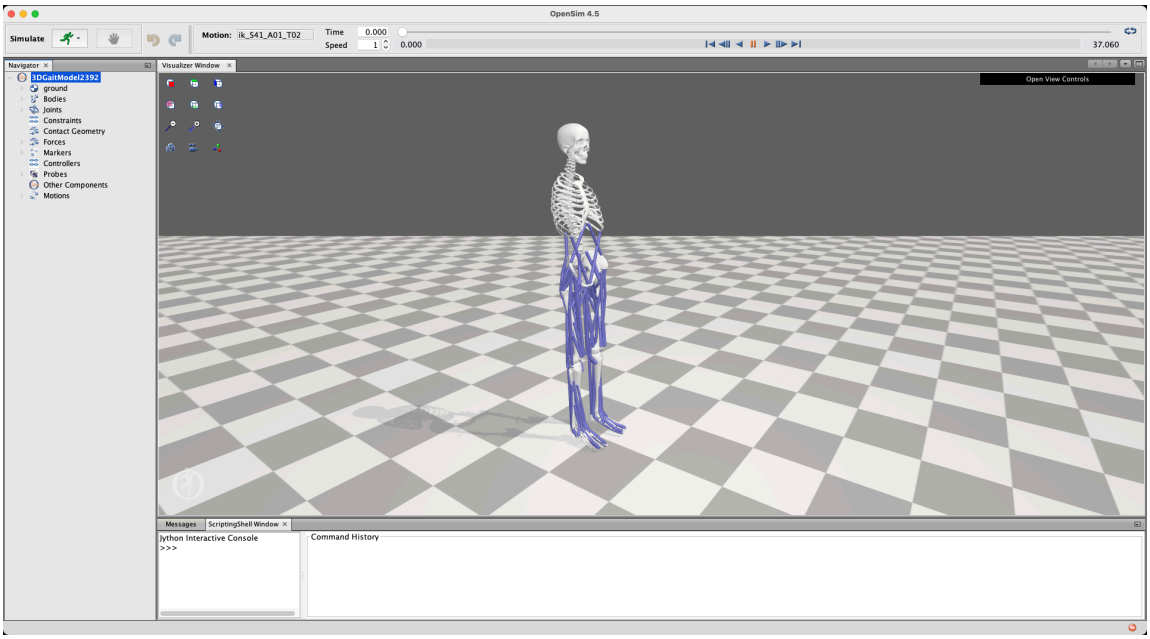


Figura 2.13. Cinemática inversa en OpenSim empleando el modelo Gait2392 (sólo articulaciones inferiores).

2.2.3. Enfoques de HPE mediante aprendizaje profundo basados en vídeo

En esta sección se describen los sistemas de HPE basados en vídeo utilizados. La elección de los modelos a incluir en la comparativa fue realizada en base a dos criterios:

1) Elegir modelos estado del arte y que empleasen tecnologías consolidadas pero recientes, siguiendo el formato Human3.6M y en base a detección monocular y en tres dimensiones.
2) Asegurar la inclusión de arquitecturas diversas y representativas dentro del campo de la visión por ordenador, utilizando diferentes diseños de red neuronal y mecanismos de atención, con el fin de enriquecer la comparativa.

Para cumplir con estos requisitos, se empleó el recurso (*Papers with Code - 3D Human Pose Estimation*, s. f.) para consultar los *benchmarks* realizados sobre el conjunto de datos Human3.6M (Ionescu et al., 2014), que constituye un estándar en la detección de pose basada en vídeo. De este modo, cada modelo finalmente seleccionado ejemplifica un enfoque distinto al problema de la HPE, lo que permite abarcar una selección más amplia de algoritmos y procedimientos durante la evaluación y la comparación.

MotionAGFormer

MotionAGFormer (Mehraban et al., s. f.) es un modelo híbrido que combina las ventajas de los mecanismos de atención basados en *transformers* junto con *Graph Convolutional Networks* (GCNs) para aumentar la precisión y la eficiencia en la estimación de pose humana en tres dimensiones. Su premisa parte de una limitación detectada en los modelos puramente basados en *self-attention*: aunque el mecanismo global captura bien las dependencias de largo alcance, tiende a descuidar las relaciones locales entre articulaciones adyacentes. Para solventarlo, los autores introducen el bloque AGFormer (Jiang et al., 2023), que procesa la señal por dos vías paralelas (una rama *transformer* y otra GCN) y fusiona de forma adaptativa ambas representaciones a lo largo de la red.

Como puede verse en la Figura 2.14, cada bloque Attention-GCNFormer contiene a su vez dos niveles jerárquicos: un MetaFormer espacial, que trata cada articulación dentro de un fotograma como un token independiente para aprender la coherencia estructural intra-frame, y un MetaFormer temporal, que considera cada fotograma como un token para capturar la dinámica inter-frame. En la primera rama, ambos niveles recurren a *Multi-Head Self-Attention* (MHSA), mientras que en la segunda se emplean convoluciones sobre grafos que respetan la topología corporal (espacial) o bien conectan cada articulación con sus k-vecinos temporales de mayor similitud (temporal). Esta combinación ofrece un compromiso equilibrado entre contexto global y precisión local, siendo más eficaz que utilizar cualquiera de los dos módulos de forma aislada o secuencial.

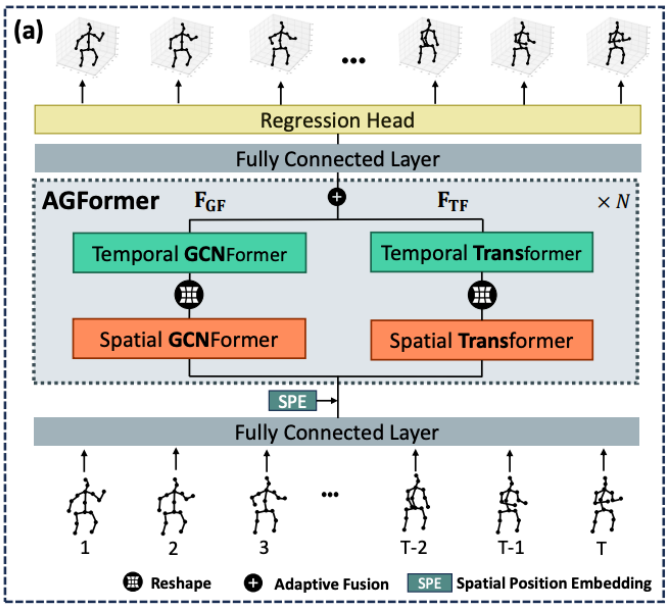


Figura 2.14. Arquitectura dual-stream con N bloques espacio-temporales (Mehraban et al., s. f.).

El modelo completo se inicia con una proyección lineal que mapea cada coordenada en 2D (-x, y, confianza) a un espacio de características de dimensión *d*. Tras añadir *embeddings* posicionales espaciales, la secuencia atraviesa *N* bloques AGFormer, produciendo una representación latente que se refina mediante una capa lineal y una activación tangente hiperbólica para generar un vector semántico de movimiento de mayor dimensión *d'*. Finalmente, una cabeza de regresión predice la trayectoria 3D completa de todas las articulaciones en una única pasada, minimizando una pérdida combinada de posición y suavidad (velocidad). Esta estrategia evita la superposición de ventanas y reduce drásticamente el coste de inferencia habitual en modelos *many-to-one*.

Este modelo consigue resultados estado del arte en los conjuntos Human3.6M (Ionescu et al., 2014) y MPI-INF-3DHP (Mehta et al., 2017) siendo computacionalmente más eficiente que sus predecesores. Con objeto de adaptarse a diferentes escenarios, los autores proponen cuatro variantes: XS, S, B y L, que escalan el número de capas (*N*), la anchura interna (*d*) y la longitud de la secuencia de entrada (*T*). En este trabajo, se optó por emplear la variante B (*Base*), ya que logra un equilibrio óptimo entre precisión (38,4 mm de MPJPE bajo protocolo P1 en Human3.6M) y eficiencia computacional (≈198 M MACs / frame), con sólo 11.7 M de parámetros.

MotionBERT

MotionBERT (Zhu et al., 2022) parte de la premisa de que muchas tareas de vídeo centradas en el ser humano (estimación de la pose 3D, reconocimiento de acciones o reconstrucción de mallas) comparten patrones cinemáticos comunes que pueden capturarse con un

único modelo y transferirse después a cada tarea con un ajuste mínimo. Para ello, los autores proponen un preentrenamiento autosupervisado que consiste en recuperar la secuencia de movimiento 3D a partir de esqueletos 2D intencionadamente degradados (con máscaras y ruido). Esta tarea obliga a la red a aprender la estructura geométrica, las restricciones anatómicas y la dinámica temporal inherentes al movimiento humano.

La arquitectura empleada es un *Dual-stream Spatio-temporal Transformer* o DSTformer (Zhang et al., 2024). Un flujo espacial-temporal y otro temporal-espacial procesan en paralelo las relaciones entre articulaciones dentro de cada fotograma y la evolución temporal de cada articulación como puede observarse en la Figura 2.15. Un mecanismo de fusión adaptativa pondera ambos flujos según la acción y la articulación analizadas, lo que permite especializar cada rama sin perder información complementaria.

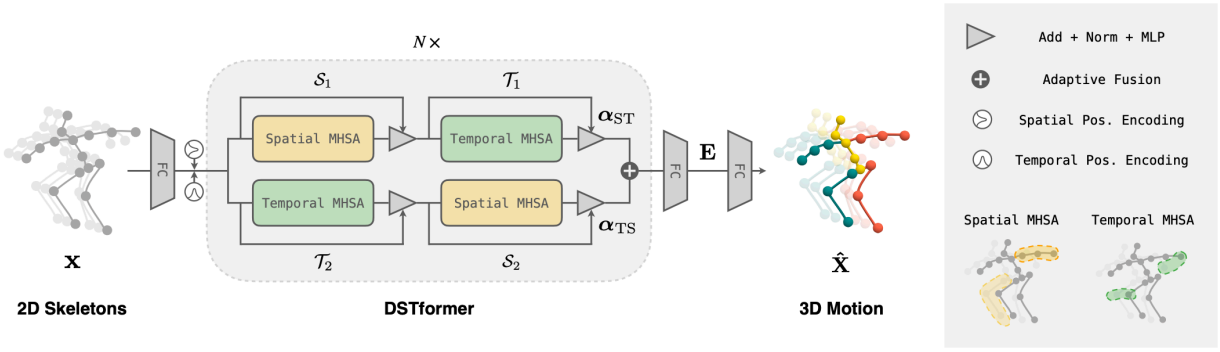


Figura 2.15. Arquitectura DSTFormer en el backbone de MotionBERT.

Para el preentrenamiento se combinan fuentes heterogéneas: datos de captura de movimiento 3D de alta precisión AMASS (Mahmood et al., 2019) y Human3.6M (Ionescu et al., 2014) proyectados ortogonalmente a 2D, anotaciones de conjuntos como PoseTrack (Andriluka et al., 2018) y secuencias 2D *in-the-wild* extraídas con detectores de pose. Una vez aprendido el codificador de movimiento, basta añadir una capa lineal o un MLP poco profundo y afinar todo el modelo durante unas pocas épocas para cada tarea específica.

Tras el *fine-tuning*, se obtiene un MPJPE de 37,5 mm en Human3.6M (clip de 243 *frames*), el mejor hasta la fecha entre los métodos secuencia a secuencia basados en *transformers*, y un MPJVE de 1,7 mm, indicador de gran coherencia temporal. En reconocimiento de acciones sobre NTU-RGB+D se alcanza una precisión Top-1 del 93.0% en Cross-Subject y del 97.2% en Cross-View, superando o igualando a métodos específicos del dominio. Asimismo, al usarse como refinador de malla humana, reduce de forma sustancial el error de vértice de soluciones RGB-basadas como SPIN o MAED.

MMPose: VideoPoseLift + VideoPose3D

MMPose es un *toolbox* de estimación de pose basado en PyTorch y desarrollado por OpenMMLab (*Overview — MMPose 1.3.2 documentation*, s. f.). En este trabajo se empleó una combinación de tres módulos especializados (detector de personas, estimador de pose 2D, y "elevador" de pose 2D a 3D) para crear una solución propia lo más eficiente posible.

En la primera etapa, RTMDet, una CNN optimizada para detección en tiempo real, localiza sujetos en cada fotograma utilizando recuadros delimitadores. En el *backbone*, se emplea CSPNet modificado con conexiones parciales y una FPN (*Feature Pyramid Network*) escalada. La segunda etapa aplica RTMPose, una red 2D que combina un *backbone* estilo HRNet con una cabeza SimCC (*Simulated Classificationbased Coordinate Regression*). Esta aproximación reformula la regresión de coordenadas discretas como dos subtareas de clasificación independientes para cada eje, obteniendo precisión subpíxel con un coste computacional bajo.

En la etapa final se emplea VideoPoseLift, una implementación customizada de VideoPose3D (Pavllo et al., 2019) consistente en un elevador de poses 2D a 3D a través de análisis espaciotemporal. Este sistema aplica convoluciones temporales de 1D sobre secuencias de marcadores 2D con bloques residuales, capturando dependencias a largo plazo. Según el trabajo original, este modelo alcanzó 37.8 mm de MPJPE en Human3.6M, mejorando en 6 mm el mejor resultado previo y reduciendo el error un 11 %. Este método se entrenó de forma semi-supervisada (ver Figura 2.16) con conjuntos a gran escala (Human3.6M y COCO) y cada submódulo está preentrenado en su dominio para reforzar la robustez. Además, se empleó el motor de inferencia de MMPose que encadena los tres modelos en cascada y aplica postprocesado acelerado por GPU.

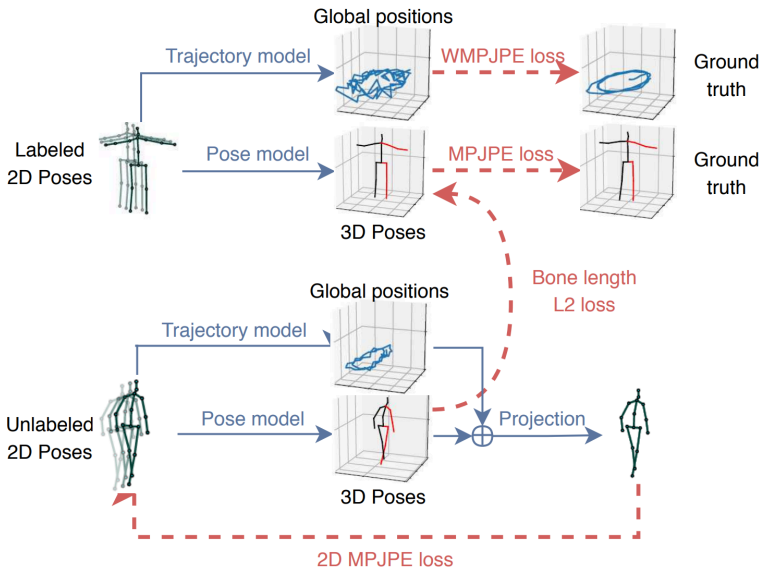


Figura 2.16. Entrenamiento semi-supervisado con poses 2D etiquetadas y no etiquetadas como entrada.

NVIDIA Maxine AR SDK BodyTrack

El sistema BodyTrack forma parte del Maxine AR SDK (*Software Development Kit*) de la compañía NVIDIA. Se trata de una solución comercial y de código cerrado de estimación y seguimiento de la pose humana en 3D mediante *bounding-boxes* usando una sola cámara en tiempo real. Este algoritmo infiere 34 marcadores anatómicos en coordenadas (x, y, z) y ofrece, además, los vectores de orientación articular, metadatos con vectores de confianza en la detección y la identificación de múltiples personas en la escena. La estimación se ejecuta sobre GPUs NVIDIA que tengan núcleos tensoriales (arquitecturas Turing, Ampere o Ada), requisito que garantiza latencias inferiores a 30 ms en resoluciones HD cuando se utiliza CUDA 11 + TensorRT.

Este modelo fue el empleado originalmente en el desarrollo de la base de datos VIDIMU, por ser una de las pocas alternativas capaces de inferir coordenadas 3D absolutas a partir de vídeo monocular, de ahí el interés de incorporarlo en este trabajo. No obstante, en los años sucesivos a dicho estudio se han ido publicando opciones más potentes y con arquitecturas con un mayor número de parámetros, las cuales han sido evaluadas en secciones posteriores.

A diferencia del resto de métodos empleados, que eran *open-source*, BodyTrack es de naturaleza propietaria, por lo que su arquitectura de red, los conjuntos de datos de entrenamiento y las estrategias de optimización no se han hecho públicos. Esta falta de transparencia limita la reproducibilidad científica y dificulta la adaptación del modelo a nuevos dominios o entornos (por ejemplo, en situaciones clínicas con necesidades de privacidad reforzada). A pesar de constituir un punto de referencia industrial, su planteamiento de “caja negra”, la dependencia del hardware NVIDIA y el licenciamiento comercial, BodyTrack refuerza la discusión sobre la viabilidad de soluciones cerradas frente a alternativas de código abierto y subraya la importancia de considerar tanto el rendimiento técnico como los aspectos relacionados con la transparencia.

2.3. Segmentación de la marcha

2.3.1. Introducción

La marcha humana se define como el patrón cíclico de movimientos coordinados que permiten el desplazamiento bípedo durante la locomoción (Prakash et al., 2018). Un ciclo de marcha abarca el intervalo comprendido entre dos contactos sucesivos del mismo pie con el suelo (también denominado zancada o *stride*). Cada ciclo se divide convencionalmente en pasos, donde un paso es la distancia o el tiempo existente entre el contacto inicial de un pie y el contacto inicial del pie contralateral. Estos ciclos presentan una periodicidad que facilita el análisis cinemático y cinético mediante técnicas instrumentales como la captura de movimiento óptica, las plataformas de fuerza o los sensores inerciales (Caldas et al., 2017; Smirnova et al., 2022).

Clásicamente se distinguen dos fases generales en cada uno de los ciclos: fase de apoyo (*stance*) y fase de oscilación (*swing*). La fase de apoyo representa aproximadamente el 62 % del ciclo y comienza con el contacto inicial del talón (*heel strike*). Durante esta fase el pie se mantiene en contacto con el suelo para proporcionar estabilidad y propulsión. La fase de oscilación, que comienza con el *toe off*, ocupa el 38 % restante y comprende el intervalo en el que el mismo pie se encuentra en el aire avanzando para preparar el siguiente contacto. Cabe destacar que esta terminología es aplicable únicamente en condiciones de marcha no patológica, ya que en pacientes patológicos puede ocurrir que se apoyen o levanten otras partes del pie en dichos instantes (Wagenaar & van Emmerik, 1994).

Como se observa en la Figura 2.17, para describir con mayor detalle la mecánica de la marcha se utilizan subfases y eventos específicos (Prakash et al., 2018):

- **Contacto inicial** (*heel strike o initial contact*). Ocurre entre el 0 y el 3 % del ciclo total. Marca el inicio del ciclo cuando el talón establece contacto con el suelo. En un patrón de marcha no patológico, se realiza una flexión de la cadera, extensión de la rodilla pequeña o prácticamente nula y flexión dorsal en el tobillo.
- **Respuesta a la carga** (*loading response*). 3-12 %. Transmite progresivamente el peso corporal a la extremidad mientras el pie de referencia completo adopta contacto plantar. Se genera una flexión de la rodilla para asegurar el equilibrio y amortiguar el impacto.
- **Apoyo medio** (*mid stance*). 12-31 %. El centro de masa se desplaza sobre el pie, de forma que la pierna de referencia aguanta todo el peso, y el cuerpo alcanza la posición más alta. La cadera y la rodilla se encuentran estabilizadas.
- **Apoyo terminal** (*terminal stance*). 31-50 %. El talón se eleva del suelo (*heel off*) de forma que el centro de masa queda desplazado hacia delante del pie y se genera la fase propulsiva del miembro.
- **Pre-oscilación** (*pre-swing o toe off*). 50-62 %. Final de la fase de apoyo, el antepié aún contacta con el suelo y ocurre el Toe Off, donde los pies se despegan del suelo.
- **Balanceo inicial** (*initial swing*). 62-75 %. El pie se despega por completo y la rodilla se flexiona para acortar la extremidad.

- **Balanceo medio** (*mid swing*). 75-87 %. La tibia se adelanta hasta quedar vertical y se produce la máxima flexión de cadera y la máxima separación entre el pie de referencia y el suelo gracias a los flexores dorsales.
- **Balanceo terminal** (*terminal swing*). 87-100 %. Se produce desaceleración y el miembro se extiende y se prepara la posición para el siguiente contacto inicial. La rodilla se encuentra extendida y el pie ha de tener una posición neutral respecto a la pierna.

Es importante subrayar la existencia de eventos temporales intermedios entre fases que facilitan la segmentación de la marcha, siendo los ejemplos más notables *heel strike*, *mid stance* y *toe off*. El conocimiento preciso de estos instantes junto con métodos de segmentación automática permite calcular parámetros como la duración de la doble fase de apoyo, la cadencia, la longitud de paso o la asimetría bilateral, métricas cruciales en la valoración clínica (Hollman et al., 2011).

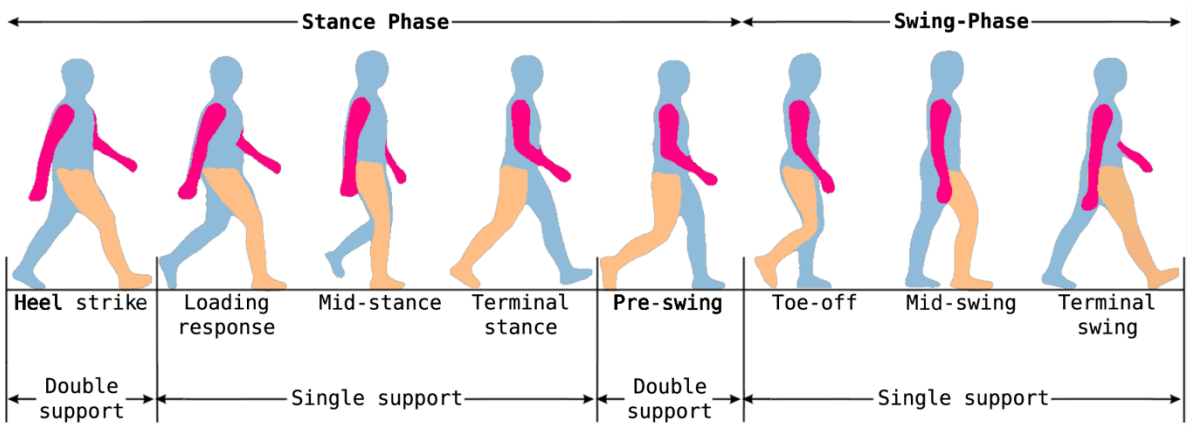


Figura 2.17. Eventos y fases principales de un ciclo de marcha (Zafra-Palma et al., 2025).

La relevancia clínica de los patrones de marcha radica en que numerosos trastornos neuromusculares, ortopédicos y cardiovasculares se manifiestan de forma temprana como alteraciones espaciales o temporales del patrón locomotor. El análisis cuantitativo de la marcha se emplea rutinariamente para el diagnóstico funcional de pacientes con lesiones medulares, ictus, enfermedad de Parkinson o parálisis cerebral (Armand et al., 2016; Morris et al., 2001). Además, constituye una herramienta de seguimiento posquirúrgico y de valoración de tratamientos farmacológicos o de rehabilitación. En la práctica deportiva y en el diseño de prótesis, ortesis y exoesqueletos, la caracterización objetiva de la marcha también es fundamental para optimizar el rendimiento y prevenir lesiones.

2.3.2. Análisis instrumentado de la marcha

Los métodos de captura de la marcha se superponen en gran medida con las técnicas descritas en el apartado de análisis instrumentado de pose humana y cinemática inversa. Ambos parten de la necesidad de reconstruir con precisión las coordenadas de los segmentos corporales a partir de placas de presión, sistemas ópticos, sensores inerciales, etc. La detección fiable de la pose constituye por tanto el punto de partida indispensable para cualquier algoritmo de segmentación del ciclo de marcha, siendo el siguiente paso la interpretación y el procesado de estos datos de pose mediante algoritmos de detección de eventos.

Además, cabe mencionar que el aprendizaje profundo puede intervenir en dos momentos diferenciados de nuestro flujo de trabajo: (i) durante la propia captura, para extraer la pose directamente de imágenes o vídeos mediante visión artificial, y (ii) en la etapa de análisis, donde redes recurrentes u otras arquitecturas especializadas aprenden a etiquetar esas secuencias en fases o eventos discretos, cerrando así el proceso automático de segmentación de la marcha.

2.3.3. Algoritmos de detección de eventos y segmentación

A lo largo de las últimas décadas se han desarrollado y consolidado numerosos enfoques para detectar eventos característicos (*heel strike*, *toe off*, *mid swing*, etc.) y, a partir de ellos, subdividir una señal en fases o ciclos completos de marcha. La bibliografía los agrupa habitualmente en tres grandes familias:

- **Métodos deterministas** (basados en reglas)
- **Modelos probabilísticos** o estadísticos
- **Técnicas de aprendizaje automático** (convencional y profundo)

A continuación se describen los algoritmos más utilizados para detectar y segmentar la marcha, así como sus principios de funcionamiento.

Métodos basados en reglas y umbrales

Son los más antiguos y siguen estando muy extendidos en contextos clínicos y técnicos por su simplicidad. Consisten en identificar picos, valles o cruces por cero en variables cinemáticas o cinéticas previamente adquiridas.

- **Cruce por cero de la velocidad angular de la tibia**. La velocidad angular en el plano sagital cambia de signo en el evento de *mid swing*. Los picos adyacentes (anterior y posterior) a este evento marcan *heel strike* y *toe off*. Es el método empleado en este estudio para el etiquetado inicial de la base de datos.

- **Picos en la aceleración vertical del talón o del sacro**. Se emplea la señal de aceleración en lugar de la velocidad. En este caso, el máximo positivo suele coincidir con *heel strike* y el mínimo posterior con *toe off*.
- **Mínimos y máximos de ángulos articulares**. Se calculan los ángulos entre articulaciones a partir de datos de pose o coordenadas 3D adquiridos previamente. El mínimo de flexión de rodilla o el máximo de extensión de cadera señalan los eventos de contacto de interés.

