

Uso de un modelo simplificado de músculo para determinar la cinemática muscular en tiempo real

Autor:
Lucas Porzio González

Tutores:
Alberto Luaces Fernández
Florian Michaud

Grado en Ingeniería Mecánica

Julio 2026

Laboratorio de Ingeniería Mecánica,
CITENI, Campus Industrial de Ferrol,
Universidade da Coruña, Ferrol, España



CITENI



UNIVERSIDADE DA CORUÑA

Índice

Objeto

Alcance

Introducción

Metodología

Análisis de resultados

Conclusiones



CITENI



CAMPUSINDUSTRIAL



UNIVERSIDADE DA CORUÑA

1. Objeto

Motivación principal: Reducir el coste computacional asociado al calculo de parámetros musculotendinosos al tener en cuenta el fenómeno del **wrapping**.

Objetivo ➡ **biorretroalimentación**

Conocer en tiempo real el estado muscular permitiría adaptar el entrenamiento, **optimizar el rendimiento y reducir el riesgo de lesión.**

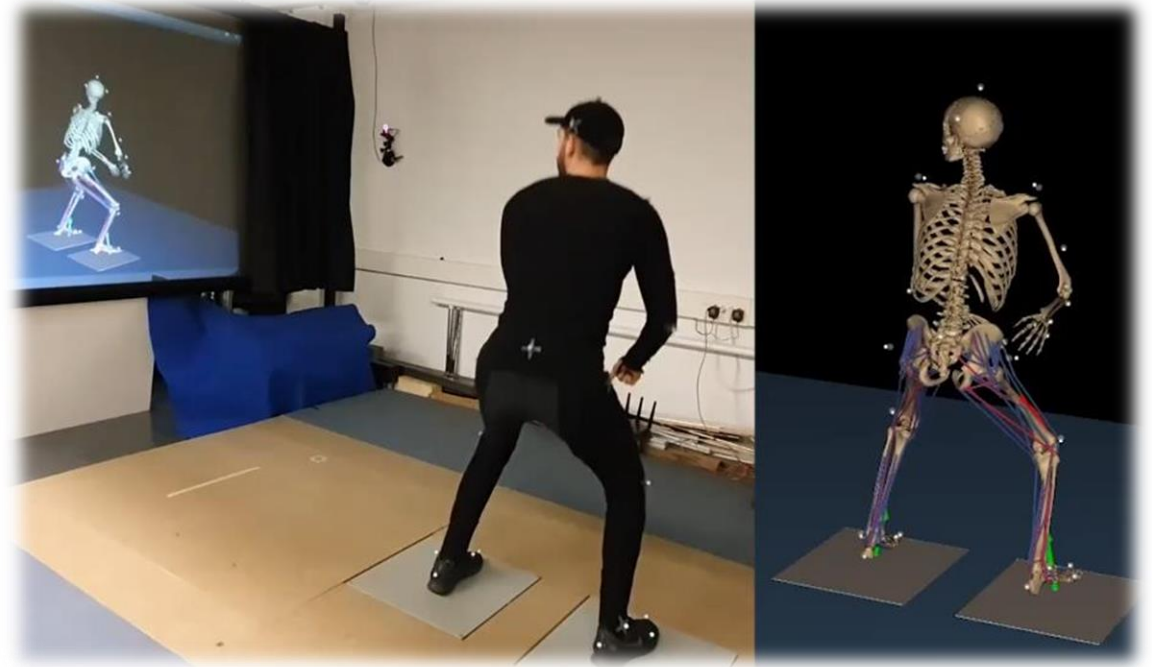


2. Alcance

Implementación de un **flujo de simulación** musculoesquelética que integre el análisis multicuerpo con distintos modelos para estimar de forma eficiente la cinemática y dinámica muscular