Sus ventajas principales son su sencillez y el coste computacional nulo. Por otro lado, estos métodos tienen una alta sensibilidad al ruido, necesitan una calibración específica por sujeto y ofrecen un desempeño bajo en patrones patológicos o movimientos a gran velocidad.

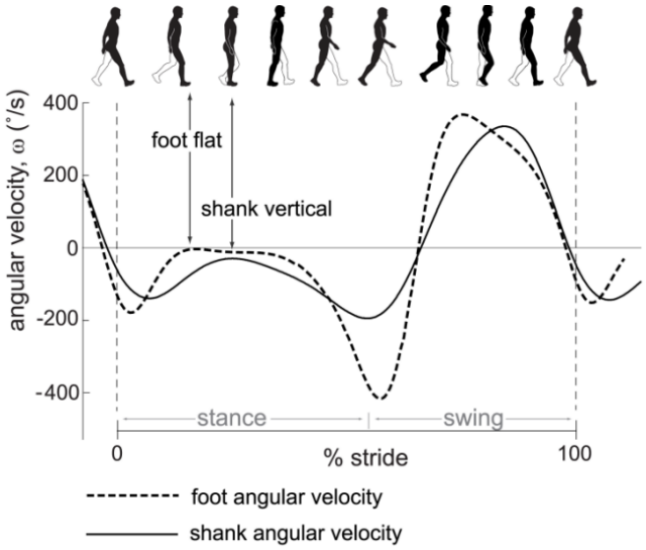


Figura 2.18. Ciclo de la marcha sobre una señal de velocidad angular.

Transformadas de dominio (CWT, STFT)

Para mejorar la robustez frente al ruido y la variabilidad, se puede aplicar la Transformada Continua Wavelet (CWT) o la Transformada de Fourier de Tiempo Reducido (STFT) a la señal de aceleración o velocidad y buscar crestas de energía en bandas de frecuencia asociadas al ciclo de marcha (~1-5 Hz) (Xue et al., 2010). La localización temporal de esos máximos proporciona instantáneas de evento. La CWT es generalmente más robusta al ruido que los métodos basados en umbrales simples y puede adaptarse mejor a variaciones en la velocidad de la marcha. No obstante, su coste computacional es superior y hay una gran dependencia de la selección de la wavelet madre.

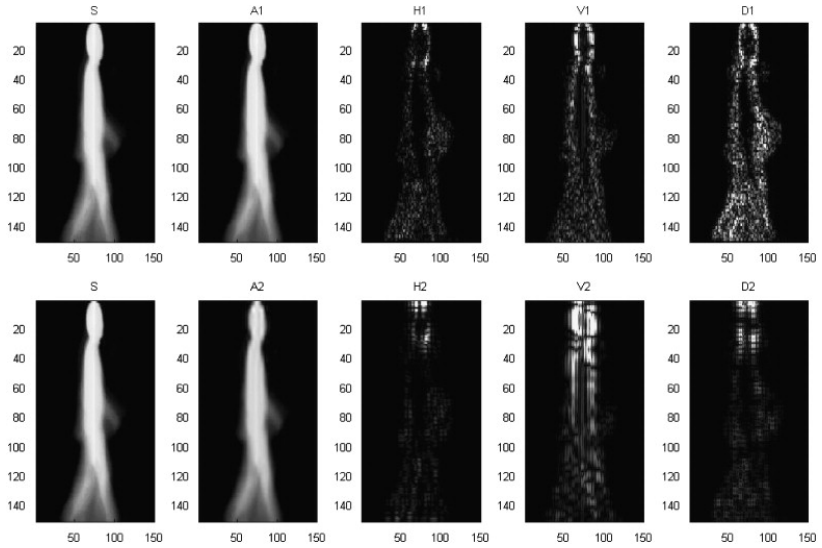


Figura 2.19. Descomposición wavelet de la energía de una imagen de marcha (Xue et al., 2010).

Modelos probabilísticos y secuenciales

Los enfoques basados en probabilidad interpolan bien cuando falta alguna muestra de sensor y cuantifican la incertidumbre, pero pierden precisión si la dinámica real difiere mucho del entrenamiento.

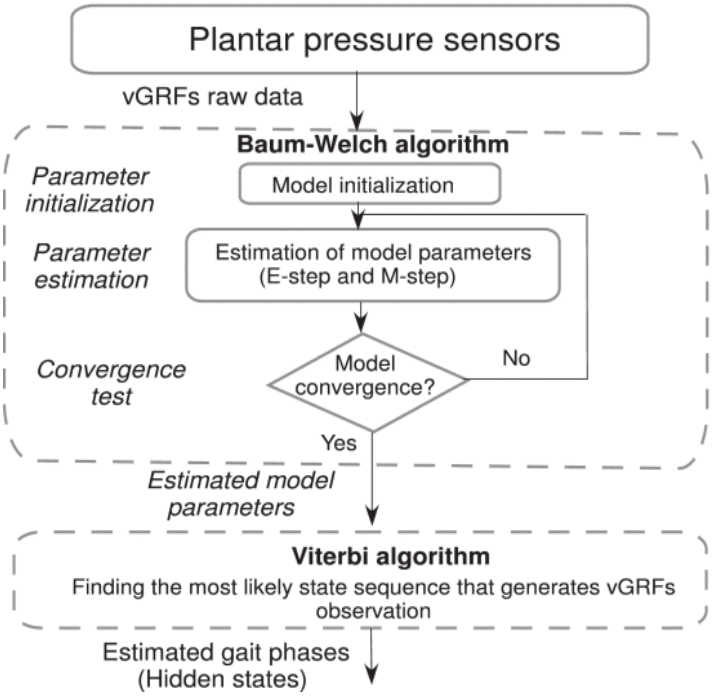


Figura 2.20. Segmentación empleando los algoritmos de Baum-Welch y de Viterbi (Attal et al., 2018).

Los más utilizados son los siguientes:

- **Hidden Markov Models (HMM).** Modelan la marcha como una cadena de estados ocultos (fase 0, 1, 2…) con probabilidades de transición aprendidas a partir de datos etiquetados. Las observaciones son vectores de características (p. ej. ángulos o aceleraciones). El algoritmo de Viterbi recupera la secuencia óptima de fases (Attal et al., 2018). Suponen independencia de primer orden y pueden quedarse cortos en trayectorias no estacionarias.
- **Conditional Random Fields (CRF).** Extienden a los HMM al relajar ciertas suposiciones de independencia y permiten usar características arbitrarias, pero exigen más datos y un entrenamiento más costoso.

Aprendizaje automático clásico

Los métodos basados en *machine learning* funcionan bien con bases moderadas y son rápidos en tiempo de inferencia, aunque no capturan dependencias temporales de largo alcance si no se añaden atributos derivados.

- **Máquinas de Soporte Vectorial (SVM) y K-Nearest Neighbours (k-NN).** Clasifican cada instante en fase usando ventanas deslizantes de características estadísticas (media, RMS, entropía, etc.) (Xue et al., 2010).
- **Árboles de decisión y Random Forest.** Crean reglas jerárquicas que son fácilmente interpretables por profesionales clínicos (Luo et al., 2020).
- **Gradient Boosting (XGBoost, LightGBM).** Generan modelos aditivos de alto rendimiento sin necesidad de una gran sintonía manual.

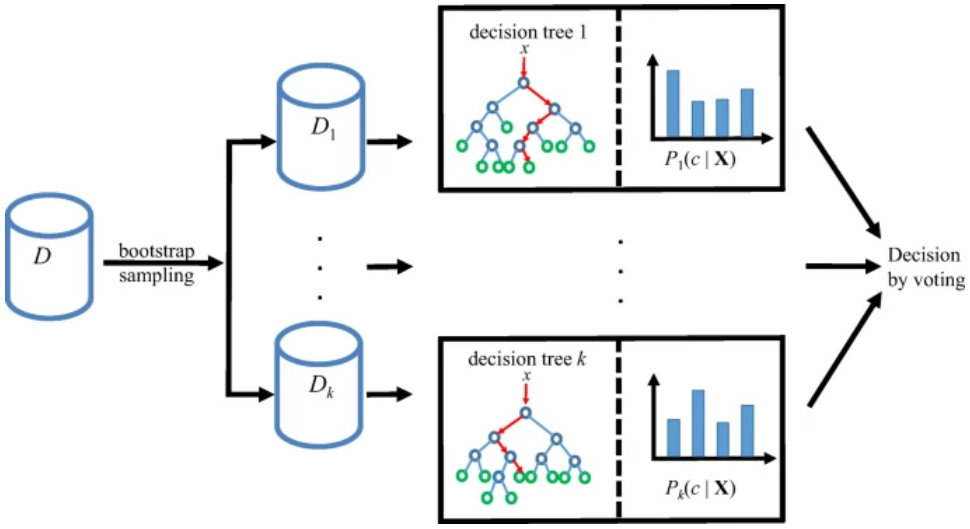


Figura 2.21. Funcionamiento del algoritmo Random Forest (Luo et al., 2020).

Redes neuronales profundas

Las distintas arquitecturas neuronales presentadas en la Sección 2.1.2 tienen una aplicabilidad directa en el problema de la segmentación de la marcha. A continuación se presenta una breve justificación de la motivación de uso de cada una de ellas y sus ventajas:

- **Redes Convolucionales 1D (CNN-1D).** Aprenden filtros espaciales sobre ventanas temporales y detectan patrones locales como la rapidez del pico de aceleración o la forma del impulso. Son invariantes a las traslaciones y robustas frente al ruido.
- **Redes MS-TCN++.** Se emplea una etapa inicial y múltiples etapas posteriores de refinamiento. Cada una utiliza convoluciones temporales dilatadas para capturar dependencias a largo plazo en la secuencia temporal. Además, introducen capas de dilatación dual que combinan campos receptivos grandes y pequeños, mejorando la precisión en la detección de transiciones entre acciones (Li et al., 2020)**.**
- **Redes Recurrentes (LSTM, GRU).** Modelan dependencias temporales a largo plazo. Son las más utilizadas en los últimos estudios clínicos y es la arquitectura adoptada en este trabajo.
- **Modelos híbridos CNN-LSTM.** Una CNN extrae características locales y una LSTM decide la fase estimada. Ofrecen buen equilibrio entre robustez y contexto, pero incrementan los parámetros y el riesgo de *overfitting* con bases de datos pequeñas.
- **Transformers y mecanismos de atención.** Capturan dependencias a cualquier distancia sin recurrencia, aunque exigen *datasets* mucho mayores y más recursos computacionales, por lo que pueden ser inadecuadas para bases clínicas pequeñas y medianas.

3. Parte I. Evaluación de arquitecturas neuronales de estimación de la pose humana

En la Parte I de este trabajo se presenta una evaluación exhaustiva de distintas arquitecturas de aprendizaje profundo para la estimación tridimensional de la pose humana a partir de vídeo, comparando sus resultados con las mediciones de referencia obtenidas mediante sensores inerciales. Para ello, se empleó el *dataset* VIDIMU, que incluye tanto secuencias de vídeo monocular como registros obtenidos a través de IMUs, y se analizaron cuatro modelos de última generación (MotionAGFormer, MotionBERT, VideoPose3D via MMPose y NVIDIA BodyTrack). Tras un proceso de preprocesado, filtrado y sincronización de señales, se calcularon métricas como RMSE, MAE, coeficiente de correlación y determinación para cuantificar la precisión y robustez de cada enfoque en diversas actividades que involucran tanto las extremidades inferiores del cuerpo humano como las superiores. A continuación, se discute la implicación de los detalles de diseño de las distintas arquitecturas en los resultados finales, y se proponen ámbitos de aplicación específicos para cada uno de los modelos evaluados. Esta comparación permite identificar las fortalezas y limitaciones de cada arquitectura en aplicaciones de análisis cinemático, estableciendo las bases para seleccionar la solución más adecuada en función de la actividad y el segmento corporal a estudiar.

3.1. Materiales y métodos

3.1.1. Descripción de la base de datos

Para la evaluación de las arquitecturas de estimación de la pose humana en 3D, se empleó el *dataset* VIDIMU (Martínez-Zarzuela et al., 2023), que comprende grabaciones en vídeo de 54 adultos sanos (36 hombres y 18 mujeres con edades entre 25.0 ± 5.4 años), realizando 13 actividades cotidianas con relevancia clínica. Para ello, se empleó una webcam Microsoft LifeCam Studio con una tasa de 30 *frames* por segundo y una resolución de 640x480 píxeles. Un *subset* de 16 sujetos fueron grabados también con cinco sensores inerciales (*Inertial Movement Units*, IMUs) propios, funcionando a 50 Hz y empleando la banda de frecuencias de 2.4 GHz. Es importante mencionar que para la realización de este trabajo y la posterior comparativa, se emplearon únicamente los datos de 16 sujetos (con identificadores S40-S42, S44, S46-S57) al ser los únicos correctamente grabados con ambas tecnologías. Durante la adquisición, los sujetos S43 y S45 se eliminaron debido a problemas técnicos detectados durante la recopilación de datos de los IMUs, y S48 se eliminó debido a errores significativos durante la detección de la pose corporal con BodyTrack causados por la poca estabilidad del enfoque de la cámara.

La ubicación de los sensores se modificó según el tipo de actividad realizada, como puede verse en la Figura 3.1. Para las actividades relativas a las extremidades superiores, los IMUs fueron

colocados mediante correas de velcro en la espalda, en la parte superior de los brazos, y en las muñecas. Para las actividades relativas a las extremidades inferiores, se colocaron en la parte baja de la espalda, en las partes superior e inferior de las piernas. Las actividades que integran este conjunto de datos pueden clasificarse en dos grupos según la parte del cuerpo donde se deseen realizar las mediciones:

- **Extremidades inferiores (A01-A04):** Caminar hacia delante, hacia atrás, siguiendo una línea, y levantarse y sentarse de manera consecutiva.
- **Extremidades superiores (A05-A013):** Mover una botella de lado a lado y beber de una botella alternando ambas manos, montar y desmontar una torre LEGO, lanzar una pelota y recogerla, coger una botella desde una posición elevada, y hacer una pelota con un papel y lanzarla.

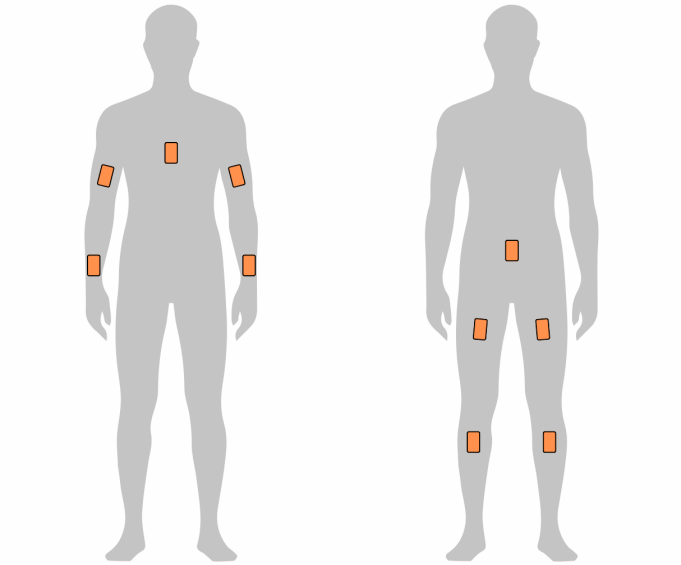


Figura 3.1. Ubicación de los sensores para articulaciones superiores (izquierda) e inferiores (derecha).

La estructura final de directorios puede verse en la Figura 3.2. La carpeta "dataset" contiene las coordenadas 3D de las articulaciones inferidas empleando los vídeos (extensión *.csv*) y los sensores inerciales (extensión *.raw, .mot* y *.sto*). La carpeta "analysis" contiene los gráficos resultado del procesamiento de señales (extensión *.svg*) y las carpetas "videosfullsize" y "videossmallsize" contienen los vídeos originales y comprimidos para reducir su tamaño respectivamente (en ambos casos con extensión *.mp4*). Algunas consideraciones adicionales para cada tipo de fichero son las siguientes:

- Ficheros *.raw*: Incluyen la información original de los cuaterniones en crudo (w, x, y, z) adquirida con los IMUs propios.
- Ficheros *.sto*: Siguen el formato tabular de OpenSim para guardar datos de series temporales.

- Ficheros *.mot*: Contienen los ángulos articulares en 3D a lo largo del tiempo computados mediante la cinemática inversa de OpenSim.
- Ficheros *.csv*: Contienen las coordenadas en 3D (x,y,z) estimadas a partir de vídeo por el sistema NVIDIA BodyTrack.

Todos los ficheros están nombrados siguiendo la expresión:

S[ID del Sujeto]_A[ID de la actividad]_T[ID del intento].[extensión del fichero]

Para el posterior procesamiento de los datos, sólo se empleó un único intento o *trial*, aunque se mantuvo el identificador debido a que durante la adquisición se descartaron intentos en ciertas actividades debido a calibraciones incorrectas, movimientos erróneos o fallos en la cámara o los sensores. El fichero "*bodyMeasurements.csv*" contiene las siguientes medidas del *subset* de 16 sujetos, todas ellas en cm: altura, peso, altura y anchura de hombros, envergadura de codos, muñecas y brazos, altura y anchura de cadera, altura de rodilla y tobillo, longitud del pie.

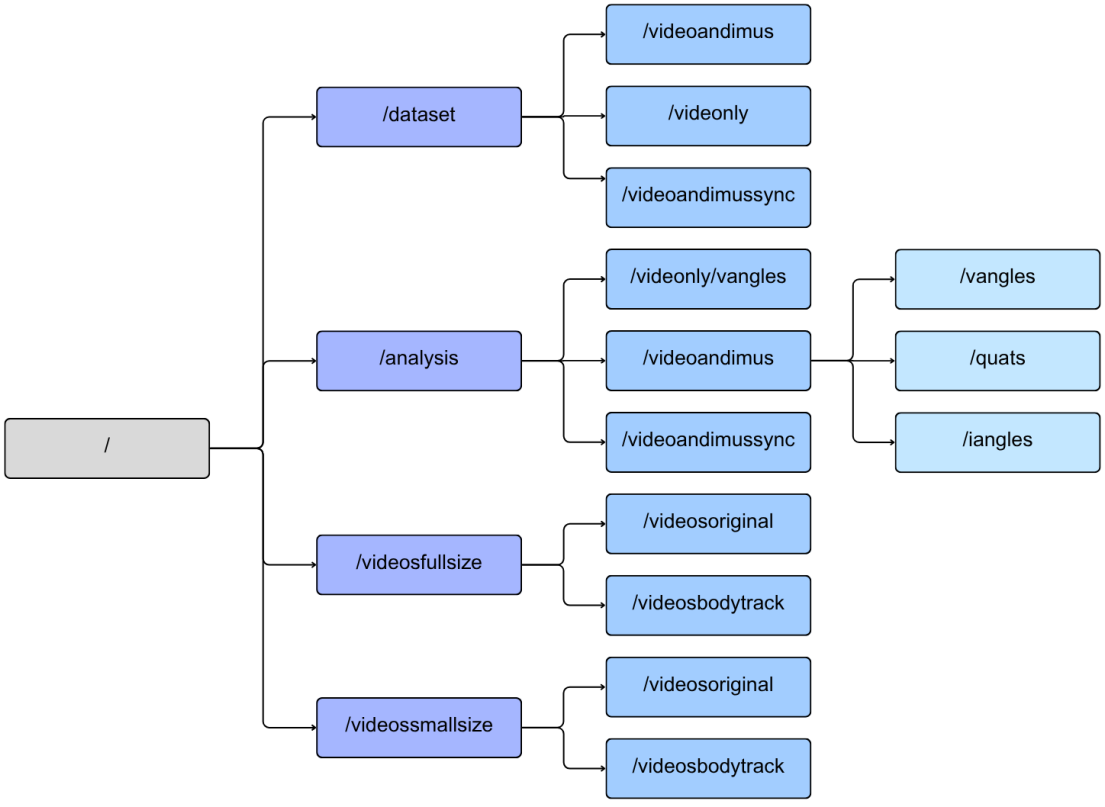


Figura 3.2. Estructura de directorios del dataset VIDIMU.

3.1.2. Protocolo de adquisición

El proceso de adquisición fue llevado a cabo en las instalaciones de la Escuela Técnica de Ingenieros de Telecomunicación de la Universidad de Valladolid, específicamente en la sala Torres Quevedo, en un período comprendido desde junio de 2022 hasta enero de 2023. Antes de comenzar con la actividad, cada sujeto se colocó adoptando una pose neutral (*N-pose*) con el objetivo de grabar las orientaciones iniciales de los IMUs para poder calibrarlos. La Tabla 3.1 resume la información sobre las actividades realizadas, incluyendo las instrucciones orales proporcionadas a los participantes para asegurar movimientos precisos y homogéneos en toda la base de datos.

Cabe destacar que para mejorar el rendimiento de los detectores de pose y evitar oclusiones de unas partes del cuerpo con otras, la configuración de la cámara respecto al sujeto grabado varía ligeramente según la tarea realizada. En las actividades A01 y A02 la marcha fue capturada desde el plano sagital, ya que los sujetos caminaron de forma perpendicular al plano de la cámara. En la actividad A03, los sujetos caminaron por una línea orientada a aproximadamente 20 grados respecto al plano de grabación. En las actividades A04 a A13, los sujetos estaban girados a la izquierda aproximadamente 45 grados respecto a la cámara frontal.

Tabla 3.1. Actividades grabadas e instrucciones proporcionadas a los sujetos durante la adquisición.

ID	Actividad	Instrucciones para el sujeto
A01	*walk_forward*	Sitúese sobre la marca situada en el suelo, gire hacia su derecha y camine de ida y vuelta en línea recta entre las dos marcas. Realice los giros mirando a la cámara y a un ritmo más lento.
A02	*walk_backward*	Sitúese sobre la marca situada en el suelo, gire hacia su izquierda y camine de ida y vuelta en línea recta entre las dos marcas en el suelo. Realice los giros mirando a la cámara y a un ritmo más lento. Puede girar la cabeza para comprobar si alcanza las marcas.
A03	*walk_along*	Sitúese sobre la marca situada en el suelo, gire hacia su derecha y camine de ida y vuelta sobre la línea. Coloque un pie delante del otro como si estuviera manteniendo el equilibrio, pero sin juntarlos. Realice los giros mirando a la cámara y a un ritmo más lento.
A04	*sit_to_stand*	Partiendo de una posición sentada, levántese y siéntese nuevamente, sin apoyarse con los brazos.
A05, A06	*move_right_arm, move_left_arm*	Partiendo de una posición sentada, con las manos sobre las rodillas, extienda la mano derecha/izquierda para coger la botella que está sobre la mesa y llévela hasta la otra marca situada en ella. A continuación, vuelva a colocar la botella en su posición original. Finalmente, regrese a la posición inicial, con las manos sobre las rodillas.
A07, A08	*drink_right_arm, drink_left_arm*	Partiendo de una posición sentada, con las manos sobre las rodillas, extienda la mano derecha/izquierda para coger la botella que está sobre la mesa y simule el gesto de beber acercándola a la boca sin llegar a tocarla. Luego, devuélvala a su posición original sobre la mesa. Finalmente, regrese a la posición inicial, con las manos sobre las rodillas.

A09	*assemble_both_arms*	Partiendo de una posición sentada, con las manos sobre las rodillas, construya una torre encajando 6 piezas: 3 situadas a su derecha y 3 a su izquierda. Comience a montar con la mano dominante y alterne manos. Una vez terminada la construcción, proceda a desmontarla: empiece con la mano dominante y alterne manos para devolver las piezas a las posiciones correspondientes a su derecha e izquierda. Finalmente, regrese a la posición inicial, con las manos sobre las rodillas.
A10	*throw_both_arms*	Sitúese sobre la marca situada en el suelo con los brazos a lo largo del cuerpo, sujetando el balón con ambas manos. Lance el balón hacia arriba, recójalo y vuelva a la posición de partida.
A11, A12	*reachup_right_arm, reachup_left_arm*	Sitúese sobre la marca situada en el suelo con los brazos a lo largo del cuerpo. Extienda el brazo derecho/izquierdo para coger la botella situada en una posición elevada. Una vez que la haya cogido, suéltela y regrese a la posición inicial.
A13	*tear_both_arms*	Sitúese sobre la marca situada en el suelo con los brazos a lo largo del cuerpo, sujetando el papel con ambas manos. Eleve el papel hasta que sus brazos formen un ángulo de 90° con el tronco. Con ambas manos, rompa el papel en 4 trozos. A continuación, forme una bola con ellos y láncela.

3.1.3. Tecnologías y entorno de desarrollo

Para realizar el procesado y llevar a cabo la evaluación, se optó por un entorno de desarrollo basado en Python 3, gestionado mediante entornos virtuales mediante la herramienta Conda para garantizar la reproducibilidad. La extracción y conversión de datos se apoya en varias librerías, dependiendo de la arquitectura elegida. Además, se emplearon Jupyter Notebooks para realizar la inferencia de modelos y el manejo de configuraciones. El tratamiento numérico y la manipulación de datos se efectuó con NumPy y Pandas, mientras que el procesamiento de señales recurrió a funciones de SciPy y rutinas propias. Para el cómputo de métricas estadísticas, se empleó scikit-learn junto a las liberías mencionadas anteriormente. La generación de gráficos y visualizaciones se realizó con Matplotlib, y la gestión de archivos y directorios con los módulos estándar de Python (os, shutil, json, etc.). La depuración y el control de flujo quedaron garantizados mediante el uso de logging y manejo de excepciones (*traceback*), y el cómputo intensivo aprovechó la aceleración por GPU mediante CUDA a través de PyTorch (usado internamente por los scripts de inferencia).

3.1.4. Preprocesado de señales obtenidas a partir de vídeo

A partir de los vídeos adquiridos, la información de todos los modelos HPE incluidos en la evaluación siguió un procedimiento común: los datos fueron procesados utilizando un flujo de trabajo consistente en (i) estimación de los marcadores articulares mediante aprendizaje profundo, (ii) conversión de ficheros a formato CSV (*Comma-Separated Values)*, (iii) cinemática inversa y cálculo de ángulos articulares, (iv) filtrado y sincronización, y (v) carga de los datos y

visualización. A continuación, se describen en detalle cada una de estas etapas, destacando las diferencias particulares para algunas de las arquitecturas.

Cada sistema ejecutado sobre los vídeos presentes en VIDIMU proporciona coordenadas 3D para cada uno de los *frames*, conformando una secuencia temporal que contiene el número de *frame*, los nombres de las 17 articulaciones detectadas, y las coordenadas (x,y,z). De este modo, conociendo la tasa de *frames* por segundo (30 fps para el vídeo), se puede obtener el instante de tiempo para una determinada detección mediante la siguiente expresión:

$$t_i = \frac{frame_i}{\text{fps}} \tag{3.1}$$

En la Figura 3.3 se puede observar la relación entre IDs y articulaciones en el formato de la base de datos Human3.6M, que es el empleado por todos los modelos analizados. En la Figura 3.4 se muestra un ejemplo del procesamiento realizado sobre un fichero de vídeo utilizando elevación de pose 2D a 3D. Cabe mencionar que el modelo NVIDIA Bodytrack genera la información de pose directamente a través de la línea de comandos, por lo que tuvo que ser escrita a ficheros personalizados con extensión *.out*. El resto de los modelos evaluados en este trabajo producen directamente datos en formato JSON (JavaScript Object Notation).

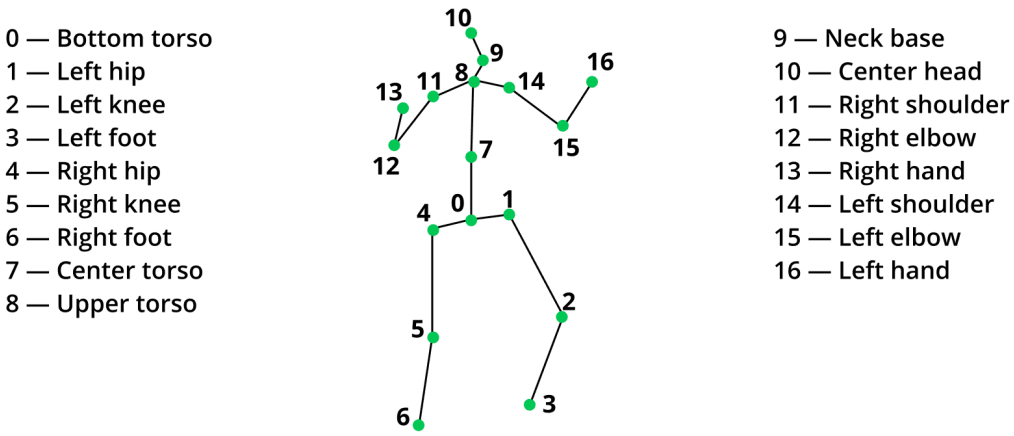


*Figura 3.3. Marcadores en el dataset Human3.6M (*3D Human Pose Estimation Experiments and Analysis*, s. f.).*

A continuación, la información de pose fue transformada a un formato común, concretamente CSV, para facilitar el análisis y la comparación posteriores con los datos extraídos de los sensores inerciales. Cada articulación se etiquetó según una nomenclatura anatómica estandarizada (p. ej. *right_shoulder*, *left_elbow* o *head*) y se emplearon tres columnas para cada una de ellas (una por coordenada). Las filas almacenan el número de *frame* del vídeo. Además, cada uno de los modelos disponía de un *script* personalizado para mapear la nomenclatura nativa del sistema a una nomenclatura común (p. ej. sustituyendo *neck_base* por *neck* o *right_foot* por *right_ankle*), de modo que se pudiera operar con los datos de todas las fuentes de manera sencilla.

Una vez convertidos y normalizados todos los ficheros, se leyeron en un *dataframe* de Pandas, de modo que los siguientes pasos son independientes del modelo empleado y aplicaron el mismo procesamiento a cualquier fichero CSV con información sobre estimación de pose 3D.

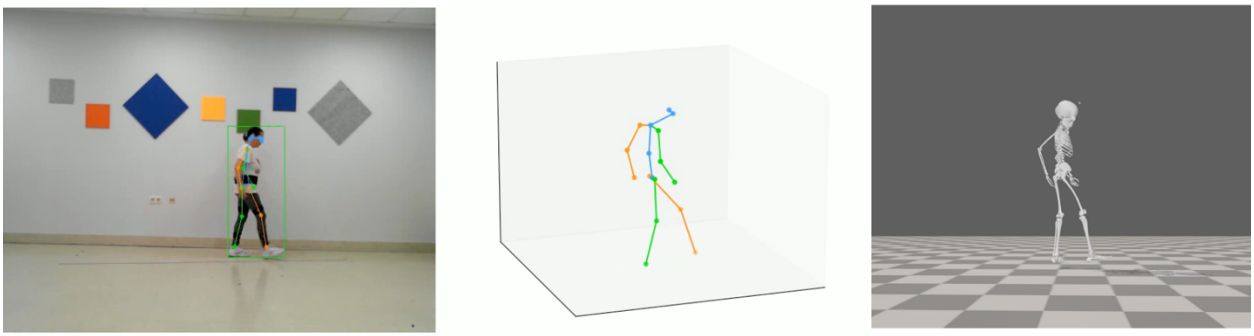


Figura 3.4. Ejemplo de procesamiento para A01 (walk_forward), S40 utilizando MMPose.

Para inferir la cinemática inversa IK (*Inverse Kinematics*) a partir de los marcadores articulares, se llevó a cabo el siguiente procedimiento: en primer lugar, se tomó un subconjunto de marcadores relevantes, que es distinto para cada articulación de interés. Los marcadores específicos empleados para cada uno de los ángulos extraídos pueden consultarse en la Tabla 3.2. Además, el ángulo calculado varía según la actividad (ver Tabla 3.3). Luego, se construyeron dos vectores equivalentes a los dos huesos relativos a la articulación (p. ej. *upper_arm* y *forearm* para el codo). De este modo, se calculó el ángulo entre estos vectores empleando el producto escalar y la función arco coseno, según la expresión:

$$\theta(t) = \arccos\left(\frac{v_1(t) \cdot v_2(t)}{||v_1(t)|||v_2(t)||}\right) \times \frac{180}{\pi} \tag{3.2}$$

Tabla 3.2. Marcadores empleados para el cálculo de cada ángulo.

Ángulo	Marcador 1	Marcador 2	Marcador 3	Marcador 4
arm_flex_r	right_shoulder	right_elbow	neck	torso
arm_flex_l	left_shoulder	left_elbow	neck	torso
elbow_flex_r	right_shoulder	right_elbow	right_elbow	right_wrist
elbow_flex_l	left_shoulder	left_elbow	left_elbow	left_wrist
knee_angle_r	right_hip	right_knee	right_knee	right_ankle
knee_angle_l	left_hip	left_knee	left_knee	left_ankle

Tabla 3.3. Ángulo evaluado en cada actividad.

Actividad	**Ángulo empleado**
A01, A03	knee_angle_l
A02, A04	knee_angle_r
A06	elbow_flex_l
A05, A09	elbow_flex_r
A08, A12	arm_flex_l
A07, A10, A11, A13	arm_flex_r

Debido a que la estimación de pose en tres dimensiones a la salida de los modelos de aprendizaje profundo a menudo contiene valores faltantes o ruido, se realizaron varios pasos de procesamiento para mejorar la señal. En primer lugar, se realizó una interpolación lineal para rellenar los huecos o valores no válidos, asegurando así señales de ángulos continuas:

$$A(t) = A(t_i) + \big(A(t_{i+1}) - A(t_i)\big)\frac{t - t_i}{t_{i+1} - t_i} \tag{3.3}$$

Luego, se aplicó un filtro de media móvil con una ventana de tamaño $N = 11$ para mitigar el ruido en las altas frecuencias y el *jitter* entre *frames*, cuya expresión es la siguiente:

$$y[n] = \frac{1}{N}\sum_{k=0}^{N-1} x[n+k] \tag{3.4}$$

Además, se utilizó un filtro de mediana de tamaño $N = 5$ con el objetivo de suavizar las señales antes de graficarlas y eliminar posibles picos:

$$y[n] = \text{median}\{x\left[n - \frac{N-1}{2}\right], \dots, x\left[n + \frac{N-1}{2}\right]\} \tag{3.5}$$

Posteriormente, se aplicó un procedimiento para centrar las señales respecto a la media, es decir, para normalizarlas en el eje vertical. Para ello, se sustrajo la media muestral de las primeras M muestras, con $M = RSME\ Samples + Max\ Sync\ Overlap$, de modo que se anulase la componente DC:

$$\bar{x} = \frac{1}{M}\sum_{i=0}^{M-1} x[i], \qquad y[n] = x[n] - \bar{x} \tag{3.6}$$

Como nuestro escenario implicó comparar los ángulos articulares derivados del vídeo con aquellos obtenidos mediante los IMUs, que son considerados como la referencia, un paso adicional de sincronización fue requerido. Las señales se desplazaron en el eje horizontal y se truncaron para alinearlas con las señales de los IMUs. Para ello, dentro de una ventana fija de 180 muestras (equivalentes a los 6 primeros segundos del vídeo) se calculó el RMSE entre ambas señales y se

desplazaron el número de muestras necesario hasta minimizar dicho error. En la expresión analítica, esto se traduce en encontrar el valor k que minimice el error:

$$\mathrm{RMSE}(k) = \sqrt{\frac{1}{L}\sum_{n=1}^{L}(x[n] - y[n+k])^2}\,, \quad k^* = \arg\min_k \mathrm{RMSE}\,(k) \tag{3.7}$$

Una vez normalizadas y sincronizadas todas las señales, el flujo de trabajo diseñado también incorporó un módulo de visualización para generar gráficos tanto de las señales en crudo (ver Figura 3.5) como de las señales procesadas en cada paso (ver Figura 3.6). Adicionalmente, se compararon las señales angulares sincronizadas derivadas de vídeo y de IMUs superponiendo todas las fuentes de datos en gráficos comunes. Estas visualizaciones (en formatos SVG y PDF) permitieron una evaluación cualitativa de las etapas de filtrado y sincronización. Todos los ficheros generados están disponibles en un repositorio de Zenodo (Medrano-Paredes et al., 2025). Esta fase sirvió como medida de control de calidad para identificar posibles errores de adquisición o de procesamiento y realizar correcciones rápidas en el flujo de trabajo.

A través de este procedimiento estandarizado, se garantizó que los cuatro métodos de HPE estudiados (BodyTrack, MotionBERT, MotionAGFormer y VideoPose3D) produjesen trayectorias angulares temporal y espacialmente comparables para nuestros posteriores pasos de *benchmarking* y evaluación.

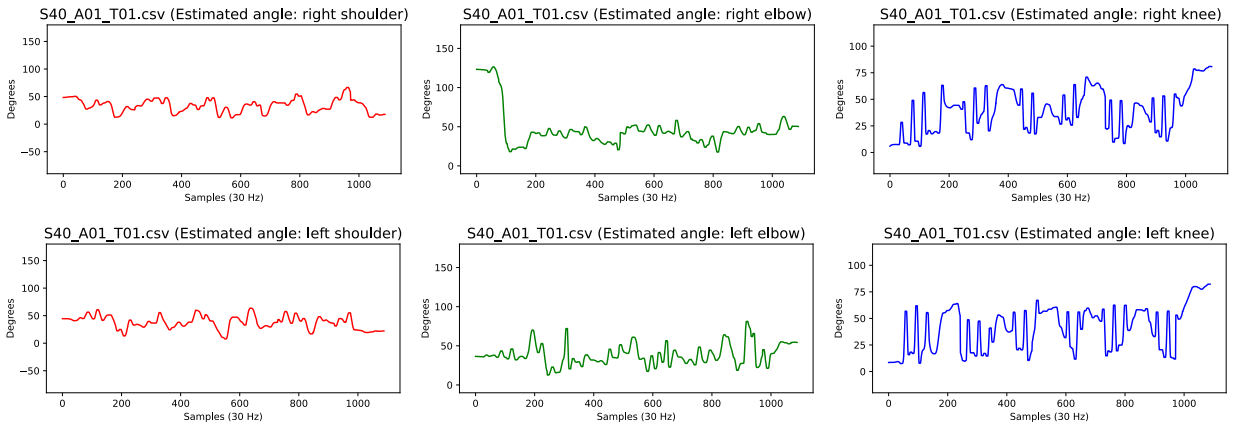


Figura 3.5. Ángulos estimados para la actividad A01 (walk_forward), sujeto S40 empleando MotionAGFormer.

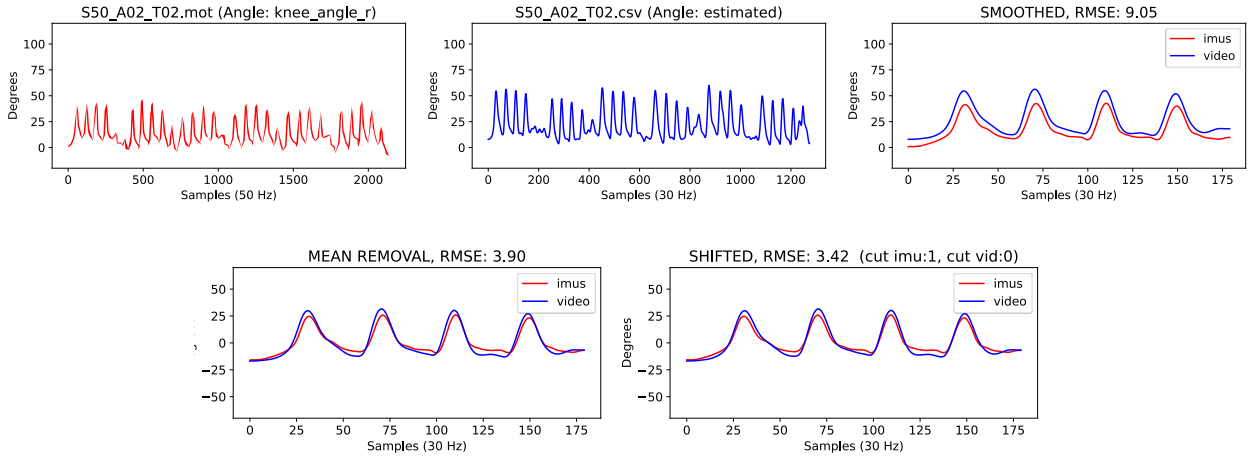


Figura 3.6. Etapas de procesado de las señales de ángulos para la actividad A02 (walk_backward), sujeto S50 empleando MotionAGFormer.

3.1.5. Preprocesado de señales de obtenidas a partir de sensores inerciales

Los datos en crudo de los sensores se adquirieron empleando cinco IMUs personalizados y se almacenaron como cuaterniones con una tasa de 50 Hz en ficheros *.raw*. Un cuaternión es un número hipercomplejo de cuatro componentes que se emplea para representar rotaciones en el espacio tridimensional de forma eficiente y sin ambigüedades de tipo *gimbal lock*. Se suele escribir como:

$$q = \mathrm{w} + \mathrm{x}i + \mathrm{y}j + \mathrm{z}k \tag{3.8}$$

donde w es la parte escalar, (x, y, z) es la parte vectorial y (i, j, k) son las unidades imaginarias que satisfacen $i^2 = j^2 = k^2 = ijk = -1$. En el contexto de IMUs, los cuaterniones permiten fusionar lecturas de giroscopio y acelerómetro para seguir la orientación del sensor sin las singularidades ni discontinuidades que presentan otros métodos (como los ángulos de Euler). En la Figura 3.7 se puede observar la representación gráfica de los cuaterniones obtenidos a través de los datos de los sensores inerciales. Antes de comenzar cada registro, los sensores se calibraron en la N-pose y los cuaterniones se exportaron también a formato *.sto* para poder procesarlos en OpenSim. Esta herramienta permitió ejecutar el algoritmo de cinemática inversa (IK) sobre un modelo musculoesquelético completo, concretamente una versión modificada de (Rajagopal et al., 2016), y generó ficheros *.mot* con los ángulos articulares resultantes, sin necesidad de implementar código propio para el cálculo de dichas variables.

Una vez obtenidos los ángulos derivados de los IMUs, el flujo siguió un procedimiento similar a aquel empleado para las señales derivadas del vídeo, consistente en: (i) extracción de datos, (ii) conversión a formato CSV, (ii) filtrado y remuestreo de la señal, (iii) sincronización temporal, y (iv) registro y visualización.

En primer lugar, las señales de los sensores (50 Hz) se remuestrearon a 30 Hz para igualar la frecuencia del vídeo. A partir de aquí, las operaciones calculadas fueron las mismas que en el caso anterior. Se realizó una interpolación para evitar valores faltantes y garantizar trayectorias continuas. Para mitigar el ruido de alta frecuencia, se aplicó un filtro de media móvil con $N = 11$. Luego, se aplicó un filtro de mediana con una ventana de $N = 5$ muestras para suavizar la señal y mejorar la representación gráfica. Finalmente, las señales se sincronizaron mediante un algoritmo iterativo de minimización del RMSE y se recortaron a una longitud común máxima. Una muestra de los ángulos obtenidos en esta etapa puede verse en la Figura 3.8.

Del mismo modo que con las señales de vídeo, el flujo de trabajo incluyó la visualización tanto de los datos en crudo como de cada etapa del procesamiento. Además, las curvas de los ángulos procedentes de IMU y las obtenidas con cada modelo de estimación 3D se superpusieron (ver Figura 3.9) para realizar una primera verificación visual de la alineación y la correcta sincronización de los datos.

Este flujo de procesado sustentó el análisis comparativo entre modelos de aprendizaje profundo basados en vídeo y las mediciones de referencia basadas en IMU, y su diseño modular garantizó flexibilidad y extensibilidad para el tratamiento de datos de movimiento multimodal.

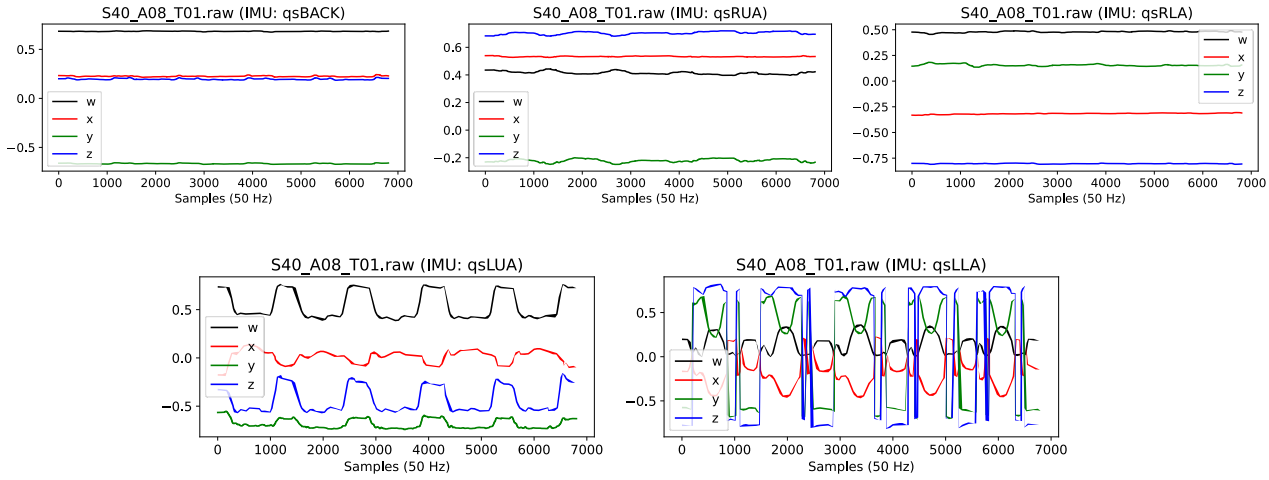


Figura 3.7. Datos en crudo de los cuaterniones obtenidos por los IMUs para la actividad A08 (drink_left_arm), sujeto S40.

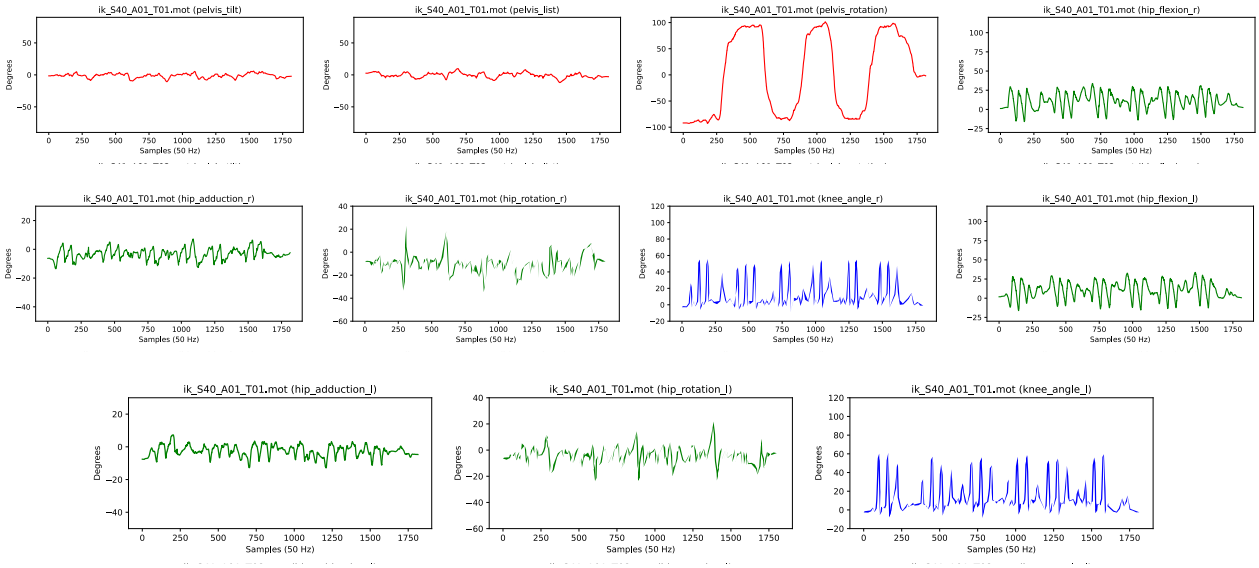


Figura 3.8. Ángulos estimados para la actividad A01 (walk_forward), sujeto S40 empleando IMUs.

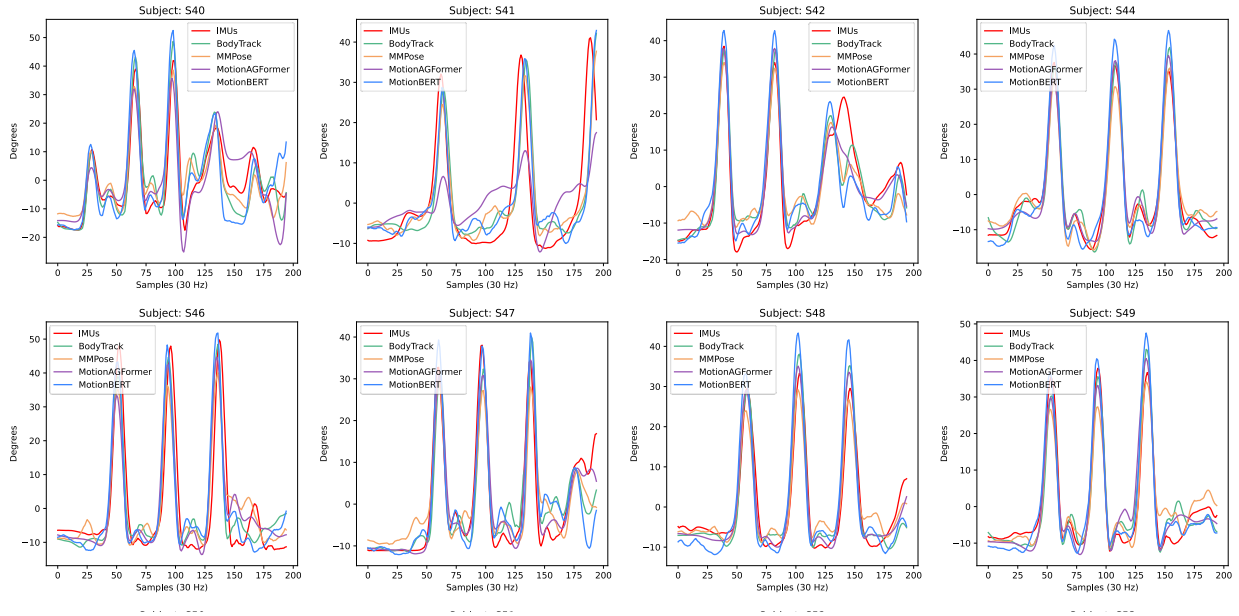


Figura 3.9. Señales procesadas por cada sistema para el ángulo knee_angle_l, actividad A03 (walk_along), 8 primeros sujetos.

3.1.6. Métricas de evaluación y análisis estadístico

El rendimiento de los distintos modelos de aprendizaje profundo para 3D HPE se evaluó comparando sus predicciones con los ángulos articulares derivados de las IMUs, considerados como *ground truth*. Se calcularon múltiples métricas para cada sujeto, que posteriormente se

agregaron por actividad y se promediaron sobre todo el conjunto de datos. A continuación, se detalla la definición de cada métrica y las expresiones para calcularlas.

Raíz del Error Cuadrático Medio (RMSE)

La métrica principal para calcular la precisión general de cada modelo de visión artificial respecto a los sensores inerciales fue la raíz cuadrada del error cuadrático medio, que mide la desviación típica de los errores. El RMSE cuantifica el error cuadrático medio entre la señal estimada y la de referencia, y se expresa en los mismos grados que los ángulos articulares obtenidos previamente. Al elevar las diferencias al cuadrado, esta métrica penaliza con mayor peso los errores grandes, por lo que es muy sensible a valores atípicos (*outliers*). Debido a esta sensibilidad, sirve como un indicador conservador de la precisión global del modelo, de modo que valores bajos del RMSE implican mayor exactitud. Su expresión es la siguiente:

$$\mathrm{RMSE} = \sqrt{\frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathrm{y_i} - \widehat{\mathrm{y}_\mathrm{i}})^2} \tag{3.9}$$

donde y_i son los valores de referencia, $\widehat{y_i}$ son los valores estimados, y N es el número de muestras.

RMSE Normalizado (NRMSE)

Para poder comparar errores entre articulaciones o actividades con rangos de movimiento distintos, el RMSE se normalizó dividiéndolo por el rango o amplitud o, cuando se especifica, por la desviación típica de la señal de referencia. Esta métrica adimensional cuantifica el error relativo y facilita la comparación del desempeño a través de distintas poses y actividades con rangos variables de movimientos articulares tridimensionales. Al igual que en el caso anterior, valores cercanos a 0 equivalen a un buen ajuste. Se define como:

$$\mathrm{NRMSE} = \frac{\mathrm{RMSE}}{\mathrm{Range(y)}} \tag{3.10}$$

donde $Range(y)$ es la diferencia entre el valor máximo y el valor mínimo de la señal de referencia.

Error Absoluto Medio (MAE)

El error absoluto medio es la media del valor absoluto de las diferencias, es decir, mide el promedio del error en valor absoluto. A diferencia del RMSE, en el MAE todas las discrepancias o errores se ponderan por igual (todos los errores tienen el mismo peso), de modo que resulta más robusto frente a unos pocos errores extremos (*outliers*). Su interpretación directa, n grados de error

promedio, lo hace especialmente intuitivo. Se calcula mediante la fórmula presentada a continuación:

$$\mathrm{MAE} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}} |\mathrm{y_i} - \widehat{\mathrm{y}_\mathrm{i}}| \tag{3.11}$$

donde y_i son los valores de referencia, $\widehat{y_i}$ son los valores estimados, y N es el número total de muestras.

Coeficiente de correlación de Pearson

El coeficiente de correlación mide la fuerza y la dirección de la relación lineal entre la señal estimada (ángulos articulares detectados a partir del vídeo) y la de referencia (obtenida mediante IK en OpenSim gracias a los IMUs). Su valor se encuentra entre -1 (correlación negativa perfecta) y +1 (correlación positiva perfecta). Valor 0 indica ausencia de relación lineal. Un alto r no implica necesariamente baja magnitud de error, pero sí que ambas señales varían de forma concordante. Se calcula como:

$$\mathrm{r} = \frac{\sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathrm{y_i} - \bar{\mathrm{y}})\left(\widehat{\mathrm{y}_\mathrm{i}} - \bar{\widehat{\mathrm{y}}}\right)}{\sqrt{\sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathrm{y_i} - \bar{\mathrm{y}})^2 \sum_{\mathrm{i}=1}^{\mathrm{N}}\left(\widehat{\mathrm{y}_\mathrm{i}} - \bar{\widehat{\mathrm{y}}}\right)^2}} \tag{3.12}$$

donde $\bar{y}$ y $\bar{\hat{y}}$ son los valores medio de las señales derivadas de los sensores y del vídeo respectivamente.

Coeficiente de determinación

El coeficiente de determinación se utilizó para evaluar la proporción de la varianza en los datos reales (variable dependiente) que es predecible o explicada por los datos del modelo (variable independiente). Un R^2 cercano a 1 indica que casi toda la variabilidad es explicada por la estimación realizada (el modelo describe mejor la variabilidad intermuestra). Valores próximos a 0 sugieren que el modelo no mejora la predicción frente a usar simplemente la media de los datos observados. Los valores negativos indican que el modelo funciona peor que la predicción de la media. Su expresión es:

$$\mathrm{R^2} = 1 - \frac{\sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathrm{y_i} - \widehat{\mathrm{y}_\mathrm{i}})^2}{\sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathrm{y_i} - \bar{\mathrm{y}})^2} \tag{3.13}$$

donde $\bar{y}$ y $\bar{\hat{y}}$ son los valores medio de las señales derivadas de los sensores y del vídeo respectivamente.

3.1.7. Realización del *benchmark*

Una vez definidas las métricas de evaluación, se desarrolló un sistema de procesamiento por lotes para realizar el cálculo de forma automatizada. Para ello, un script recorrió para cada arquitectura de HPE en 3D, todas las actividades y todos los sujetos disponibles. En cada iteración, se cargaron simultáneamente el archivo *.mot* con los ángulos articulares de los sensores inerciales y el *.csv* correspondiente a cada modelo de aprendizaje profundo. Tras el preprocesado y la sincronización, se creó un diccionario con todas las métricas para esa combinación modelo-sujeto-actividad. Dichos diccionarios se volcaron en un *dataframe* de Pandas.

Para cada actividad, se generó una tabla que presenta la media y la desviación típica de cada métrica y modelo. Estas tablas se exportaron a CSV y se renderizaron también como figuras vectoriales (SVG) y como PDF para facilitar su inclusión posterior en documentos de texto. La visualización se completó añadiendo gráficos de barras para realizar un análisis de la precisión sujeto a sujeto. A nivel global, se incorporó un *radar plot* superpuesto con todos los parámetros de evaluación normalizados, así como tablas indicando los resultados agregados totales, los resultados divididos por actividad y por métrica, y otro gráfico de barras con los promedios. Este análisis hizo un uso eficiente de los parámetros calculados, permitiendo evaluar de forma consistente el rendimiento de cada arquitectura, tanto en tareas específicas como en el comportamiento general del conjunto de datos. La distribución de ficheros y directorios con los resultados finales del *benchmark* puede consultarse en la Tabla 3.4.

Tabla 3.4. Estructura de directorios final y formatos de los ficheros obtenidos.

Subcarpeta	**Detalles**
/pose3d	Cada directorio contiene una subcarpeta para cada sujeto, y un fichero por cada actividad e intento.
/pose3d/motionagformer	**.json*: Marcadores inferidos para las articulaciones.
/pose3d/motionbert	**.json*: Marcadores inferidos para las articulaciones.
/pose3d/mmpose	**.json*: Marcadores inferidos para las articulaciones.
/pose3d/bodytrack	**.out*: Ficheros de texto plano con los marcadores inferidos para las articulaciones.
/jointangles	Cada directorio contiene una subcarpeta para cada sujeto, y un fichero por cada actividad e intento.
/jointangles/imus	**.raw*: Grabaciones originales de cuaterniones con los IMUs. **.sto*: Ficheros tabulares con las grabaciones originales de cuaterniones y ficheros tabulares con los errores de orientación generados mediante IK. **.mot*: Ficheros tabulares con los ángulos articulares generados mediante IK.

/jointangles/motionagformer	**.csv*: Ángulos articulares calculados a partir de vídeo.
/jointangles/motionbert	**.csv*: Ángulos articulares calculados a partir de vídeo.
/jointangles/mmpose	**.csv*: Ángulos articulares calculados a partir de vídeo.
/jointangles/bodytrack	**.csv*: Ángulos articulares calculados a partir de vídeo.
/analysis	Cada directorio contiene un fichero por cada actividad e intento.
/analysis/motionagformer/jointangles	**.svg*: Gráficos de los ángulos articulares calculados a partir de la estimación de pose.
/analysis/motionagformer/synchronized	**.svg*: Gráficos de las señales de ángulos de vídeo y de sensores sincronizadas y RMSE.
/analysis/motionbert/jointangles	**.svg*: Gráficos de los ángulos articulares calculados a partir de la estimación de pose.
/analysis/motionbert/synchronized	**.svg*: Gráficos de las señales de ángulos de vídeo y de sensores sincronizadas y RMSE.
/analysis/mmpose/jointangles	**.svg*: Gráficos de los ángulos articulares calculados a partir de la estimación de pose.
/analysis/mmpose/synchronized	**.svg*: Gráficos de las señales de ángulos de vídeo y de sensores sincronizadas y RMSE.
/analysis/bodytrack/jointangles	**.svg*: Gráficos de los ángulos articulares calculados a partir de la estimación de pose.
/analysis/bodytrack/synchronized	**.svg*: Gráficos de las señales de ángulos de vídeo y de sensores sincronizadas y RMSE.
/results	Cada directorio contiene un fichero por cada actividad e intento.
/results/comparison	**.svg*: Gráficos de las señales de ángulos superpuestas para todos los modelos y para los sensores.
/results/evaluation	**.csv*: Tablas con los resultados de evaluación. **.svg*: Tablas y gráficos con los resultados de la evaluación.

3.2. Resultados

Los resultados recogen los valores medios ± desviación típica de RMSE, MAE, coeficiente de correlación de Pearson (r) y coeficiente de determinación (R^2) para cada modelo y tarea. Dentro de cada actividad, la evaluación se realizó también para cada uno de los pacientes del *dataset*. De este modo, el rendimiento de los modelos fue evaluado tanto de manera holística como poniendo el foco en actividades específicas orientadas a la movilidad en distintas partes del cuerpo.

3.2.1. Resultados por actividad

Para evaluar el impacto que la naturaleza del movimiento, las autooclusiones (donde una parte del cuerpo tapa otra respecto al plano de cámara) y el entorno tiene sobre la exactitud de la estimación 3D, las métricas se desglosaron por actividad (A01-A13). En la Tabla 3.5 y la Tabla 3.6 se puede observar una muestra de los resultados para la raíz del error cuadrático medio y la correlación, aunque el análisis comprende datos de todas las métricas descritas anteriormente.

En las tareas de extremidades inferiores, caracterizadas por trayectorias relativamente regulares y pocas oclusiones, los errores fueron sistemáticamente menores. Este resultado inicial tiene implicaciones especialmente relevantes para aplicaciones directas de la estimación de pose humana, como por ejemplo la segmentación de la marcha. Por ejemplo, en walk_forward (A01) MMPose obtuvo el RMSE y el MAE más bajos con 6.54° ± 2.64° y 4.90 ± 1.75 respectivamente, y el mayor R^2 0.76 ± 0.20, mientras que en walk_backward (A02) el liderazgo correspondió a BodyTrack con RMSE 4.72° ± 1.40° y un r de 0.92 ± 0.04. Durante walk_along (A03) MotionAGFormer alcanzó la mejor precisión con RMSE: 5.98° ± 2.92°, r: 0.90 ± 0.08. La transición sit_to_stand (A04) resultó más exigente: MotionBERT destacó con un RMSE de 5.75° ± 2.04° y un R^2 de 0.95 ± 0.04, superando en varianza explicada al resto de sistemas.

Las actividades de extremidades superiores mostraron diferencias más pronunciadas entre modelos a causa de movimientos más rápidos, amplitudes variables y oclusiones frecuentes. MotionAGFormer fue el más consistente: presentó los menores errores en move_right_arm (A05) RMSE: 6.14° ± 2.64°, MAE: 5.00 ± 2.25°, R^2: 0.79 ± 0.16, en drink_right_arm (A07) RMSE: 5.58° ± 2.86°, MAE: 4.64 ± 2.54°, r: 0.98 ± 0.02 y en gestos bilaterales como throw_both_arms (A10) RMSE: 9.74° ± 9.09°. La tarea move_left_arm (A06) concentró los mayores errores absolutos de todo el conjunto (hasta aproximadamente 34° de RMSE y 30° de MAE con MotionBERT) pero MotionAGFormer siguió siendo la opción menos alejada del *ground-truth*. En drink_left_arm (A08) BodyTrack alcanzó la correlación más alta r: 0.97 ± 0.02 con un error similar al de los otros métodos. Las tareas de coordinación fina assemble_both_arms (A09) y reach-up_right_arm (A11) volvieron a situar a MMPose y a MotionAGFormer, respectivamente, como las alternativas más fiables. Por su parte, MotionBERT sobresalió en los gestos altos del brazo izquierdo: obtuvo el mínimo RMSE en reach-up_left_arm (A12) RMSE: 6.22° ± 2.43° y el máximo R^2: 0.98 ± 0.02. Finalmente, en tear_both_arms (A13) MotionAGFormer fue de nuevo el más preciso con RMSE: 4.99° ± 2.48°, r: 0.98 ± 0.02.

Aunque todos los modelos proporcionaron estimaciones clínicamente utilizables, el análisis por actividad revela que su rendimiento es sensible al tipo de gesto y al segmento corporal implicado, información crucial a la hora de seleccionar la arquitectura más adecuada para aplicaciones específicas.

Tabla 3.5. RMSE por actividad para cada modelo evaluado.

ID	Descripción	MotionAGFormer	MotionBERT	MMPose	BodyTrack
A01	*walk_forward*	8.12 ± 3.42	7.40 ± 2.56	6.54 ± 2.64	7.07 ± 2.73
A02	*walk_backward*	6.34 ± 2.33	6.55 ± 2.18	5.74 ± 1.96	4.72 ± 1.40
A03	*walk_along*	5.66 ± 2.72	7.09 ± 2.54	6.25 ± 2.61	6.02 ± 2.26
A04	*sit_to_stand*	12.59 ± 7.71	5.71 ± 2.10	9.53 ± 2.02	6.99 ± 2.00
A05	*move_right_arm*	6.14 ± 2.64	13.47 ± 3.90	10.19 ± 2.70	13.02 ± 3.02
A06	*move_left_arm*	18.22 ± 8.63	33.64 ± 7.45	19.29 ± 6.37	25.70 ± 8.69
A07	*drink_right_arm*	5.58 ± 2.86	8.92 ± 3.91	8.23 ± 3.83	15.04 ± 3.49
A08	*drink_left_arm*	8.94 ± 3.21	8.39 ± 3.14	8.49 ± 2.62	8.40 ± 2.97
A09	*assemble_both_arms*	6.94 ± 5.38	10.75 ± 3.57	6.53 ± 5.38	10.15 ± 4.01
A10	*throw_both_arms*	9.80 ± 9.38	19.49 ± 13.45	17.46 ± 12.86	13.01 ± 8.76
A11	*reachup_right_arm*	9.90 ± 4.54	15.24 ± 6.05	19.01 ± 4.78	14.41 ± 3.20
A12	*reachup_left_arm*	17.28 ± 7.04	6.22 ± 2.43	8.75 ± 2.86	9.49 ± 2.68
A13	*tear_both_arms*	4.99 ± 2.48	16.79 ± 6.42	17.57 ± 3.53	7.61 ± 2.55

Tabla 3.6. Coeficiente de correlación por actividad para cada modelo evaluado.

ID	Descripción	MotionAGFormer	MotionBERT	MMPose	BodyTrack
A01	*walk_forward*	0.83 ± 0.12	0.87 ± 0.10	0.88 ± 0.11	0.87 ± 0.12
A02	*walk_backward*	0.87 ± 0.11	0.80 ± 0.14	0.86 ± 0.10	0.92 ± 0.04
A03	*walk_along*	0.92 ± 0.06	0.89 ± 0.07	0.90 ± 0.06	0.91 ± 0.05
A04	*sit_to_stand*	0.86 ± 0.33	0.98 ± 0.02	0.96 ± 0.03	0.97 ± 0.02
A05	*move_right_arm*	0.92 ± 0.06	0.79 ± 0.11	0.87 ± 0.06	0.64 ± 0.19
A06	*move_left_arm*	0.20 ± 0.48	-0.22 ± 0.26	-0.04 ± 0.36	-0.17 ± 0.36
A07	*drink_right_arm*	0.98 ± 0.02	0.93 ± 0.07	0.97 ± 0.03	0.74 ± 0.21
A08	*drink_left_arm*	0.91 ± 0.13	0.94 ± 0.06	0.93 ± 0.05	0.97 ± 0.02
A09	*assemble_both_arms*	0.83 ± 0.18	0.61 ± 0.18	0.87 ± 0.10	0.59 ± 0.21
A10	*throw_both_arms*	0.89 ± 0.31	0.80 ± 0.28	0.88 ± 0.26	0.86 ± 0.25
A11	*reachup_right_arm*	0.99 ± 0.01	0.97 ± 0.02	0.97 ± 0.02	0.94 ± 0.02
A12	*reachup_left_arm*	0.96 ± 0.07	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01
A13	*tear_both_arms*	0.98 ± 0.02	0.88 ± 0.10	0.93 ± 0.05	0.94 ± 0.06

Tabla 3.7. Número de actividades donde cada modelo ha superado al resto en una métrica determinada.

Modelo	Número de veces con la mejor métrica				
	RMSE	**MAE**	**r**	**R^2**	**Total**
MotionAGFormer	7	7	7	6	27
MotionBERT	3	2	2	2	9
MMPose	2	3	2	2	9
BodyTrack	1	1	2	3	7

3.2.2. Resultados globales

Al promediar las métricas obtenidas en las 13 actividades y los 16 sujetos se evidencian diferencias notables entre las cuatro arquitecturas evaluadas. La Tabla 3.8 y la Figura 3.10 muestran que MotionAGFormer fue, de manera consistente, el sistema más cercano al *ground truth,* ya que registró el RMSE medio más bajo con 9.27° ± 4.80° y el menor MAE 7.86° ± 4.18°, a la vez que alcanzó el coeficiente de correlación de Pearson más alto 0.86 ± 0.15 y el mayor coeficiente de determinación R^2: 0.67 ± 0.28. Estos valores apuntan a un equilibrio óptimo entre precisión absoluta y capacidad para reproducir la variabilidad del movimiento humano.

En el extremo opuesto, MotionBERT fue la solución con mayor discrepancia respecto a los IMUs: presentó el RMSE más elevado 12.28° ± 4.59° y también el MAE más alto 10.15° ± 3.86°. Además, su R^2: 0.16 ± 0.50 indica que esta arquitectura explica una fracción sustancialmente menor de la varianza respecto al resto, lo que traduce una menor robustez cuando se combinan tareas heterogéneas.

Los resultados de BodyTrack y MMPose se sitúan en un término medio. El primero obtuvo un RMSE: 10.89° ± 3.67° y un MAE: 9.00° ± 3.12°, con valores moderados de r: 0.78 ± 0.12 y R^2: 0.44 ± 0.31. MMPose, por su parte, alcanzó un RMSE de 11.05° ± 4.17° y un MAE de 9.35° ± 3.61°, pero mejoró a BodyTrack en correlación 0.84 ± 0.10 y en el coeficiente de determinación 0.58 ± 0.26.

Estos resultados destacan la viabilidad de la HPE 3D basada en vídeo como alternativa escalable a los sensores inerciales, especialmente para aplicaciones que priorizan la accesibilidad y la eficiencia de costes. Aunque todos los métodos evaluados son viables para estimar la pose humana en 3D en entornos de la vida cotidiana, los resultados ponen de manifiesto claras desventajas en ciertas situaciones prácticas. Por lo tanto, la elección del modelo debe alinearse con los requisitos clínicos específicos, teniendo en cuenta que algunos de los sistemas mostraron mejores resultados en las actividades relativas a las articulaciones inferiores, mientras que otros obtuvieron mejores resultados en actividades más generales y holísticas.

Tabla 3.8. Métricas globales de evaluación.

Modelo	RMSE	MAE	r	R^2
BodyTrack	10.89 ± 3.67	9.0 ± 3.12	0.78 ± 0.12	0.44 ± 0.31
MMPose	11.04 ± 4.17	9.35 ± 3.61	0.84 ± 0.1	0.58 ± 0.26
MotionAGFormer	9.27 ± 4.8	7.86 ± 4.18	0.86 ± 0.15	0.67 ± 0.28
MotionBERT	12.28 ± 4.59	10.15 ± 3.86	0.79 ± 0.11	0.16 ± 0.5

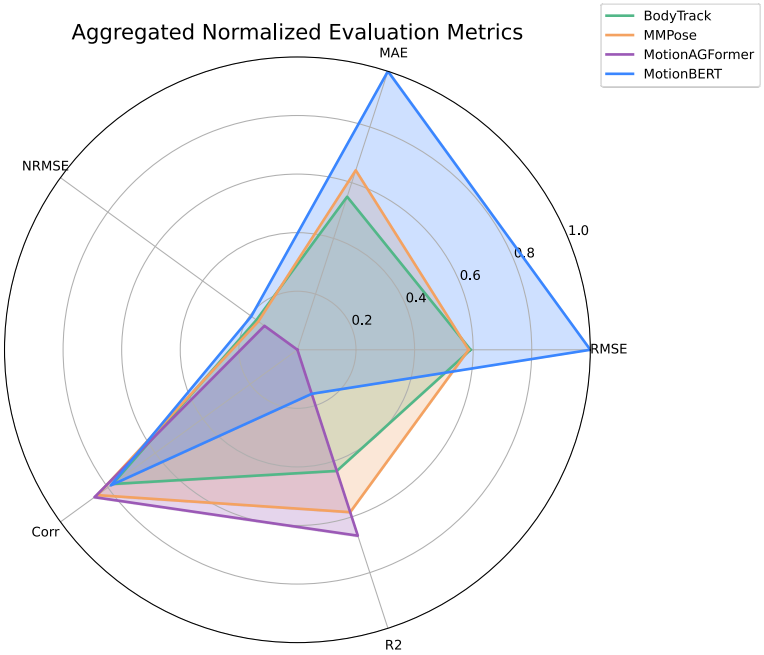


Figura 3.10. Métricas agregadas y normalizadas entre [0,1] para todos los sujetos y actividades.

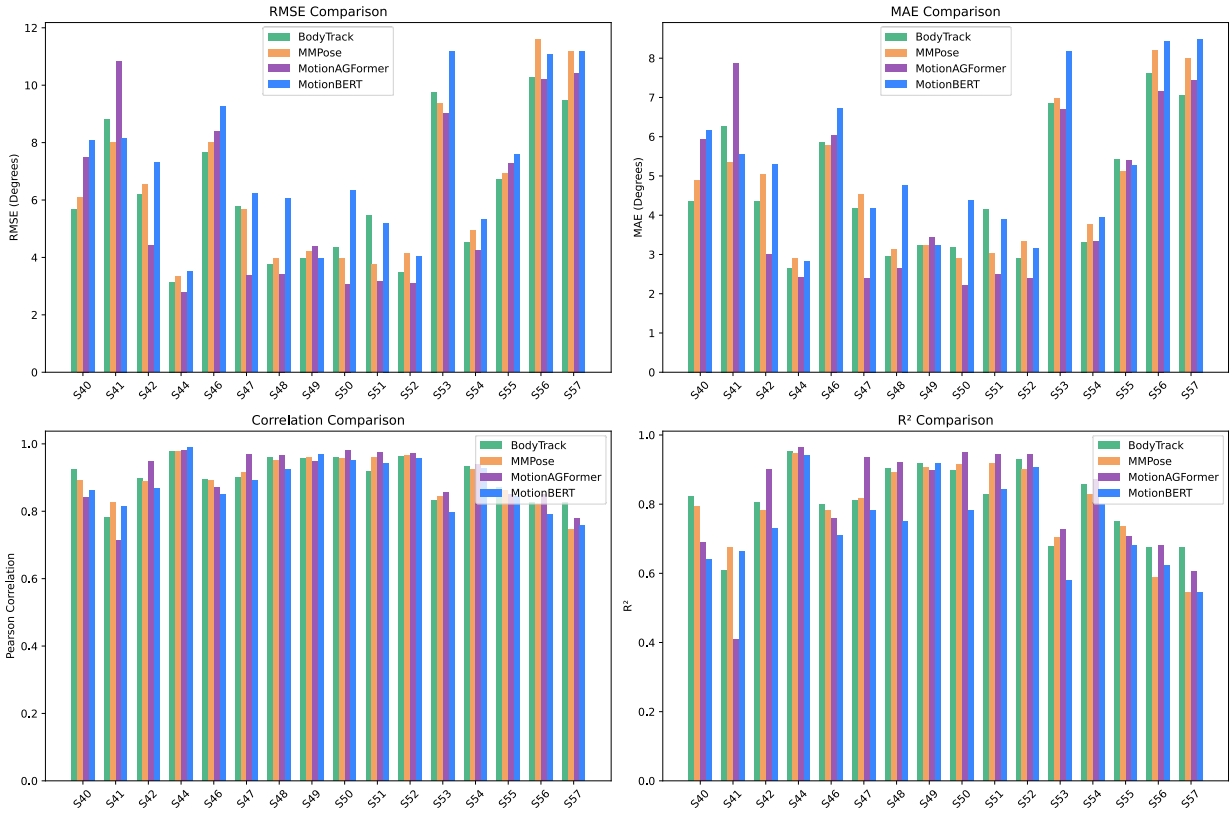


Figura 3.11. Métricas de evaluación por paciente y modelo para A03 (walk_along).

3.3. Discusión y limitaciones

3.3.1. Discusión

El análisis comparativo revela que cada modelo de HPE 3D exhibe fortalezas y limitaciones distintivas, que están estrechamente ligadas a sus arquitecturas subyacentes, paradigmas de entrenamiento y aplicaciones.

MotionAGFormer emergió como el modelo con mejor rendimiento general (RMSE: 9.27° ± 4.80° y R^2: 0.67 ± 0.28), logrando consistentemente errores más bajos, y la correlación y determinación más altas en varias actividades. Su arquitectura híbrida *dual-stream*, que integra transformers para dependencias espaciotemporales globales con Redes Convolucionales de Grafos (GCNs) para capturar relaciones locales entre articulaciones, permite una fusión adaptativa de características críticas tanto para detectar movimientos complejos como para capturar dependencias articulares detalladas de manera más efectiva. Por ejemplo, durante actividades bimanuales, su diseño de doble flujo resuelve eficientemente oclusiones y cambios rápidos de movimiento, mientras que las predicciones a nivel de secuencia aseguran la suavidad temporal. Este diseño también proporciona una representación más equilibrada de la dinámica espaciotemporal, lo cual es crítico en escenarios cotidianos no restringidos. Cabe mencionar que,

aunque es el sistema más preciso, MotionAGFormer también es el que más tiempo tarda en inferir, lo que puede ser una desventaja en situaciones donde se necesitan evaluaciones rápidas o incluso en futuros desarrollos orientados a la detección de pose en tiempo real.

En contraste, MotionBERT demuestra un perfil de rendimiento más mixto, donde se puede destacar un tiempo de inferencia reducido con respecto al resto de los modelos. Si bien su estrategia de preentrenamiento, centrada en aprender representaciones robustas del movimiento a partir de entradas 2D corruptas, es prometedora para una gama de tareas posteriores, las métricas agregadas indican errores más altos (p. ej., RMSE: 12.28º ± 4.59º r: 0.79 ± 0.11) debido a su dependencia del preentrenamiento autosupervisado. Por ejemplo, a pesar de un fuerte rendimiento en ciertas tareas de las extremidades inferiores, como se ve en la Actividad A04 (*sit_to_stand*), con un MAE de 4.48º ± 1.68º y una correlación r de 0.98 ± 0.02, su rendimiento se deteriora en otros escenarios, lo que sugiere que los conocimientos previos sobre el movimiento aprendidos pueden no generalizarse uniformemente a todos los tipos de movimientos. Si bien el preentrenamiento de MotionBERT sobre esqueletos corruptos mejora la robustez al ruido y las oclusiones, puede generalizar inadecuadamente a la cinemática articular detallada en tareas dinámicas, particularmente donde se requiere una estimación precisa de la profundidad. Además, aunque el enfoque de preentrenamiento le permite incorporar conocimientos previos geométricos y cinemáticos, su rendimiento puede verse limitado por los desafíos de transferir estas representaciones aprendidas a las condiciones específicas del conjunto de datos VIDIMU. Adicionalmente, aunque su diseño refleja un marco unificado innovador para la comprensión del movimiento humano, sus elecciones arquitectónicas y su fecha de lanzamiento anterior en relación con algunos de los modelos más nuevos podrían haber limitado su capacidad para capturar la dinámica sutil requerida para una estimación precisa del ángulo articular.

MMPose demostró un fuerte rendimiento en tareas de las extremidades inferiores como la Actividad A01 (*walk_forward*), con un R^2 de 0.76 ± 0.20, mientras que mostró resultados intermedios en el resto de las tareas. Su proceso modular de tres etapas (detección, estimación de pose 2D, y elevación de pose de 2D a 3D) se beneficia del preentrenamiento específico del dominio en conjuntos de datos como Human3.6m, que enfatiza el análisis de la marcha. Sin embargo, su rendimiento se degradó en tareas de las extremidades superiores (p. ej., en la Actividad A13 *tear_both_arms*, con un RMSE de 17.57º ± 3.53º y una r de 0.93 ± 0.05), probablemente debido al limitado tamaño de 243 fotogramas de la ventana temporal en VideoPoseLift, que tiene dificultades con movimientos prolongados y no repetitivos. Esto probablemente se explica por el hecho de que la dependencia de un proceso secuencial introduce errores acumulativos potenciales, particularmente en actividades con oclusiones significativas o movimientos rápidos. No obstante, sus métricas de rendimiento competitivas en algunas tareas sugieren que, en escenarios donde las detecciones 2D son fiables, el enfoque de elevación puede proporcionar estimaciones aceptables de poses 3D.

Por otro lado, NVIDIA BodyTrack logró resultados competitivos tanto en actividades de las extremidades inferiores como superiores. Sin embargo, sus métricas de error más altas en

escenarios ocluidos (p. ej., en la Actividad A09 *assemble_both_arms* con un RMSE de 10.15º ± 4.00º y un R^2 de 0.19 ± 0.38) destacan una fuerte dependencia de entornos controlados no ocluidos. Aunque BodyTrack se beneficia de la optimización comercial para aplicaciones en tiempo real y más características además de HPE, su falta de transparencia y las limitaciones potenciales en la adaptación a entornos diversos pueden obstaculizar su rendimiento en configuraciones que difieren de las originalmente previstas por el desarrollador. La interoperabilidad también se ve afectada por el hecho de que BodyTrack produce 34 marcadores corporales humanos, mientras que el resto de los modelos producen un total de 17 marcadores, por lo que hubo que realizar un *downsampling* de los datos.

Varios factores contribuyen a estas diferencias de rendimiento. Las arquitecturas de modelos más nuevos como MotionAGFormer (2023) reflejan avances recientes en Aprendizaje Profundo, donde los diseños híbridos ofrecen mejores compromisos entre precisión y eficiencia computacional. El lanzamiento a menudo se correlaciona con dichos avances, ya que los modelos lanzados más recientemente se han beneficiado de conjuntos de datos más grandes y diversos y de metodologías de entrenamiento refinadas que abordan limitaciones conocidas en enfoques anteriores, especialmente en campos como el aprendizaje profundo y la visión computacional, donde el progreso científico avanza a un ritmo rápido. Por otro lado, los modelos desarrollados anteriormente, como MMPose (2022), MotionBERT (2022) o NVIDIA BodyTrack (2021), pueden carecer de estas optimizaciones, lo cual es evidente en la brecha de rendimiento observada en este estudio.

3.3.2. Limitaciones

A pesar de los resultados optimistas, deben reconocerse las limitaciones de esta evaluación comparativa. El conjunto de datos VIDIMU, aunque diseñado para replicar actividades cotidianas, involucra un número limitado de sujetos, tareas y parámetros controlados que podrían no capturar completamente la variabilidad, los movimientos patológicos y las oclusiones que se encuentran en escenarios del mundo real. Por ejemplo, es necesario considerar las condiciones de iluminación, las perspectivas y la ubicación de la cámara, y los patrones de oclusión. El rendimiento de cada modelo también está influenciado por su resolución temporal (p. ej., las predicciones fotograma a fotograma carecen de coherencia a largo plazo, mientras que el uso de ventanas temporales maneja mejor las actividades prolongadas) así como por su régimen específico de preentrenamiento o *fine-tuning*, que podría no ser óptimo para el conjunto de actividades incluidas en nuestra evaluación. Por otro lado, también deben considerarse las limitaciones de hardware: los datos de referencia (*ground truth*) basados en IMUs están inherentemente influenciados por la precisión de la calibración IMU-segmento, lo que potencialmente sesga las métricas e introduce una fuente sistemática de error. En el caso de BodyTrack, surge el debate entre los modelos propietarios y aquellos *open-source*, ya que la dependencia del ecosistema NVIDIA limita la reproducibilidad, mientras que los modelos de código abierto ofrecen transparencia de datos y flexibilidad, pero

requieren importantes recursos computacionales para el entrenamiento. Además, su naturaleza de código cerrado limita el modularidad y plantea algunas preocupaciones sobre la privacidad al evaluar datos de pacientes en entornos clínicos o en aplicaciones industriales.

Conociendo las limitaciones y fortalezas de los sistemas evaluados, es importante mencionar los casos de uso y las aplicaciones más adecuadas para cada uno. MotionAGFormer sobresale en actividades complejas y dinámicas tanto de las extremidades inferiores como superiores, lo que lo hace ideal para aplicaciones que requieren alta precisión, como la telerehabilitación avanzada y el análisis del rendimiento deportivo. En contraste, MotionBERT destaca en trayectorias predecibles, estructuradas y lineales debido a su preentrenamiento autosupervisado sobre datos de movimiento diversos, pero tiene dificultades con movimientos abruptos y el posicionamiento. Sin embargo, sus márgenes de error ligeramente más altos pueden limitar su uso en aplicaciones donde la precisión exacta del ángulo articular es primordial. MMPose, aprovechando un enfoque secuencial de elevación de 2D a 3D, proporciona una solución equilibrada cuando se dispone de detecciones de puntos clave 2D de alta calidad y son aceptables ligeras concesiones en la precisión. Por lo tanto, representa una alternativa adecuada para aplicaciones centradas en las extremidades inferiores y la marcha. Por otra parte, BodyTrack, con un rendimiento equilibrado e integrado en plataformas comerciales de Realidad Aumentada (AR), es más apto para escenarios que priorizan la velocidad y la simplicidad operativa.

4. Parte II. Desarrollo de un modelo para la segmentación de la marcha

La segunda parte de este trabajo se centra en el diseño y la validación de un modelo de aprendizaje profundo capaz de segmentar las dos fases principales de la marcha a partir de la evolución temporal de los ángulos articulares. Partiendo de la base de datos descrita en la Parte I, se construye un conjunto de secuencias sincronizadas que combina el vector de tiempos, los ángulos obtenidos mediante cinemática inversa y las etiquetas de fase derivadas de la velocidad angular de los sensores inerciales. Sobre este corpus se plantea una arquitectura LSTM que se beneficia la capacidad de las redes neuronales recurrentes (RNN) para capturar la dependencia temporal de largo alcance presente en la marcha humana. El objetivo final es generar un clasificador robusto que identifique en los intervalos de *Stance* y *Swing*, así como la etiqueta *Turn*.

4.1. Materiales y métodos

4.1.1. Preparación de los datos. Procesado de señales

La primera etapa para el desarrollo del modelo de aprendizaje profundo fue la preparación de los datos requeridos para su entrenamiento y validación. Al no existir etiquetas previas en los datos presentes en la base de datos VIDIMU descrita en la Parte I, se ideó un método para el etiquetado de datos semiautomático basado en la obtención de la señal de velocidad angular. Para ello, se empleó únicamente la actividad A01 (*walk_forward*). Cada adquisición (16 en total, una por sujeto) contiene seis recorridos consecutivos a lo largo del plano sagital, con tres o cuatro pasos por pierna en cada tramo, grabados tanto con la cámara de vídeo como con los cinco IMUs. En específico, se emplearon los datos de los sensores fijados en la parte frontal de la tibia (identificados como qsLLL para la pierna izquierda y qsRLL para la derecha), ya que captan con mayor amplitud el movimiento pendular del segmento inferior en el plano sagital y son ampliamente utilizados en aplicaciones de análisis de la marcha al generar picos de velocidad angular bien definidos.

En primer lugar, se cargaron los datos de los sensores desde los ficheros en crudo (*.raw*) con Pandas. Se realizó un preprocesado que consistió en descartar valores vacíos y normalizar el rango de tiempo restando las marcas de tiempo primera y última. Además, se descartaron las primeras filas, destinadas a la calibración del hardware. Cada fichero *.raw* tiene las siguientes cabeceras:

- *QUAT*: Cuaternión seleccionado. Hay cinco disponibles, uno por cada sensor (qsHIPS para la cadera, qsRUL y qsLUL para la parte superior de cada pierna, y qsLLL y qsRLL para la parte inferior de cada pierna).
- *w,x,y,z*: Coordenadas que definen el cuaternión según la expresión $q_k = \{\mathrm{w}, \mathrm{x}, \mathrm{y}, \mathrm{z}\}$.

- *timestamp*: Marcas de tiempo con una separación aproximada de $\Delta t = 0.02$ s (ya que el muestreo se realizó a 50 Hz).

Tabla 4.1. Cuaterniones extraídos de los 5 IMUs.

Etiqueta	**Leyenda**
qsHIPS	Hips
qsRUL	Right Upper Leg
qsRLL	Right Lower Leg
qsLUL	Left Upper Leg
qsLLL	Left Lower Leg

El siguiente paso en el flujo de procesamiento es el cálculo de la velocidad angular (ω) como la derivada en el tiempo de la posición. Los cuaterniones se forzaron a ser continuos en signo, de modo que si $q_{k-1} \cdot q_k < 0$, entonces se calcula $q_k = -q_k$. Luego, se pasó a diferenciar y la velocidad angular instantánea se obtuvo mediante el producto:

$$d_{q_k} = q_k \otimes \overline{q}_{k-1}, \quad \theta_k = 2\arccos\left(d_{q_k}^w\right), \quad \omega_k = \frac{2\,\theta_k}{\Delta t}\,u_k \tag{4.1}$$

donde $\theta_k = 2\arccos\left(d q_{k,0}\right)$ y u_k es el eje de giro. Entre las tres componentes de w_k se escogió la de mayor varianza por ser aquella que coincide con la dirección dominante de la pierna. La Figura 4.1 muestra la señal de velocidad angular en crudo para el sujeto S47 realizando la actividad A01 (*walk_forward*), en el intento T01 y medida con el sensor qsLLL.

Posteriormente, se realizó un filtrado paso bajo con un filtro Butterworth de cuarto orden y con frecuencia de corte 0.5 Hz con el objetivo de suavizar la señal de velocidad angular y eliminar los posibles *outliers*. Este valor elimina el ruido de alta frecuencia sin distorsionar la envolvente fundamental (aproximadamente 1 Hz para una cadencia normal de marcha). El resultado se muestra en la Figura 4.2.

El siguiente paso fue la corrección del signo de la velocidad angular. Dado que el sujeto camina de un lado a otro en el plano sagital, y al llegar al final del plano de cámara gira sobre sí mismo y comienza a caminar en dirección opuesta, existe un cambio de signo en cada "conjunto de pasos" de la señal original de velocidad angular, como puede observarse en la Figura 4.1 y en la Figura 4.2. Para unificar el signo en todo el ensayo se calculó la envolvente mediante la transformada de Hilbert, se suavizó con una media móvil de 0.4 s y se obtuvieron segmentos de marcha continua. En cada segmento la mediana de los valores con magnitud relevante ($|\omega| > 0.15\sigma$) determina la orientación predominante. Si esta orientación difiere de la del primer segmento, la señal dentro del segmento actual se multiplica por -1. De este modo los picos positivos representan siempre la fase de balanceo en ambas direcciones del recorrido. Sobre la señal ya filtrada y con el signo corregido se detectaron algorítmicamente los eventos principales del ciclo de la marcha humana, que son los siguientes:

- ***Mid Swing***. Se localizan los máximos de ω mediante la búsqueda de "picos" con una distancia mínima de 0.3s, prominencia de 0.25 σ y una altura mayor que $\mu + 0.10\sigma$.
- **Contactos con el suelo (Valles de ω)**. Se aplicó el mismo algoritmo, esta vez a la señal $-\omega$. Estos valles deben ser clasificados como HS o TO según cuando aparezcan.
- ***Heel Strike* (HS) y *Toe Off* (TO)**. Para cada máximo de *mid swing*, el último valle anterior a dicho pico se marca como TO y el primer valle posterior se marcha como HS. Además, el algoritmo diseñado evita detecciones espurias imponiendo una distancia mínima de 0.30 s entre eventos consecutivos. La Figura 4.3 muestra un ejemplo de la detección de eventos de la marcha para el sujeto S47 realizando la actividad A01.

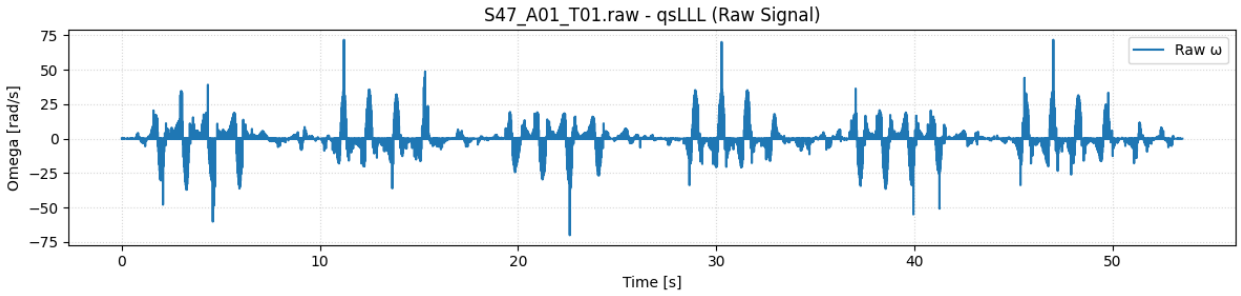


Figura 4.1. Velocidad angular derivada a partir de los cuaterniones.

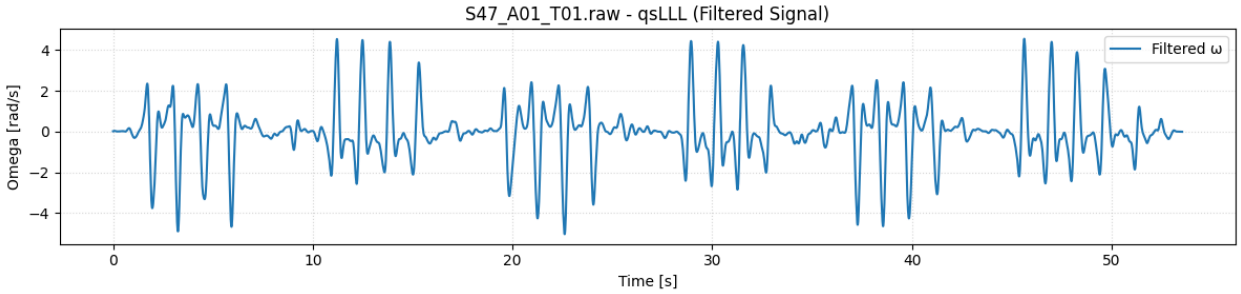


Figura 4.2. Velocidad angular tras el filtrado paso-bajo.

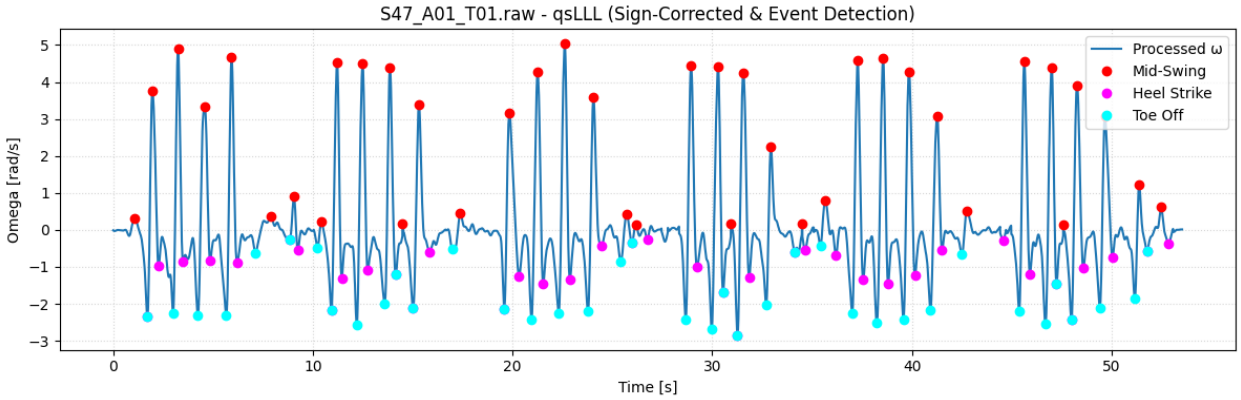


Figura 4.3. Velocidad angular con corrección de signo y detección de eventos de la marcha.

Tras la obtención de los eventos de la marcha, se generó una lista ordenada con los HS y TO con sus correspondientes marcas de tiempo. Luego, se asignaron las etiquetas iniciales como se puede apreciar en la Tabla 4.2.

Tabla 4.2. Etiquetas empleadas para clasificar los datos.

Etiqueta	**Leyenda**	**Detalles**
0	*Stance*	Intervalo [HS, TO).
1	*Swing*	Intervalo [TO, HS).
2	*Turn*	Puntos cuya envolvente se encuentra por debajo del percentil 15%. Corresponde al giro que el sujeto realiza al final de cada recorrido.
-1	*Unclassified*	Muestras erróneas o que no encajan en las reglas anteriores.

La Figura 4.4 ilustra un ejemplo de la segmentación inicial y el etiquetado que el algoritmo realizó automáticamente en base a los máximos y mínimos de la señal. La detección contiene algunas etiquetas erróneas, por lo que fue necesario realizar una inspección manual de algunos de los ficheros con el objetivo de corregir los datos y mejorar su calidad global para disponer de un mejor *input* a la hora de entrenar y validar el modelo de aprendizaje profundo.

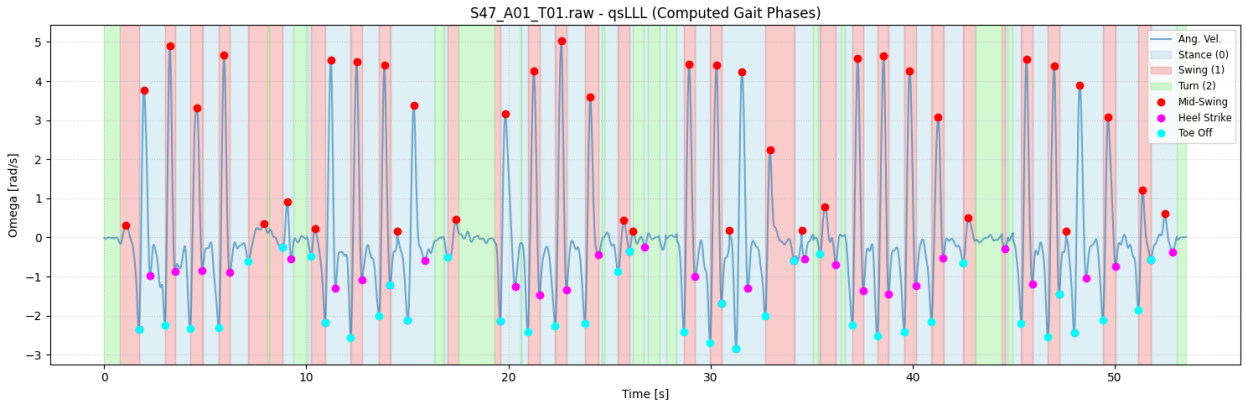


Figura 4.4. Estimación inicial automatizada de las fases de la marcha.

4.1.2. Etiquetado de la base de datos

Para garantizar que el modelo LSTM recibiera datos de entrenamiento de la máxima calidad posible, se revisaron manualmente las etiquetas con las fases de la marcha generadas de forma automática por el algoritmo diseñado. Para hacer este proceso más sencillo y eficiente, se desarrolló una interfaz gráfica de usuario (*Graphical User Interface,* GUI) específica en Python con PyQt 6 y los *backends* de Matplotlib, SciPy, Numpy y Pandas entre otros. El enfoque fue construir una herramienta básica pero funcional que mostrase las señales de velocidad angular y los datos preetiquetados, y permitiese hacer correcciones sobre los mismos y guardarlos para formar un nuevo *dataset* de información corregida y clasificada.

A continuación, se presenta una breve descripción de la interfaz gráfica construida (ver Figura 4.5). La ventana principal se divide en dos paneles. A la izquierda se muestra una lista de los ficheros que han sido preprocesados por el algoritmo de etiquetado de fases automático y que están pendientes de revisar manualmente. Al recorrer la lista de archivos mediante los botones "Previous File" y "Next File", se van guardando automáticamente los cambios realizados en el fichero abierto actualmente para evitar la pérdida de información accidental. En el panel de la derecha aparece representada la señal de velocidad angular filtrada, con corrección de signo, con eventos de la marcha indicados y con las fases de la marcha etiquetadas de forma automática.

La edición de fases se controló mediante dos modos de interacción conmutables mediante los botones correspondientes:

- **"Modify existing phase segment".** Este modo detecta los intervalos ya definidos previamente por el algoritmo de búsqueda de máximos y mínimos, de modo que un clic dentro de un intervalo dado permite cambiar la etiqueta de fase a una nueva en ese rango temporal. Es decir, en este modo se pueden seleccionar y modificar las etiquetas de los intervalos ya existentes.

- **“Define new phase region”.** Este modo permite definir nuevos intervalos. El usuario puede introducir el inicio de una nueva región cualquiera con el primer clic y fijar el final de dicha región con el segundo clic.

Además, la interfaz dispone de algunas funcionalidades internas para hacer el proceso más visual y fluido. Por ejemplo, en cualquiera de los modos el área seleccionada se marca con un rectángulo semitransparente y puede asignarse a cualquiera de las clases disponibles mediante un botón Stance (0), Swing (1), Turn (2) o Unclassified (-1). Por otro lado, cuando el usuario pulsa en las proximidades de un evento de la marcha (HS en magenta o TO en cian) el cursor se ajusta de forma automática al instante exacto del evento, lo que acelera la anotación de las muestras y mejora la precisión. Este efecto “magnético” se controla mediante un valor de tolerancia modificable (0.1 s por defecto).

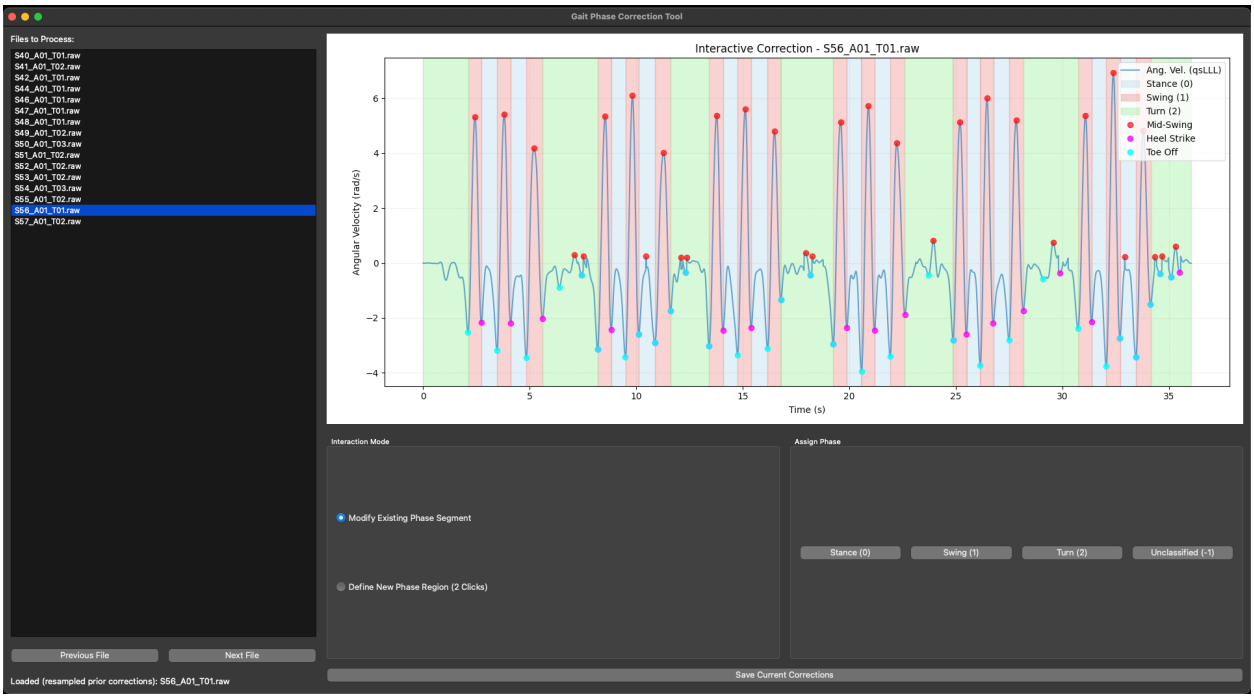


Figura 4.5. Interfaz gráfica para el etiquetado manual de fases de la marcha.

El flujo de trabajo para la corrección de las muestras viene definido por el *backend* de la GUI. En primer lugar, se cargaron los ficheros RAW correspondientes a todas las señales de velocidad angular calculadas en el apartado anterior (una por adquisición y sujeto), con la tasa de muestreo de origen (50 Hz para los cuaterniones). El script comprobaba si ya existía un fichero corregido, y cargó aquellos donde ya se habían realizado modificaciones previamente. Luego, se inspeccionó visualmente la señal, cambiando las fases erróneas y creando nuevos intervalos en los puntos donde era necesario. La inspección permitió descartar varios ensayos en los que la señal resultaba inservible debido a fallos de calibración o a pérdidas de datos en los IMUs. Un caso representativo de esto fue el del sujeto S46 para el sensor tibial izquierdo (qsLLL), o los sujetos

S42, S48, S51 y S57 para el sensor tibial derecho (qsRLL) donde la velocidad angular mostraba valores saturados y un ruido que impedía identificar picos válidos (motivo por el cual se optó por clasificar la marcha únicamente en la pierna izquierda). Estos ficheros se eliminaron de la base de datos final para no sesgar el entrenamiento.

Una vez corregidos todos los ficheros, el script tomó la información sobre las etiquetas de fases de la marcha y realizó un remuestreo a 30 Hz para que la información fuese compatible con los ángulos ya calculados mediante OpenSim y guardados en ficheros MOT según se explica en la Parte I. Luego, se combinaron estos datos con los ángulos articulares, de modo que los ficheros resultantes, preparados para servir como datos de entrenamiento para el BiLSTM, contenían las siguientes columnas:

- **time**: marca temporal en segundos desde el inicio del ensayo.
- **knee_angle_l**: ángulo de la rodilla izquierda en grados.
- **knee_angle_r**: ángulo de la rodilla derecha en grados.
- **phase**: etiqueta de fase (0 = Stance, 1 = Swing, 2 = Turn, -1 = Unclassified).

En la Figura 4.6 puede verse el resultado de la corrección manual de las fases de la marcha. Una vez corregidos todos los ficheros, se pasó al diseño de una red neuronal recurrente que modelase estas relaciones y fuese capaz de estimar las fases en ficheros de ángulos articulares no vistos previamente.

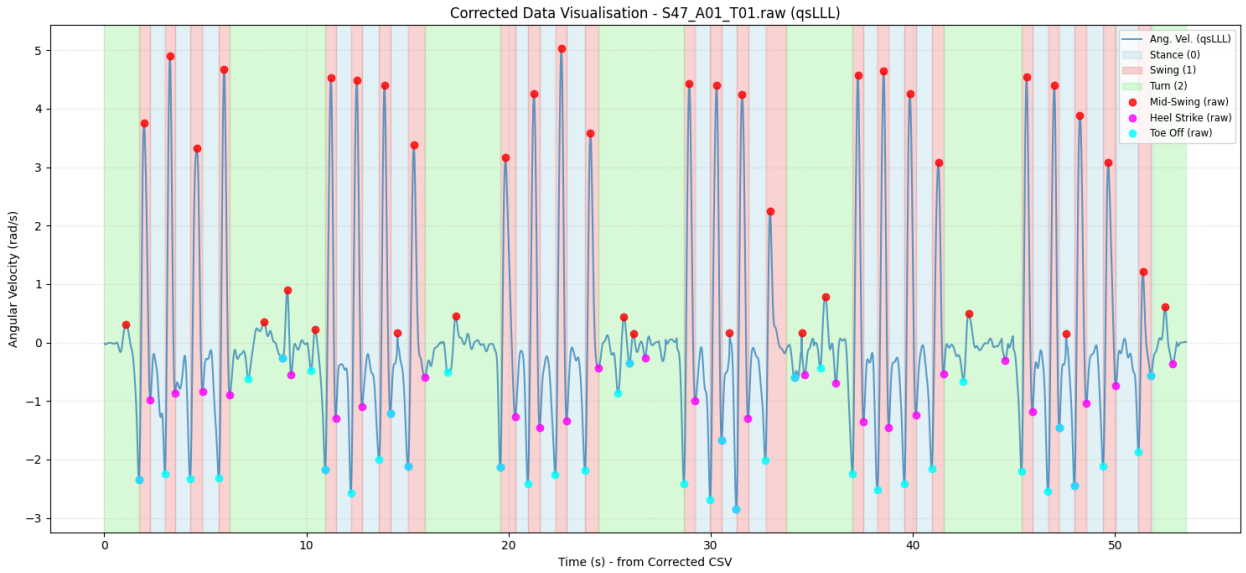


Figura 4.6. Estimación final de las fases de la marcha tras la corrección manual.

4.1.3. Modelo BiLSTM para la segmentación de la marcha

Arquitectura BiLSTM y selección de características

Tras obtener una base de datos robusta y con datos etiquetados, se pasó a conceptualizar y diseñar un modelo para la segmentación de las dos principales fases de la marcha a partir de ángulos articulares. La arquitectura final se seleccionó tras una experimentación comparativa que incluyó cuatro variantes progresivamente más complejas de Redes Neuronales Recurrentes (*Recurrent Neural Networks,* RNN). En concreto, se realizaron pruebas con los sistemas presentados a continuación:

i) LSTM (*Long Short-Term Memory*) unidireccional sencillo.
ii) BiLSTM (*Bidirectional Long Short-Term Memory*).
iii) BiLSTM-CNN 1D (*BiLSTM Convolutional Neural Network One Dimension).*
iv) BiLSTM-CNN 1D con mecanismos de atención.

Dado que la base de datos creada contenía un número limitado de ciclos de marcha, los modelos con mayor número de capas y parámetros, como por ejemplo BiLSTM-CNN con y sin mecanismos de atención, manifestaron un sobreajuste acusado incluso aplicando técnicas de regularización y validación cruzada. Asimismo, las primeras pruebas con *transformers* se descartaron por resultar desproporcionadas en relación con el volumen de datos y por requerir tiempos de entrenamiento y GPUs muy superiores. En cambio, la arquitectura BiLSTM ofreció un equilibrio óptimo entre capacidad de modelado temporal, convergencia estable y coste computacional moderado. Además, la literatura sobre análisis automático de la marcha respalda el uso de este modelo como una de las soluciones que mejor rendimiento han mostrado para segmentación en contextos clínicos, por lo que se consideró la opción más razonable para este trabajo.

El modelo desarrollado es una red BiLSTM de 1 capa LSTM con 64 *hidden layers* por dirección (128 por instante temporal en total al ser bidireccional). Se aplicó una capa *fully connected* que proyectó cada vector de estado a tres *logits* correspondientes a *Stance*, *Swing* y *Turn*. No se incluyó ninguna función de activación explícita en la cabeza de clasificación, ya que la función de pérdida *CrossEntropyLoss* de PyTorch añade de manera implícita la operación softmax necesaria para obtener probabilidades durante el cálculo del gradiente.

Las características de entrada se limitaron a los ángulos de las rodillas izquierda y derecha. Antes de construir las secuencias estos dos canales se normalizaron con StandardScaler ajustado exclusivamente sobre los datos de entrenamiento para evitar fuga de información. Cada ensayo se fragmentó mediante una ventana deslizante de 75 muestras con un desplazamiento de una sola muestra, lo que generó secuencias de 1.5 segundos de duración a 50 Hz. El vector asociado de etiquetas incluyó una clase por paso temporal, de modo que la red aprendió a emitir una predicción

frame a *frame* y no a nivel de ciclo. La columna de tiempo se conservó sólo para alinear las salidas durante la fase de inferencia y para la representación gráfica posterior.

Carga de datos y secuencias para la validación cruzada

El *data loader* del modelo recorrió la carpeta de salida de la interfaz gráfica descrita anteriormente para detectar los ficheros de cada sujeto y agrupó los ensayos por identificador para garantizar que toda la información de un mismo participante permaneciera en el mismo bloque. Se empleó la técnica de validación cruzada, concretamente K-Fold para hacer el entrenamiento más robusto. De este modo, se dividieron los datos en *k* partes o *folds* y se entrenó el modelo *k* veces, de modo que en cada iteración los sujetos de entrenamiento y de validación fueron variando. Para cada pliegue se ajustó un StandardScaler únicamente sobre las características de entrenamiento y se utilizó después para transformar los conjuntos de validación y prueba, preservando así la coherencia de la escala. Este objeto *scaler* se serializó dentro del diccionario de estado que acompaña a cada *checkpoint* junto con los pesos de la red y el optimizador, lo que permitió restaurar exactamente el mismo preprocesado en la fase de inferencia sin riesgo de desalinear las distribuciones.

Técnicas de regularización

La limitación del conjunto de entrenamiento (que contaba con un número reducido de adquisiciones y, en última instancia, de ciclos de marcha) hizo imprescindible recurrir a un repertorio amplio de técnicas de regularización para evitar que la red memorizara secuencias concretas y perdiera capacidad de generalización, fenómeno conocido como *overfitting* o sobreajuste.

El primer método de regularización consistió en aplicar *dropout* a dos niveles: un 0.50 entre las capas de la red BiLSTM y otro 0.50 en la proyección lineal final o capa de salida. Al anular de forma aleatoria la mitad de las neuronas en cada pasada se forzó a la red a mantener rutas de información redundantes y se redujo la dependencia de conexiones específicas. Paralelamente se activó la regularización L2 (*weight decay*) con un coeficiente de 1e-4 dentro de AdamW, añadiendo al término de pérdida una penalización proporcional al cuadrado de los pesos que empujó la optimización hacia soluciones de norma menor, menos sensibles al ruido y, por tanto, más estables frente a sujetos no vistos. El modelo se dimensionó de manera conservadora con 64 *hidden layers* por dirección (128 en total) y una capa para limitar sus parámetros entrenables, aunque manteniéndolos adecuados al tamaño del corpus. Además, se estableció un protocolo de *early stopping* que vigilaba la pérdida de validación y detenía el aprendizaje si no se observaba mejora durante veinte épocas consecutivas (parámetro de paciencia igual a 20), evitando así la degradación típica de las últimas iteraciones. Finalmente, la estrategia de ventana deslizante con paso de una sola muestra descrita antes actuó como una forma implícita de aumento de datos, ya que expuso a la red a variaciones temporales mínimas de cada señal.

Hiperparámetros de entrenamiento

Los hiperparámetros constituyen aquellas variables de configuración que determinan el tamaño, la velocidad de aprendizaje y las reglas de regularización de una red antes de iniciar el entrenamiento. A diferencia de los parámetros internos que la red ajusta mediante el descenso de gradiente, los hiperparámetros se fijan externamente y controlan la capacidad de generalización, la rapidez de convergencia y la estabilidad numérica del proceso. Una elección inadecuada puede conducir a sobreajuste, divergencia o tiempos de entrenamiento excesivos. El procedimiento sistemático seguido para ajustar estos valores se describe en detalle en la siguiente sección. A continuación se resumen los hiperparámetros finalmente adoptados, su función en el modelo y los *trade-offs* que plantea la elección de distintos valores.

Tabla 4.3. Hiperparámetros finalmente empleados para el BiLSTM.

Parámetro	**Valor**	**Descripción**
Preprocesado		
sequence_length	75	Longitud de cada secuencia (2 segundos a 50 Hz).
normalization_method	standard	Escalado con media cero y varianza unitaria.
hiperarámetros globales		
model_type	LSTM	Tipo declarado en el config (implementado como BiLSTM).
lstm_hidden_size	128	Unidades ocultas por dirección en cada capa LSTM. Define la capacidad de memoria temporal.
num_lstm_layers	1	Profundidad de la red recurrente. Mayor número aumenta la abstracción pero eleva el riesgo de sobreajuste.
lstm_dropout	0.5	Dropout aplicado entre capas LSTM.
bidirectional_lstm	True	Procesa la secuencia en ambos sentidos temporales.
linear_dropout	0.5	Dropout antes de la capa de clasificación.
Hiperparámetros de entrenamiento		
learning_rate	0.001	Tasa inicial de aprendizaje del optimizador AdamW. Controla la velocidad de convergencia.
batch_size	64	Número de secuencias por lote durante el entrenamiento. Supone un compromiso entre la estabilidad del gradiente y el uso de memoria.
num_epochs	100	Límite máximo de épocas (early stopping activo).
weight_decay	0.001	Penalización L2 para limitar la magnitud de los pesos.
Validación cruzada		
k_folds	5	Número de pliegues en la validación cruzada por sujetos. Aumenta la fiabilidad de la estimación de rendimiento.
test_split_ratio	0.15	Proporción de sujetos reservados para prueba si K_FOLDS = 1.
validation_split_ratio	0.15	Porción interna dedicada a validación cuando no hay K-fold.
random_seed	42	Semilla para reproducibilidad.

Early stopping		
early_stopping_patience	20	Épocas sin mejora antes de detener el entrenamiento.
early_stopping_delta	0.0005	Umbral mínimo de mejora para reiniciar la paciencia.
Inferencia		
inference_batch_size	64	Tamaño de lote durante la inferencia.

Función de pérdida ponderada

En lo relativo al criterio de pérdida (*cross-entropy*), cabe destacar que se incorporó una función de pérdida ponderada para mitigar la sobrerrepresentación de la clase *Turn* en los primeros resultados, resultado de una mayor presencia de etiquetas de dicha clase en el conjunto de entrenamiento. A partir de estos datos, se calcularon pesos inversamente proporcionales a la frecuencia de cada clase, asignando un peso sustancialmente mayor a *Turn*. Estos coeficientes hicieron que cada error cometido sobre muestras de esta clase penalizó más el gradiente y evitó que la red ignorara este patrón minoritario.

Estrategia de validación

Para estimar la capacidad de generalización del modelo se empleó la técnica de validación cruzada (*cross-validation*), y más específicamente K-Fold. De este modo, se implementó GroupKFold tomando cada identificador de sujeto como grupo, garantizando así que los registros de un mismo individuo nunca aparecieran simultáneamente en entrenamiento y validación, y evitando fugas de información derivadas de la dependencia intrasujeto. En cada iteración el modelo se entrenó sobre aproximadamente la mitad de los sujetos y se validó sobre la otra mitad, repitiendo el proceso para invertir los papeles de los pliegues y obtener así dos estimaciones independientes del rendimiento.

Durante el entrenamiento se calcularon en cada época las siguientes métricas sobre el conjunto de validación: pérdida de entropía cruzada ponderada (*validation loss*), exactitud (*validation accuracy*) y F1-score con promediado ponderado. Este último se utilizó como métrica primaria para el *early stopping*, configurado con una paciencia de 20 épocas y un delta mínimo de mejora de 0.0005. Si transcurrían 20 iteraciones sin que esta métrica disminuyera en al menos ese umbral, el entrenamiento se detenía para impedir sobreajuste y ahorrar tiempo de cómputo. Cada vez que la pérdida alcanzaba un nuevo mínimo se almacenaba un *checkpoint* que incluía los pesos de la red, el optimizador, el escalador de características y el vector de pesos de la función de pérdida, de forma que la versión con mejor desempeño estuviera disponible para la prueba y la inferencia. La presencia de la clase sobrerrepresentada *Turn* se vigiló mediante el F1-score ponderado y la inspección de dichas matrices. Los pesos aplicados en la función de pérdida contribuyeron a equilibrar su influencia y a mejorar la sensibilidad del modelo en esta fase.

Inferencia

Una vez finalizado el entrenamiento, se ejecutó el módulo de inferencia para cuantificar el rendimiento del modelo y generar salidas legibles. Se cargó automáticamente el *checkpoint* con menor pérdida de validación, se restauró el escalador y se recorrió cada archivo en el *dataset* de inferencia con ventanas de 75 muestras y un lote de 64 secuencias. Las predicciones se agregaron a un vector de longitud igual al número de instantes originales, de modo que cada índice temporal recibió una etiqueta correspondiente a la fase estimada.

El *dataset* de inferencia fue construido a partir de los ángulos articulares generados por las distintas arquitecturas de estimación de pose analizadas en la Parte I. Esta evaluación cruzada permitió valorar la capacidad de generalización del modelo BiLSTM ante entradas derivadas de vídeo, no vistas durante el entrenamiento y potencialmente afectadas por los errores sistemáticos de cada estimador 3D.

Para cada ensayo se exportó un CSV con las columnas *time, knee_angle_l, knee_angle_r, predicted_phase_model* y, cuando existían, *true_phase_gui*. El script almacenó estos ficheros en el directorio de salida y generó automáticamente dos tipos de gráficas:

- **Predicted vs. true phases.** Superpone la etiqueta del modelo y la de la GUI, lo que facilita la inspección puntual de falsos positivos o transiciones adelantadas.
- **Predicted phases vs. angular velocity.** Recupera la velocidad angular del sensor bruto correspondiente, la interpola sobre la rejilla temporal del CSV y colorea cada segmento según la fase asignada. Además, marca los eventos *heel strike, toe off* y *mid swing* calculados anteriormente. Esta representación visual permitió confirmar que las fronteras entre *Stance* y *Swing* coincidían con los valores extremos de la señal dinámica.

Métricas de evaluación

La fase de evaluación concluyó con la generación automática de las gráficas de aprendizaje (evolución de la pérdida y de la precisión tanto en entrenamiento como en validación) y de las matrices de confusión asociadas al mejor modelo obtenido en cada *fold*. Las primeras permiten apreciar la convergencia y detectar posibles signos de sobreajuste, mientras que las segundas cuantifican el acierto por clase y revelan los patrones de error más frecuentes, en particular la confusión entre las clases *Swing* y *Turn*. El análisis detallado de estas métricas, junto con los valores de precisión, pérdida y F1-score calculados sobre los conjuntos de validación y prueba, se presenta de forma exhaustiva en la sección de Resultados.

4.1.4. Ajuste de hiperparámetros mediante optimización Bayesiana

Con el fin de refinar el rendimiento del modelo BiLSTM de forma sistemática, se recurrió a Optuna, un *framework* de optimización de hiperparámetros de código abierto que automatiza la

búsqueda de configuraciones óptimas con un coste computacional reducido. Optuna modela la exploración del espacio de hiperparámetros como un estudio compuesto por *trials* independientes donde cada *trial* representa un entrenamiento completo con un conjunto de valores sugeridos para las variables que se desean optimizar.

En este trabajo se empleó el *sampler* TPE (*Tree-structured Parzen Estimator*), un método de optimización Bayesiana que consiste en lo siguiente: Tras una exploración inicial basada en *start-up trials* aleatorios (en nuestro caso 10), ajusta dos distribuciones (una "buena" y otra "mala" según la métrica obtenida), y sugiere nuevos puntos maximizando la razón de verosimilitud de la mejora prevista entre los dos modelos construidos. Además, en paralelo se activó un *Median Pruner* que interrumpe de forma temprana los intentos cuyo desempeño intermedio queda por debajo de la mediana histórica, liberando recursos para combinaciones previsiblemente más prometedoras.

Cabe destacar que en este punto la elección de la métrica a optimizar mediante este estudio automatizado es importante y depende en gran medida del problema a resolver. Se realizaron varios intentos preliminares donde se estableció la *validation loss* como métrica a optimizar, pero en estos casos el modelo tendía a asignar la mayoría de las muestras a la clase *Turn* (la más frecuente), de modo que se inflaba artificialmente la pérdida sin reflejar un cambio real en la calidad de la estimación. Por este motivo, se cambió la métrica objetivo para maximizar el F1-score ponderado, más representativo para conjuntos de datos desbalanceados. La Figura 4.7 muestra los hiperparámetros más importantes a la hora de mejorar el F1-Score con nuestro modelo.

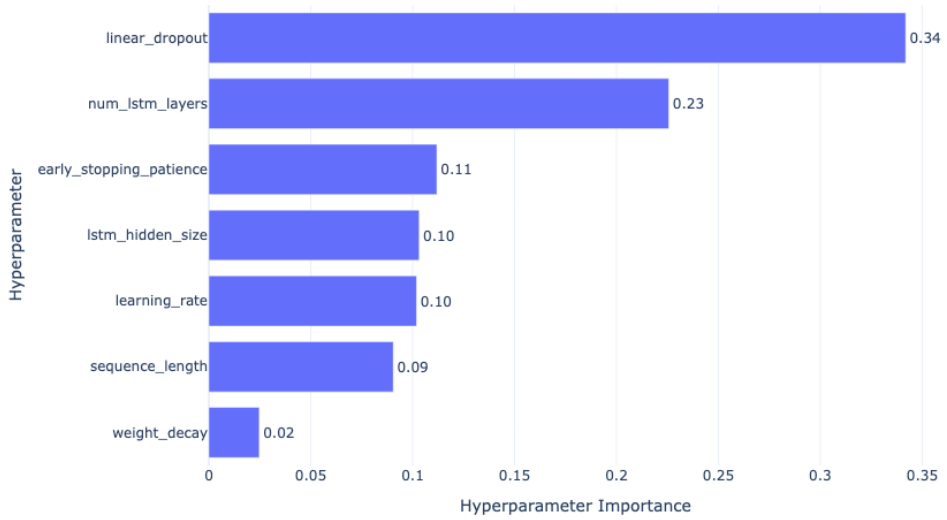


Figura 4.7. Hiperparámetros más relevantes para este estudio según PED-ANOVA (Watanabe et al., 2023).

La configuración concreta del estudio realizado se describe a continuación. Se realizaron aproximadamente 100 iteraciones (una por combinación sugerida), con un único proceso en paralelo para garantizar la estabilidad y el espacio de búsqueda de hiperparámetros indicado en la

Tabla 4.4. Además, la Figura 4.8 ilustra como cierta elección en los parámetros del modelo influye significativamente en la calidad de la estimación cuantificada por el F1-Score.

Tabla 4.4. Rangos elegidos para los hiperparámetros ajustados con Optuna.

Hiperparámetro	**Rango**
lstm_hidden_size	{32, 64, 96, 128}
num_lstm_layers	{1, 2}
lstm_dropout	0.1 - 0.5 (solo si *num_lstm_layers*>1)
linear_dropout	0.3 - 0.6 (step 0.1)
weight_decay	1×10^{-5} - 1×10^{-3} (log-uniforme)
learning_rate	1×10^{-5} - 2×10^{-3} (log-uniforme)
sequence_length	{75, 100, 125}
early_stopping_patience	10 - 20 epochs

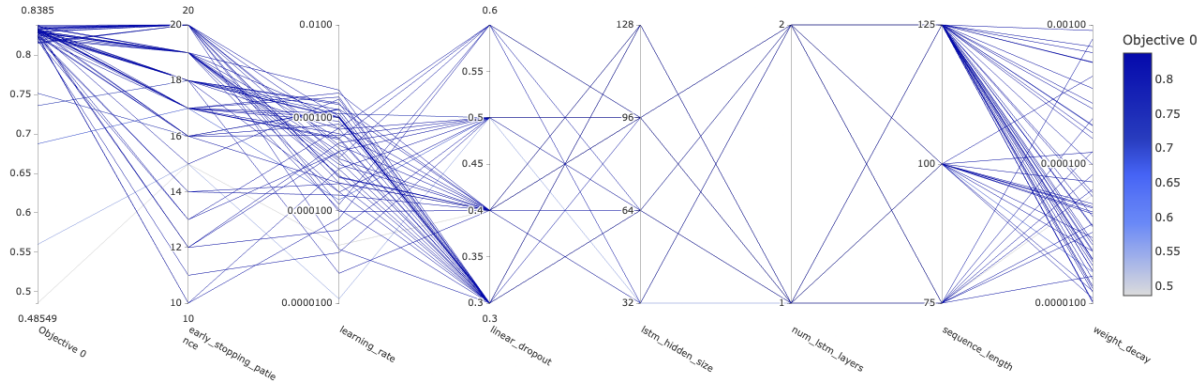


Figura 4.8. Gráfico de coordenadas paralelas para el espacio de hiperparámetros.

Con el fin de conservar la trazabilidad, los sucesivos ensayos de Optuna quedaron registrados en una base de datos SQLite, lo que permitió pausar/reanudar la búsqueda, consultar la mejor iteración, o replicar una ejecución dada. Además, el módulo de optimización fue encargado de lanzar el entrenamiento sobre el pliegue 0 en cada iteración, reescribir dinámicamente los valores de configuración y crear una carpeta aislada donde redirigir los *checkpoints*, gráficas de entrenamiento y matrices de confusión. Todos los resultados de la mejor época en cada *trial* fueron añadidos a un CSV de logs junto con los hiperparámetros empleados en dicha época para el posterior análisis de los datos.

4.2. Resultados

Para evaluar el rendimiento de la arquitectura BiLSTM diseñada, el análisis se dividió en dos etapas. En primer lugar, se evaluó el modelo utilizando como datos de entrada los ángulos articulares derivados directamente de los sensores inerciales (IMUs), que sirvieron como datos de referencia o "ideales". Posteriormente, se evaluó la capacidad de generalización del modelo al ser alimentado con los ángulos articulares generados por cada una de las cuatro arquitecturas de estimación de la pose humana (HPE) basadas en vídeo. Conviene destacar que, a pesar de realizar el entrenamiento con los ángulos de ambas rodillas para mejorar el contexto de aprendizaje, el modelo estima la fase de la marcha para una única pierna, en nuestro caso la izquierda.

4.2.1. Resultados por modelo HPE

Inicialmente, la red neuronal recurrente fue entrenada utilizando únicamente los ángulos articulares calculados a partir de los IMUs, muestreados a 50 Hz. Esta configuración representa el escenario de mejor rendimiento posible (F1-Score superior al 80 % en todos los casos), ya que los datos de entrada provienen de la misma fuente que los utilizados para generar las etiquetas de referencia (aunque los sujetos empleados para el entrenamiento y para la validación no son los mismos al emplear K-fold). La Figura 4.9 muestra la matriz de confusión para el conjunto de entrenamiento, donde se observan valores muy altos en la diagonal principal, lo que indica una capacidad de aprendizaje adecuada a los datos presentados. La Figura 4.10 hace lo propio para el conjunto de inferencia. Las confusiones más notables se producen entre las clases *Stance* (Fase 0) y *Turn* (Fase 2), y en menor medida entre *Swing* (Fase 1) y *Turn*, lo que es coherente con la naturaleza de los giros, que a menudo incluyen características cinemáticas similares a dichas fases.

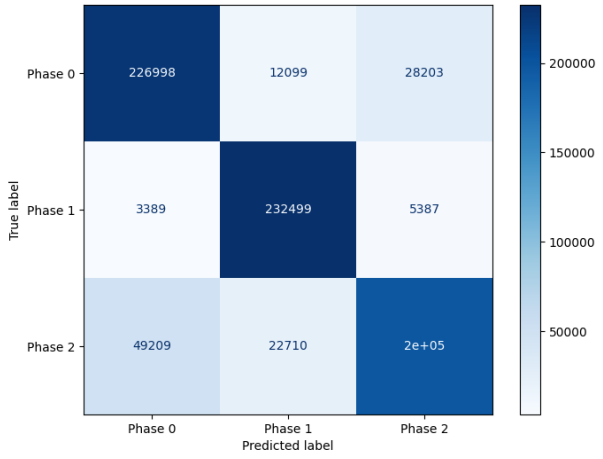


Figura 4.9. Matriz de confusión en entrenamiento para ángulos obtenidos mediante IMUs (50 Hz).

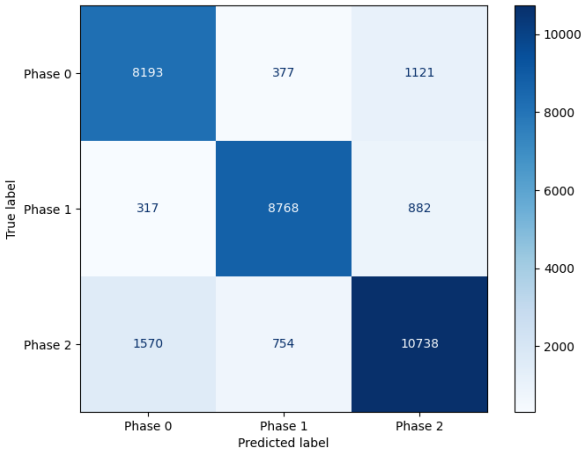


Figura 4.10. Matriz de confusión en inferencia para ángulos obtenidos mediante IMUs (50 Hz).

Con objeto de realizar una validación cualitativa, la Figura 4.11 ilustra un ejemplo (sujeto S50) donde se superponen las fases estimadas por el BiLSTM sobre la señal de velocidad angular de la pierna izquierda. Se puede apreciar cómo, de forma general, las transiciones y las regiones de fase se alinean correctamente con los picos y valles de la señal, que corresponden a los eventos biomecánicos clave en el ciclo de marcha. De manera similar, la Figura 4.12 compara directamente las etiquetas estimadas por el modelo con las etiquetas reales obtenidas de forma semiautomática y revisadas con la GUI. La alta correspondencia entre ambas señales confirma visualmente el rendimiento del modelo en este escenario de referencia.

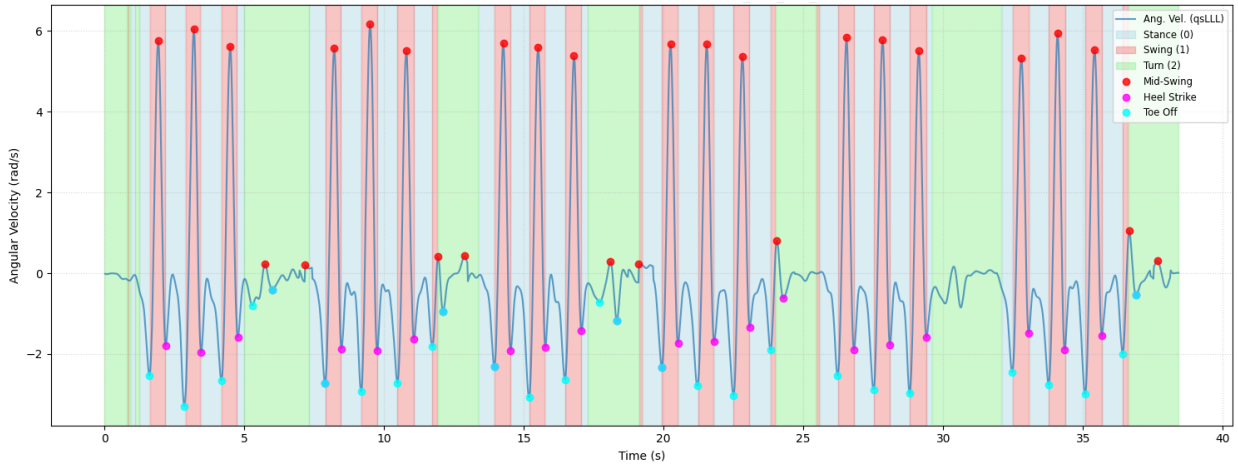


Figura 4.11. Fases estimadas en base a los IMUs representadas sobre la velocidad angular, sujeto S50.

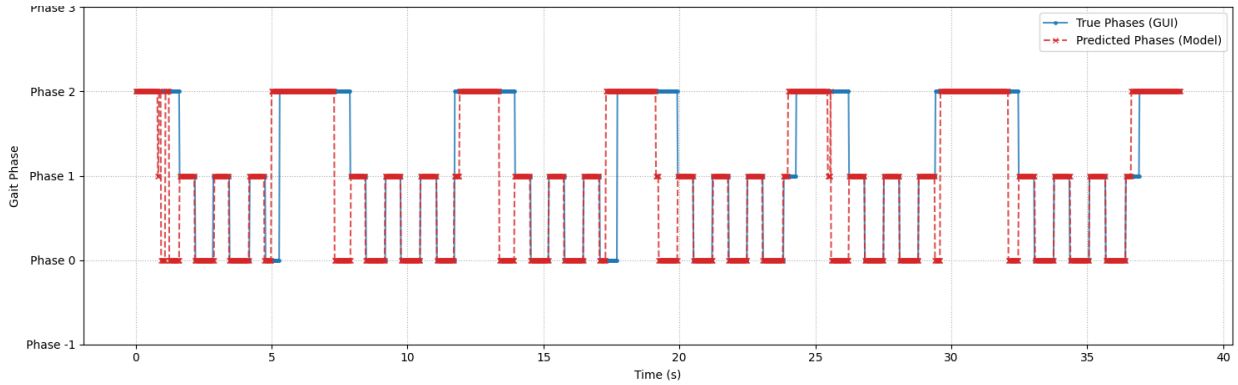


Figura 4.12. Fases estimadas en base a los IMUs vs fases reales, sujeto S50.

La segunda fase de la evaluación consistió en probar el modelo con los ángulos articulares derivados de los distintos sistemas HPE basados en vídeo revisados en la Parte I. Para ello, fue necesario adaptar el BiLSTM a la frecuencia de muestreo del vídeo adquirido en VIDIMU (30 Hz) mediante un remuestreo de los datos etiquetados de marcha. La Figura 4.13 presenta la matriz de

confusión de entrenamiento para los datos angulares etiquetados convertidos a 30 Hz, mostrando de nuevo una correcta capacidad de aprendizaje.

A continuación, se evaluó el rendimiento del modelo en inferencia para cada uno de los cuatro conjuntos de datos angulares, generados por MotionAGFormer, MotionBERT, MMPose y BodyTrack. Las Figuras 4.14 a 4.17 muestran las matrices de confusión correspondientes a dichos conjuntos respectivamente. En general, se observa que el modelo es capaz de generalizar y realizar la segmentación de la marcha a partir de los datos de vídeo, aunque con una precisión inferior a la obtenida con los datos de IMUs, lo que es esperado teniendo en cuenta la reducción en la tasa de muestreo y el ruido adicional. La clase *Turn* (Fase 2) sigue siendo la mejor identificada en la mayoría de los casos, probablemente debido a que los giros presentan un patrón angular muy distintivo y prologando, y a que aparecen con mayor frecuencia.

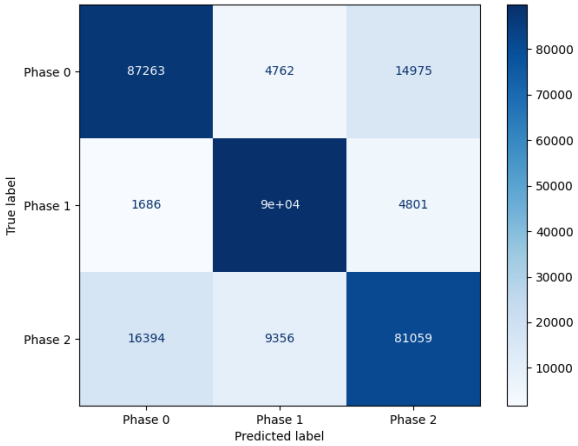


Figura 4.13. Matriz de confusión en entrenamiento (30 Hz).

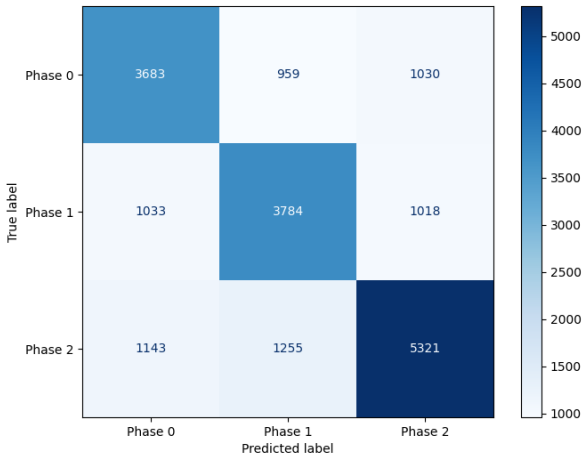


Figura 4.14. Matriz de confusión en inferencia para ángulos obtenidos con MotionAGFormer.

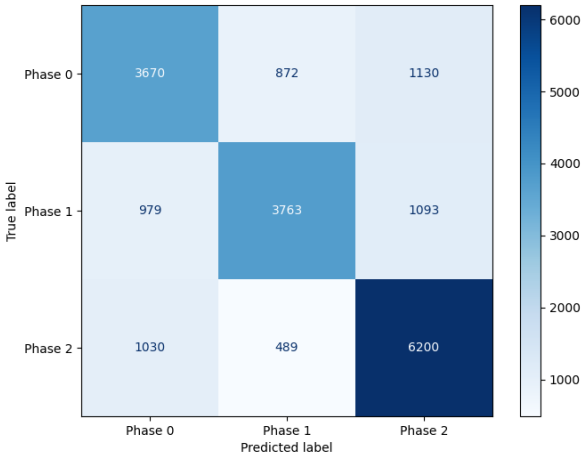


Figura 4.15. Matriz de confusión en inferencia para ángulos obtenidos con MotionBERT.

Figura 4.16. Matriz de confusión en inferencia para ángulos obtenidos con MMPose.

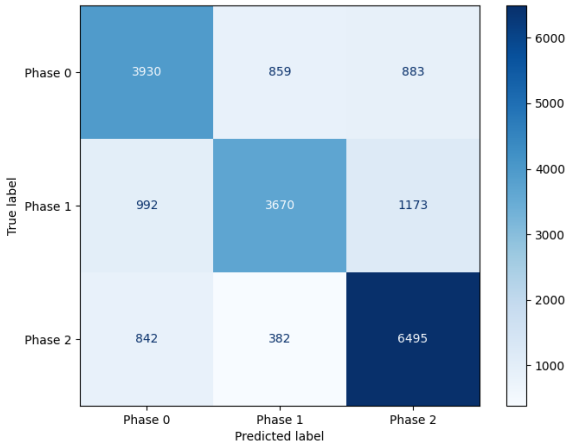


Figura 4.17. Matriz de confusión en inferencia para ángulos obtenidos con BodyTrack.

Al igual que en el caso anterior, la Figura 4.18 y la Figura 4.19 ofrecen un ejemplo visual del rendimiento del modelo, pero esta vez con datos de vídeo. Se observa una correspondencia general entre las fases predichas y los patrones de la señal de velocidad angular, así como con las etiquetas reales. Sin embargo, también se aprecian mayores discrepancias en las transiciones de fase, así como ruido en algunos instantes, lo que refleja las imprecisiones y ambigüedades inherentes a los datos de HPE y que se propagan al modelo de segmentación.

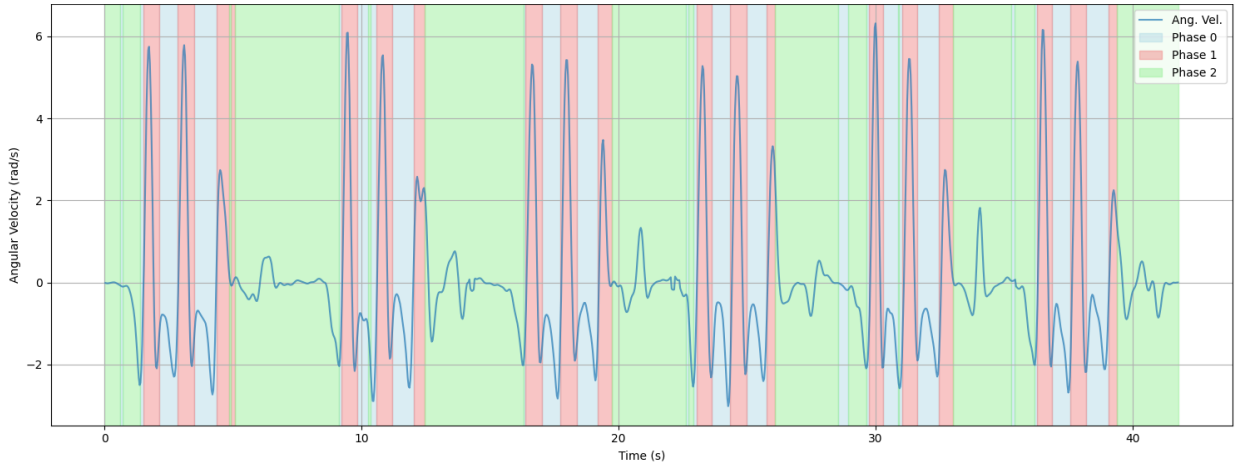


Figura 4.18. Fases estimadas en base al vídeo representadas sobre la velocidad angular, sujeto S42.

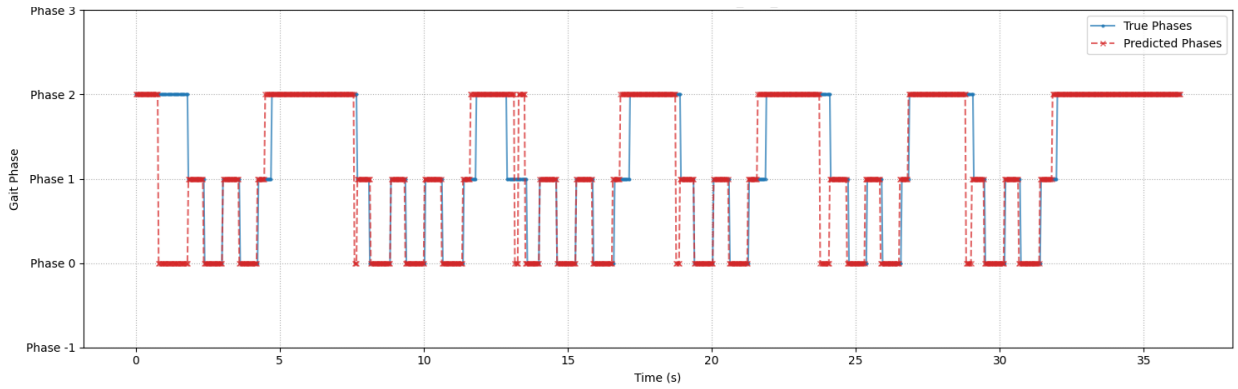


Figura 4.19. Fases estimadas en base al vídeo vs fases reales, sujeto S42.

Finalmente, la Tabla 4.5 desglosa las métricas de *Precision*, *Recall* y *F1-Score* durante el entrenamiento para cada modelo HPE. Como habíamos visto en las matrices de confusión correspondientes a la inferencia, se confirma que la clase *Turn* obtiene consistentemente los valores más altos de F1-Score, superando el 0.70 en todos los casos. Por otro lado, vemos que MotionBERT destacó en la detección de *Swing* (F1-Score = 0.6867) mientras que BodyTrack lo hizo en *Stance* (F1-Score = 0.6873), lo que sugiere que la calidad espacial obtenida por cada pipeline HPE influye de forma diferencial en cada fase de la marcha. En cuanto a las fases de *Stance* y Swing, se observa un rendimiento más variable. Por ejemplo, modelos como MotionBERT y BodyTrack muestran una precisión relativamente alta para la clase *Swing* (0.7344 y 0.7473 respectivamente), aunque con un Recall más bajo, lo que sugiere que cuando predicen dicha clase, tienden a acertar, pero pueden fallar en identificar todas las instancias de esta fase.

Tabla 4.5. Métricas de entrenamiento para cada clase.

Clase	**Precision**	**Recall**	**F1-Score**
MotionAGFormer			
Stance	0.6286	0.6493	0.6388
Swing	0.6309	0.6485	0.6396
Turn	0.7221	0.6893	0.7053
MotionBERT			
Stance	0.6462	0.647	0.6466
Swing	0.7344	0.6449	0.6867
Turn	0.7361	0.8032	0.7682
MMPose			
Stance	0.623	0.6474	0.635
Swing	0.7115	0.6106	0.6572
Turn	0.7108	0.7666	0.7376
BodyTrack			
Stance	0.6818	0.6929	0.6873
Swing	0.7473	0.629	0.683
Turn	0.7596	0.8414	0.7984

4.2.2. Resultados globales

Para obtener una visión comparativa del rendimiento del modelo BiLSTM al ser alimentado con datos de las diferentes arquitecturas de HPE, se agregaron las métricas de evaluación de todos los sujetos y actividades. La Tabla 4.6 resume el rendimiento global de la red para cada uno de los cuatro modelos de estimación de pose, presentando los valores de *Accuracy, Precision, Recall* y *F1-Score*.

Al analizar los resultados globales, se observa que el modelo BiLSTM alcanza el mayor rendimiento cuando se utilizan los datos del sistema BodyTrack, logrando un F1-Score ponderado de 0.7331 y una exactitud del 0.7306. Este resultado es notable, ya que sugiere que, a pesar de que BodyTrack no fue el sistema con el menor RMSE en la Parte I, los patrones temporales de los ángulos articulares que genera son los más consistentes y discriminativos para que la red recurrente aprenda a segmentar las fases de la marcha.

En el segundo lugar se encuentra MotionBERT, con un F1-Score de 0.7076 y una exactitud del 0.7091. Por otro lado, MMPose (F1-Score 0.6829) y MotionAGFormer (F1-Score 0.6657) obtienen un rendimiento ligeramente inferior. En el caso de MotionAGFormer, este resultado es particularmente interesante, ya que a pesar de haber sido el modelo con la mayor precisión en la estimación del ángulo articular en la Parte I, sus datos no resultan ser los más idóneos para esta tarea en específico. No obstante, el intervalo de apenas 6 puntos porcentuales entre el mejor y el peor modelo pone de manifiesto que, incluso con ángulos derivados de redes HPE muy distintas, el algoritmo mantuvo índices de acierto superiores al 65 %, un valor competitivo para una arquitectura LSTM ligera entrenada sobre un conjunto de datos de tamaño reducido.

Tabla 4.6. Rendimiento de la red BiLSTM para los distintos modelos HPE.

Modelo HPE	Accuracy	Precision	Recall	F1-Score
MotionAGFormer	0.6651	0.6668	0.6651	0.6657
MotionBERT	0.7091	0.7091	0.7091	0.7076
MMPose	0.6841	0.6851	0.6841	0.6829
BodyTrack	0.7331	0.7329	0.7331	0.7306

La Figura 4.20 ofrece una representación visual de estos resultados, comparando directamente el F1-Score ponderado y la exactitud para cada modelo HPE. Esto evidencia la brecha de rendimiento con respecto a MMPose y MotionAGFormer. Es interesante notar que los valores de Accuracy son consistentemente similares a los de F1-Score, lo que sugiere que el BiLSTM no presenta un sesgo excesivo hacia la clase mayoritaria y logra un rendimiento balanceado entre las distintas fases.

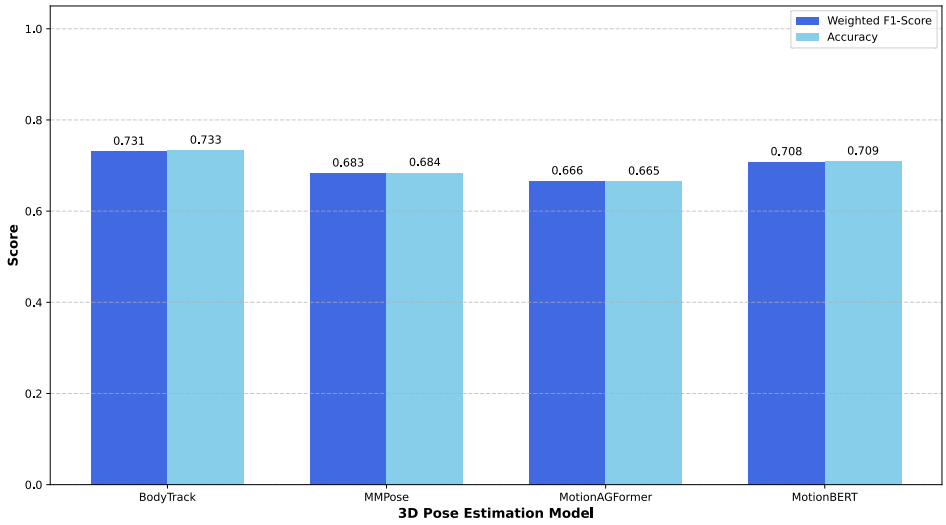


Figura 4.20. F1-Score y accuracy para cada modelo HPE.

4.3. Discusión y limitaciones

4.3.1. Discusión

El análisis de los resultados obtenidos, especialmente cuando se ponen en perspectiva con los hallazgos de la primera parte del trabajo, permite no solo extraer una serie de conclusiones de gran interés, sino también deducir cuáles son aquellos factores que afectan en mayor medida a la aplicación estudiada.

Como era de esperar, el rendimiento del BiLSTM fue superior cuando se entrenó con los ángulos derivados de los IMUs (mostrando un F1-Score aproximadamente un 10% mayor), lo que puede atribuirse a dos factores principales. El primero es la resolución temporal: los datos de los IMUs se procesaron a 50 Hz, mientras que los datos de vídeo estaban limitados a 30 Hz. Este submuestreo efectivo, reduce la "finura" con la que se pueden delimitar las transiciones entre fases, afectando tanto a la precisión de las etiquetas de *ground truth* (el número de muestras etiquetadas se redujo aproximadamente al 60 % del conjunto original) como a la capacidad del modelo para aprender los instantes exactos de los eventos. El segundo factor es el ruido inherente a la estimación. Como se documentó en la Parte I, los ángulos reconstruidos a partir de vídeo, aunque precisos en promedio, contienen un mayor nivel de error y de *jitter* que las señales de los IMUs. Este ruido en la señal de entrada dificulta la tarea de aprendizaje de la red BiLSTM, que debe esforzarse más por discernir el patrón cinemático subyacente. Esto se refleja en las diagonales algo dispersas de las matrices de confusión en las Figuras 4.14 a 4.17. Por otro lado, la fase 2 aparece en un color mucho más intenso que las otras dos debido a que hay un número superior de muestras en el *dataset* que pertenecen a dicha clase, ya que los giros representan una gran cantidad de tiempo respecto al total de cada adquisición.

En cuanto al comportamiento del clasificador, se debe destacar que la implementación de una función de pérdida ponderada y la optimización del F1-score fueron estrategias efectivas para mitigar el sesgo hacia la clase *Turn* durante el entrenamiento (ver Figura 4.13), forzando al modelo a no ignorar las clases minoritarias (*Stance* y *Swing*). Sin embargo, en las matrices de confusión en inferencia (Figuras 4.14 a 4.17) se observa que, aunque el modelo discrimina correctamente, todavía existe una tendencia a clasificar erróneamente muestras de apoyo o balanceo como giro. Este sesgo puede deberse a que las fronteras entre las fases de marcha y los giros son en ocasiones ambiguas. Los pequeños pasos de transición que da el sujeto al iniciar o finalizar un giro comparten características cinemáticas con las fases de apoyo y balanceo, lo que convierte a estas regiones en las más difíciles de discernir para el modelo. En otras palabras, al enfrentarse a las señales más ruidosas, el modelo puede recurrir a clasificar los tramos más inciertos como *Turn*, actuando como una especie de "clase por defecto" para patrones no claros.

Resulta particularmente revelador que MotionAGFormer, el modelo con el mejor rendimiento global en la estimación de ángulos (menor RMSE y mayor correlación) en la Parte I, no produjera los mejores resultados para la segmentación de la marcha en la Parte II. En cambio, BodyTrack, con un rendimiento intermedio en la comparativa, fue el que proporcionó la entrada

más efectiva para el modelo BiLSTM. Esta aparente paradoja subraya una conclusión fundamental: la precisión absoluta de un ángulo en un instante dado en términos geométricos absolutos (medida por el RMSE, MAE, correlación y coeficiente de determinación) no es la única característica relevante para las tareas de análisis de secuencias temporales. Un modelo secuencial como un BiLSTM no solo depende de la exactitud de cada punto, sino que también es especialmente sensible a la consistencia temporal y suavidad de la señal a lo largo del tiempo. Es plausible que, aunque MotionAGFormer tuviera un error promedio más bajo, sus estimaciones pudieran contener un *jitter* de alta frecuencia o artefactos que corrompiesen el patrón dinámico que la red recurrente intenta aprender. Por el contrario, BodyTrack, al ser un producto comercial, podría incorporar algoritmos de filtrado y suavizado más avanzados, y generar una señal temporalmente más consistente y suave, aunque con un posible sesgo o error de *offset* mayor, lo que facilitaría a la BiLSTM la extracción de patrones dinámicos relevantes para la segmentación de fases.

Finalmente, el proceso de etiquetado semiautomático demostró ser al mismo tiempo una limitación y una ventaja. Aunque la corrección manual es laboriosa y susceptible a la subjetividad, la herramienta gráfica desarrollada fue indispensable para crear un conjunto de datos suficientemente limpio como para permitir el entrenamiento del modelo. No obstante, este enfoque pragmático introdujo en algunas ocasiones ambigüedades que luego se trasladaron al BiLSTM, especialmente en las transiciones entre los últimos pasos de cada ciclo y la etapa de giro (ver Figura 4.11).

En definitiva, esta segunda parte del trabajo no solo valida la viabilidad de la segmentación de la marcha mediante un flujo integral basado en aprendizaje profundo, sino que también enriquece las conclusiones de la Parte I. Se ha demostrado que la elección de una arquitectura HPE no debe basarse únicamente en métricas de error estadístico, sino que también debe considerarse la naturaleza de la tarea final. Podemos concluir que, en aplicaciones que dependen del análisis de patrones dinámicos, la consistencia temporal de las señales estimadas puede ser un factor tan crucial como su precisión absoluta a la hora de obtener un buen rendimiento.

4.3.2. Limitaciones

El principal condicionante del estudio fue el tamaño y la naturaleza del conjunto de datos utilizado para el entrenamiento. La base de datos se compone de un número relativamente pequeño de sujetos y ciclos de marcha (los correspondientes a la actividad A01, teniendo en cuenta que se produjeron descartes por errores de adquisición o calibración), lo que impuso restricciones significativas en el diseño de la arquitectura neuronal elegida. Esto obligó a descartar modelos más complejos, como arquitecturas híbridas LSTM-CNN o Transformers, ya que su mayor número de parámetros habría requerido un volumen de datos sustancialmente mayor para un entrenamiento efectivo y una correcta generalización. La elección de una arquitectura BiLSTM relativamente sencilla fue, por tanto, una decisión pragmática para mitigar este riesgo.

Directamente relacionado con lo anterior, el tamaño limitado del *dataset* incrementó el riesgo de sobreajuste. Aunque se implementó un repertorio amplio de técnicas de regularización (como *Dropout, Weight Decay* o *Early Stopping*), el modelo es propenso a memorizar las particularidades de los pocos ejemplos de entrenamiento. Esta circunstancia influyó de manera decisiva en la configuración de hiperparámetros, forzando por ejemplo a elegir un número bajo de capas LSTM con objeto de reducir la complejidad.

Otro desafío significativo fue la sobrerrepresentación de clases. El protocolo para la actividad A01, donde los sujetos caminaban de un lado a otro en el plano sagital, generó una gran cantidad de instantes donde el sujeto llegaba al final del plano y giraba sobre sí mismo. Esto provocó un desequilibrio de clases en el *dataset* durante el etiquetado, de modo que el modelo inicialmente mostraba un sesgo hacia la clase *Turn*. Para abordar este problema, fue necesario implementar una función de pérdida ponderada (*Weighted Loss*). Además, a la hora de configurar los hiperparámetros, la optimización inicial basada en la minimización de la pérdida de validación resultó ser ineficaz, ya que el modelo aprendía rápidamente a reducir la pérdida favoreciendo a la clase más representada, por lo que finalmente se optimizó por maximizar el F1-Score ponderado. Esta métrica es más representativa del rendimiento en conjuntos de datos desbalanceados y condujo hacia soluciones más equilibradas.

El proceso de etiquetado, facilitado por la GUI, se basó parcialmente en la inspección visual y la corrección manual. Este método, aunque necesario dada la ausencia de un *ground truth* absoluto, es susceptible de errores humanos y subjetividad, especialmente en la delimitación precisa de las transiciones entre fases. Una validación concurrente con sistemas *gold standard* como plataformas de fuerza y Vicon habría permitido un etiquetado más objetivo y preciso, aunque esto conllevaría un aumento considerable en el tiempo y el coste de la adquisición de datos, contraviniendo los objetivos de accesibilidad del trabajo.

Finalmente, cabe señalar que para inferir las fases de la marcha sólo se analizaron los ángulos de ambas rodillas obtenidos a través de las adquisiciones de vídeo. Incluir otros planos articulares, como la cadera o el tobillo, habrían permitido capturar la marcha de manera más exhaustiva, entrenar al modelo con datos más representativos sobre el movimiento global del cuerpo, y mejorar la robustez del sistema frente a patrones complejos y patológicos.

5. Conclusiones y líneas futuras

5.1. Consecución de los objetivos

A lo largo del desarrollo de este Trabajo de Fin de Grado, se han perseguido una serie de objetivos específicos con el fin de alcanzar el objetivo global propuesto en la sección de Introducción. A continuación, se detalla el grado de consecución de cada uno de ellos, justificando cómo las diferentes etapas y secciones del trabajo han contribuido a su cumplimiento:

i) Se ha realizado una revisión exhaustiva de la literatura sobre estimación de pose humana y segmentación de la marcha, identificando las arquitecturas de aprendizaje profundo y los enfoques biomecánicos más relevantes.

ii) Se han implementado e integrado tres modelos representativos de estimación de pose, calculando los ángulos articulares a partir de las coordenadas detectadas. Además, se ha diseñado un flujo de procesado para llevar a cabo la sincronización y comparación con la referencia inercial.

iii) Se han discutido las ventajas y limitaciones de cada arquitectura estudiada, proponiendo los casos de uso más adecuados para cada una de ellas.

iv) Se ha desarrollado un sistema que calcula la velocidad angular de la tibia, aplica filtrado de Butterworth y corrige los cambios de signo, complementado con una interfaz gráfica que permite la corrección manual de eventos y la exportación automática de ficheros.

v) Se ha construido un conjunto de datos etiquetado de fases de la marcha a partir de los ángulos articulares procesados, estableciendo un corpus robusto para el entrenamiento del modelo de segmentación.

vi) Se ha desarrollado, entrenado y validado un modelo BiLSTM que segmenta la marcha a partir de series temporales de ángulos articulares, incorporando técnicas de regularización, función de pérdida ponderada y optimización de hiperparámetros automatizada.

vii) Se ha evaluado el modelo desarrollado de forma sistemática, empleando herramientas como el seguimiento de las métricas de pérdida, precisión y F1, así como la generación de matrices de confusión y gráficas de entrenamiento.

viii) Se han analizado los resultados obtenidos, identificando las posibles vías de mejora y las limitaciones del enfoque adoptado. Finalmente, se han propuesto algunas líneas de investigación que puedan servir como punto de partida para futuros trabajos que busquen desarrollar soluciones accesibles para el análisis cinemático.

5.2. Conclusiones

Este Trabajo de Fin de Grado ha permitido extraer conclusiones significativas en sus dos vertientes principales. Por un lado, la evaluación exhaustiva de múltiples arquitecturas de estimación de pose humana en 3D a partir de vídeo monocular ha demostrado que las técnicas de aprendizaje profundo constituyen una alternativa viable y accesible para el análisis cinemático. A

pesar de haber obtenido métricas de error medio comparables a las de los sensores inerciales, se ha constatado que no existe una única arquitectura universalmente superior, sino que el rendimiento varía en función de la actividad específica y la articulación analizada. Aunque los sistemas HPE basados en vídeo presentan limitaciones, especialmente en la captura de movimientos complejos o con oclusiones severas frente a la robustez de los IMUs, su carácter no intrusivo y de bajo coste los posiciona como herramientas prometedoras para la democratización del análisis de movimiento.

En paralelo, se ha comprobado que las series temporales de ángulos articulares proporcionan información suficiente como para segmentar las fases de la marcha a través de una red BiLSTM. El proceso de obtención de etiquetas de fase mediante el procesado de señales de velocidad angular de IMUs y el uso de una herramienta de etiquetado gráfico desarrollada ad-hoc permitió generar un conjunto de datos de calidad para el entrenamiento. El modelo entrenado demostró una capacidad prometedora para identificar correctamente las fases de *Stance*, *Swing* y *Turn* a partir de las secuencias de los ángulos de rodilla, indicando la viabilidad de un enfoque basado en aprendizaje profundo y datos cinemáticos. Si bien los resultados son optimistas, también sugieren que existen áreas de mejora, como la ampliación del conjunto de datos o la exploración de características de entrada más amplias para mejorar la generalización y el comportamiento frente a patrones de marcha atípicos o patológicos.

De forma global, este trabajo subraya el potencial de las técnicas de aprendizaje profundo para abordar problemas complejos en el ámbito de la biomecánica y el análisis del movimiento humano. A este respecto, se ha desarrollado un flujo de trabajo íntegramente basado en software que elimina marcadores físicos, reduce la logística de las adquisiciones y acerca la evaluación de la marcha a entornos no controlados como podría ser el domicilio del paciente. Las limitaciones expuestas representan oportunidades para la investigación futura, pero no invalidan la conclusión principal: la unión de visión artificial y redes recurrentes constituye una alternativa realista, rentable y escalable a los sistemas de laboratorio tradicionales para el análisis clínico de la marcha.

5.3. Líneas futuras

Este trabajo ha sentado las bases para diversas líneas de investigación futuras que podrían expandir y mejorar significativamente los resultados obtenidos. Una dirección prioritaria es la ampliación y diversificación del conjunto de datos VIDIMU. El número actual de sujetos y recorridos es suficiente para una prueba de concepto, pero no para cubrir la enorme variabilidad inter e intrasujeto que aparece en la práctica. Sería beneficioso incrementar el número de participantes y adquisiciones, incluyendo una mayor diversidad antropométrica, para mejorar la generalización de los modelos. Además, el *dataset* podría enriquecerse con adquisiciones de diversas patologías que afecten a la marcha. Esto permitiría entrenar y validar modelos específicamente diseñados para la detección y caracterización de patrones de marcha alterados, aumentando su relevancia clínica. Por otro lado, la grabación de actividades en entornos más

variados, como domicilios o exteriores, permitiría evaluar la robustez de los sistemas en condiciones menos controladas.

En cuanto a las tecnologías de adquisición de datos, se propone la comparación de los modelos HPE no solo con IMUs, sino también con *gold standards* como sistemas Vicon y plataformas de fuerza, lo que proporcionaría una validación cinemática y cinética más rigurosa. También sería interesante evaluar el rendimiento de los modelos HPE utilizando configuraciones de vídeo multicámara para mitigar oclusiones y mejorar la estimación de la profundidad. Para el refinamiento y expansión del modelo de segmentación de la marcha se podría investigar el impacto de incluir ángulos articulares adicionales (cadera, tobillo, tronco) como entrada al modelo LSTM. De cara a la transferencia tecnológica, resulta interesante explorar la inferencia en tiempo real en el "borde" de la red sobre dispositivos embebidos o teléfonos inteligentes, lo que exige trabajar en técnicas de destilación y cuantización de modelos.

Finalmente, la continua evolución del campo requiere la incorporación al *benchmark* de arquitecturas de visión más recientes, incluyendo nuevos modelos basados en difusión, enfoques híbridos y sistemas multimodales que combinen información de múltiples fuentes (vídeo, IMUs, sensores médicos, etc.). A este respecto, también es prometedora la investigación en modelos personalizados que se adapten rápidamente a nuevos usuarios o a cambios en su patrón de marcha. Por otro lado, técnicas de frontera como el aprendizaje autosupervisado o el aprendizaje por refuerzo podrían utilizarse para pre-entrenar modelos sin necesidad de etiquetas, y los modelos generativos podrían emplearse para la aumentación de datos. La aplicación de técnicas de interpretabilidad (*Explainable AI,* XAI) ayudaría a comprender las decisiones y el funcionamiento intrínseco de las redes neuronales utilizadas. Por último, cabe destacar que, más allá de las decisiones técnicas, la colaboración con profesionales clínicos para aplicar estas herramientas en estudios piloto con pacientes reales (por ejemplo, para el seguimiento de la rehabilitación), es fundamental para el desarrollo eficiente e iterativo de este campo de investigación.

Referencias

3.1. Cross-validation: Evaluating estimator performance. (s. f.). Scikit-Learn. Recuperado 25 de mayo de 2025, de https://scikit-learn/stable/modules/cross_validation.html

3D Human Pose Estimation Experiments and Analysis. (s. f.). KDnuggets. Recuperado 28 de abril de 2025, de https://www.kdnuggets.com/3d-human-pose-estimation-experiments-and-analysis

Andriluka, M., Iqbal, U., Insafutdinov, E., Pishchulin, L., Milan, A., Gall, J., & Schiele, B. (2018). *PoseTrack: A Benchmark for Human Pose Estimation and Tracking* (No. arXiv:1710.10000). arXiv. https://doi.org/10.48550/arXiv.1710.10000

Armand, S., Decoulon, G., & Bonnefoy-Mazure, A. (2016). *Gait analysis in children with cerebral palsy*. https://eor.bioscientifica.com/view/journals/eor/1/12/2058-5241.1.000052.xml

Attal, F., Amirat, Y., Chibani, A., & Mohammed, S. (2018). Automatic Recognition of Gait Phases Using a Multiple-Regression Hidden Markov Model. *IEEE/ASME Transactions on Mechatronics*, *23*(4), 1597-1607. https://doi.org/10.1109/TMECH.2018.2836934

Ben Gamra, M., & Akhloufi, M. A. (2021). A review of deep learning techniques for 2D and 3D human pose estimation. *Image and Vision Computing*, *114*, 104282. https://doi.org/10.1016/j.imavis.2021.104282

Caldas, R., Mundt, M., Potthast, W., Buarque de Lima Neto, F., & Markert, B. (2017). A systematic review of gait analysis methods based on inertial sensors and adaptive algorithms. *Gait & Posture*, *57*, 204-210. https://doi.org/10.1016/j.gaitpost.2017.06.019

Caramia, F., D'Angelantonio, E., Lucangeli, L., & Camomilla, V. (2025). Validation of Low-Cost IMUs for Telerehabilitation Exercises. *Sensors*, *25*(10), Article 10. https://doi.org/10.3390/s25103129

Chen, K., Gabriel, P., Alasfour, A., Gong, C., Doyle, W. K., Devinsky, O., Friedman, D., Dugan, P., Melloni, L., Thesen, T., Gonda, D., Sattar, S., Wang, S., & Gilja, V. (2018). Patient-Specific Pose Estimation in Clinical Environments. *IEEE Journal of Translational Engineering in Health and Medicine*, *6*, 1-11. IEEE Journal of Translational Engineering in Health and Medicine. https://doi.org/10.1109/JTEHM.2018.2875464

Chen, Y., Tian, Y., & He, M. (2020). Monocular human pose estimation: A survey of deep learning-based methods. *Computer Vision and Image Understanding*, *192*, 102897. https://doi.org/10.1016/j.cviu.2019.102897

Confusion Matrix: How To Use It & Interpret Results [Examples]. (s. f.). Recuperado 25 de mayo de 2025, de https://www.v7labs.com/blog/confusion-matrix-guide

CS 230—Recurrent Neural Networks Cheatsheet. (s. f.). Recuperado 15 de junio de 2025, de https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* (No. arXiv:1810.04805). arXiv. https://doi.org/10.48550/arXiv.1810.04805

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, *36*(4), 193-202. https://doi.org/10.1007/BF00344251

Grimmer, M., Schmidt, K., Duarte, J. E., Neuner, L., Koginov, G., & Riener, R. (2019). Stance and Swing Detection Based on the Angular Velocity of Lower Limb Segments During Walking. *Frontiers in Neurorobotics*, *13*. https://doi.org/10.3389/fnbot.2019.00057

Hollman, J. H., McDade, E. M., & Petersen, R. C. (2011). Normative spatiotemporal gait parameters in older adults. *Gait & Posture*, *34*(1), 111-118. https://doi.org/10.1016/j.gaitpost.2011.03.024

Ienaga, N., Takahata, S., Terayama, K., Enomoto, D., Ishihara, H., Noda, H., & Hagihara, H. (2022). Development and Verification of Postural Control Assessment Using Deep-Learning-Based Pose Estimators: Towards Clinical Applications. *Occupational Therapy International*, *2022*(1), 6952999. https://doi.org/10.1155/2022/6952999

Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., & Schiele, B. (2016). *DeeperCut: A Deeper, Stronger, and Faster Multi-Person Pose Estimation Model* (No. arXiv:1605.03170). arXiv. https://doi.org/10.48550/arXiv.1605.03170

Ionescu, C., Papava, D., Olaru, V., & Sminchisescu, C. (2014). Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *36*(7), 1325-1339. https://doi.org/10.1109/TPAMI.2013.248

Jiang, B., Xu, F., Zhang, Z., Tang, J., & Nie, F. (2023). *AGFormer: Efficient Graph Representation with Anchor-Graph Transformer* (No. arXiv:2305.07521). arXiv. https://doi.org/10.48550/arXiv.2305.07521

Kanko, R. M., Laende, E. K., Davis, E. M., Selbie, W. S., & Deluzio, K. J. (2021). Concurrent assessment of gait kinematics using marker-based and markerless motion capture. *Journal of Biomechanics*, *127*, 110665. https://doi.org/10.1016/j.jbiomech.2021.110665

Khokhlova, M., Migniot, C., Morozov, A., Sushkova, O., & Dipanda, A. (2019). Normal and pathological gait classification LSTM model. *Artificial Intelligence in Medicine*, *94*, 54-66. https://doi.org/10.1016/j.artmed.2018.12.007

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, *25*. https://proceedings.neurips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

Lee, Y., Lama, B., Joo, S., & Kwon, J. (2024). Enhancing Human Key Point Identification: A Comparative Study of the High-Resolution VICON Dataset and COCO Dataset Using BPNET. *Applied Sciences*, *14*(11), Article 11. https://doi.org/10.3390/app14114351

Li, S., Farha, Y. A., Liu, Y., Cheng, M.-M., & Gall, J. (2020). *MS-TCN++: Multi-Stage Temporal Convolutional Network for Action Segmentation* (No. arXiv:2006.09220). arXiv. https://doi.org/10.48550/arXiv.2006.09220

Liu, T., Inoue, Y., & Shibata, K. (2010). A wearable force plate system for the continuous measurement of triaxial ground reaction force in biomechanical applications. *Measurement Science and Technology*, *21*(8), 085804. https://doi.org/10.1088/0957-0233/21/8/085804

Loshchilov, I., & Hutter, F. (2019). *Decoupled Weight Decay Regularization* (No. arXiv:1711.05101). arXiv. https://doi.org/10.48550/arXiv.1711.05101

Luo, G., Zhu, Y., Wang, R., Tong, Y., Lu, W., & Wang, H. (2020). Random forest–based classsification and analysis of hemiplegia gait using low-cost depth cameras. *Medical & Biological Engineering & Computing*, *58*(2), 373-382. https://doi.org/10.1007/s11517-019-02079-7

Mahmood, N., Ghorbani, N., Troje, N. F., Pons-Moll, G., & Black, M. J. (2019). *AMASS: Archive of Motion Capture as Surface Shapes* (No. arXiv:1904.03278). arXiv. https://doi.org/10.48550/arXiv.1904.03278

Martínez-Zarzuela, M., González-Alonso, J., Antón-Rodríguez, M., Díaz-Pernas, F. J., Müller, H., & Simón-Martínez, C. (2023). Multimodal video and IMU kinematic dataset on daily life activities using affordable devices. *Scientific Data*, *10*(1), 648. https://doi.org/10.1038/s41597-023-02554-9

Medrano-Paredes, M., Fernández-González, C., Díaz-Pernas, F.-J., Saoudi, H., González-Alonso, J., & Martínez-Zarzuela, M. (2025). *Paving the Way Towards Kinematic Assessment Using Monocular Video: A Benchmark of State-of-the-Art Deep-Learning-Based 3D Human Pose Estimators Against Inertial Sensors in Daily Living Activities* [Dataset]. Zenodo. https://zenodo.org/records/15088423

Mehraban, S., Adeli, V., & Taati, B. (s. f.). *MotionAGFormer: Enhancing 3D Human Pose Estimation with a Transformer-GCNFormer Network*. https://github.com/

Mehta, D., Rhodin, H., Casas, D., Fua, P., Sotnychenko, O., Xu, W., & Theobalt, C. (2017). *Monocular 3D Human Pose Estimation In The Wild Using Improved CNN Supervision* (No. arXiv:1611.09813). arXiv. https://doi.org/10.48550/arXiv.1611.09813

Morris, M. E., Huxham, F., McGinley, J., Dodd, K., & Iansek, R. (2001). The biomechanics and motor control of gait in Parkinson disease. *Clinical Biomechanics*, *16*(6), 459-470. https://doi.org/10.1016/S0268-0033(01)00035-3

Overview—MMPose 1.3.2 documentation. (s. f.). Recuperado 13 de mayo de 2025, de https://mmpose.readthedocs.io/en/latest/overview.html

Pang, H. E., Cai, Z., Yang, L., Zhang, T., & Liu, Z. (2022). *Benchmarking and Analyzing 3D Human Pose and Shape Estimation Beyond Algorithms* (No. arXiv:2209.10529). arXiv. https://doi.org/10.48550/arXiv.2209.10529

Papers with Code—3D Human Pose Estimation. (s. f.). Recuperado 15 de mayo de 2025, de https://paperswithcode.com/task/3d-human-pose-estimation

Park, S., Hwang, J., & Kwak, N. (2016). 3D Human Pose Estimation Using Convolutional Neural Networks with 2D Pose Information. En G. Hua & H. Jégou (Eds.), *Computer Vision – ECCV 2016 Workshops* (pp. 156-169). Springer International Publishing. https://doi.org/10.1007/978-3-319-49409-8_15

Pavllo, D., Feichtenhofer, C., Grangier, D., & Auli, M. (2019). *3D Human Pose Estimation in Video With Temporal Convolutions and Semi-Supervised Training*. 7753-7762. https://openaccess.thecvf.com/content_CVPR_2019/html/Pavllo_3D_Human_Pose_Estimation_in_Video_With_Temporal_Convolutions_and_CVPR_2019_paper.html

Prakash, C., Kumar, R., & Mittal, N. (2018). Recent developments in human gait research: Parameters, approaches, applications, machine learning techniques, datasets and challenges. *Artificial Intelligence Review*, *49*(1), 1-40. https://doi.org/10.1007/s10462-016-9514-6

Press release: The Nobel Prize in Physics 2024. (s. f.). NobelPrize.Org. Recuperado 23 de abril de 2025, de https://www.nobelprize.org/prizes/physics/2024/press-release/

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (s. f.). *Improving Language Understanding by Generative Pre-Training*.

Rajagopal, A., Dembia, C. L., DeMers, M. S., Delp, D. D., Hicks, J. L., & Delp, S. L. (2016). Full-Body Musculoskeletal Model for Muscle-Driven Simulation of Human Gait. *IEEE Transactions on Bio-Medical Engineering*, *63*(10), 2068-2079. https://doi.org/10.1109/TBME.2016.2586891

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386-408. https://doi.org/10.1037/h0042519

Salminen, M., Perttunen, J., Avela, J., & Vehkaoja, A. (2024). A novel method for accurate division of the gait cycle into seven phases using shank angular velocity. *Gait & Posture*, *111*, 1-7. https://doi.org/10.1016/j.gaitpost.2024.04.006

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2009). The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, *20*(1), 61-80. https://doi.org/10.1109/TNN.2008.2005605

Shi, X., Wang, Z., Zhao, H., Qiu, S., Liu, R., Lin, F., & Tang, K. (2023). Threshold-Free Phase Segmentation and Zero Velocity Detection for Gait Analysis Using Foot-Mounted Inertial

Sensors. *IEEE Transactions on Human-Machine Systems*, *53*(1), 176-186. https://doi.org/10.1109/THMS.2022.3228515

Smirnova, V., Khamatnurova, R., Kharin, N., Yaikova, E., Baltina, T., & Sachenkov, O. (2022). The Automatization of the Gait Analysis by the Vicon Video System: A Pilot Study. *Sensors*, *22*(19), Article 19. https://doi.org/10.3390/s22197178

Su, B., & Gutierrez-Farewik, E. M. (2020). Gait Trajectory and Gait Phase Prediction Based on an LSTM Network. *Sensors*, *20*(24), Article 24. https://doi.org/10.3390/s20247127

Toshev, A., & Szegedy, C. (2014). DeepPose: Human Pose Estimation via Deep Neural Networks. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 1653-1660. https://doi.org/10.1109/CVPR.2014.214

Vafadar, S., Skalli, W., Bonnet-Lebrun, A., Assi, A., & Gajny, L. (2022). Assessment of a novel deep learning-based marker-less motion capture system for gait study. *Gait & Posture*, *94*, 138-143. https://doi.org/10.1016/j.gaitpost.2022.03.008

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). *Attention Is All You Need* (No. arXiv:1706.03762). arXiv. https://doi.org/10.48550/arXiv.1706.03762

Wagenaar, R. C., & van Emmerik, R. E. A. (1994). Dynamics of pathological gait. *Human Movement Science*, *13*(3), 441-471. https://doi.org/10.1016/0167-9457(94)90049-3

Wang, J., Tan, S., Zhen, X., Xu, S., Zheng, F., He, Z., & Shao, L. (2021). Deep 3D human pose estimation: A review. *Computer Vision and Image Understanding*, *210*, 103225. https://doi.org/10.1016/J.CVIU.2021.103225

Watanabe, S., Bansal, A., & Hutter, F. (2023). *PED-ANOVA: Efficiently Quantifying Hyperparameter Importance in Arbitrary Subspaces* (No. arXiv:2304.10255). arXiv. https://doi.org/10.48550/arXiv.2304.10255

Windolf, M., Götzen, N., & Morlock, M. (2008). Systematic accuracy and precision analysis of video motion capturing systems—Exemplified on the *Vicon-460* system. *Journal of Biomechanics*, *41*(12), 2776-2780. https://doi.org/10.1016/j.jbiomech.2008.06.024

Xu, Y., Zhang, J., Zhang, Q., & Tao, D. (2022). ViTPose: Simple Vision Transformer Baselines for Human Pose Estimation. *Advances in Neural Information Processing Systems*, *35*, 38571-38584.

Xue, Z., Ming, D., Song, W., Wan, B., & Jin, S. (2010). Infrared gait recognition based on wavelet transform and support vector machine. *Pattern Recognition*, *43*(8), 2904-2910. https://doi.org/10.1016/j.patcog.2010.03.011

Yu, J., Zhang, S., Wang, A., & Li, W. (2020). Human gait analysis based on OpenSim. *2020 International Conference on Advanced Mechatronic Systems (ICAMechS)*, 278-281. https://doi.org/10.1109/ICAMechS49982.2020.9310111

Zafra-Palma, J., Marín-Jiménez, N., Castro-Piñero, J., Cuenca-García, M., Muñoz-Salinas, R., & Marín-Jiménez, M. J. (2025). Health & Gait: A dataset for gait-based analysis. *Scientific Data*, *12*(1), 44. https://doi.org/10.1038/s41597-024-04327-4

Zhang, S., Li, X., Hu, C., Xu, J., & Liu, H. (2024). DSTFormer: 3D Human Pose Estimation with a Dual-scale Spatial and Temporal Transformer Network. *2024 International Conference on Advanced Robotics and Mechatronics (ICARM)*, 484-489. https://doi.org/10.1109/ICARM62033.2024.10715863

Zheng, C. E., Wu, W., Chen, C., Shah, M., Zheng, C., Yang, T., Zhu, S., Shen, J., & Kehtarnavaz, N. (2018). Deep Learning-Based Human Pose Estimation: A Survey. *J. ACM*, *37*(111), 45469. https://doi.org/10.1145/1122445.1122456

Zheng, C., Zhu, S., Mendieta, M., Yang, T., Chen, C., & Ding, Z. (2021). *3D Human Pose Estimation With Spatial and Temporal Transformers*. 11656-11665. https://openaccess.thecvf.com/content/ICCV2021/html/Zheng_3D_Human_Pose_Estimation_With_Spatial_and_Temporal_Transformers_ICCV_2021_paper.html

Zhu, W., Wu, W., & Wang, Y. (2022). MotionBERT: Unified Pretraining for Human Motion Analysis. *arXiv*.