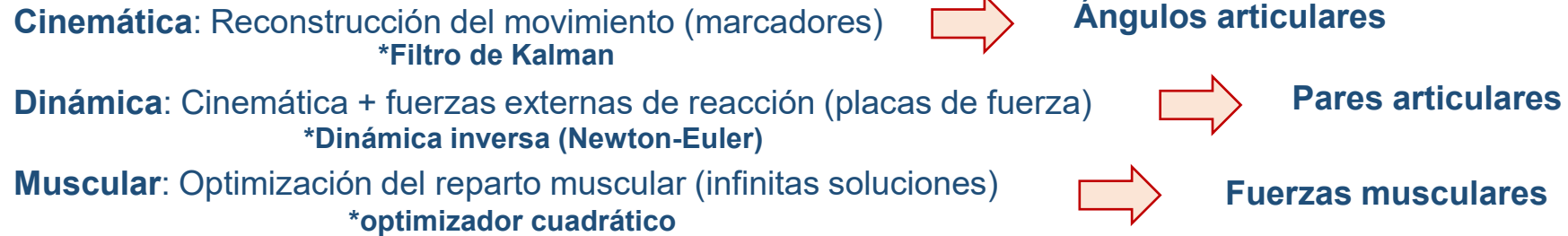
El desarrollo queda **restringido al tren inferior del cuerpo humano**, empleándose como prueba de concepto previa a una futura implementación en tiempo real.

En esta fase, el flujo desarrollado **funciona fuera del bucle temporal de captura**, permitiendo validar la metodología y evaluar su viabilidad computacional.

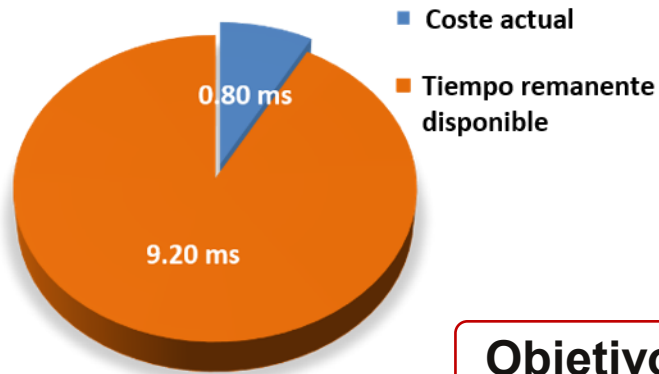


3. Introducción

Situación actual: (El software de captura de movimiento actual del laboratorio resuelve...)



Mejoras a añadir: Problema de implementación en tiempo real



Objetivo: resolver wrapping, modelo fisiológico y fatiga en < 9,2 ms / fotograma

*Frecuencia de muestreo 100Hz



CITENI



CAMPUS INDUSTRIAL

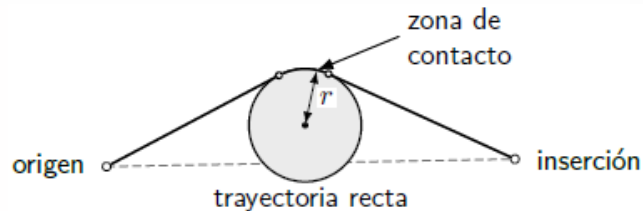


UNIVERSIDADE DA CORUÑA

1. Problema geométrico



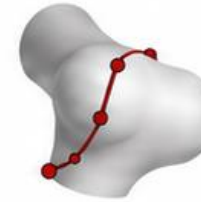
- La trayectoria muscular no siempre es una línea recta entre origen e inserción.
- El *wrapping* modifica la longitud musculotendinosa, los brazos de momento y la línea de acción.



2. Métodos clásicos

A) NGV (Natural Geodesic Variations)

- Cálculo geométrico preciso sobre superficies anatómicas.
- Elevado coste computacional.



B) Multidimensional cubic splines

- Aproximación mediante interpolación a partir de datos precomputados.
- Más eficiente, pero con limitaciones de escalabilidad.

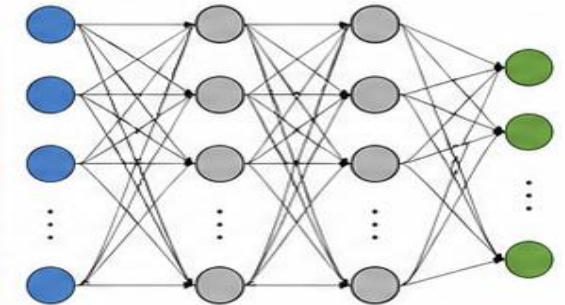


Ambos métodos permiten resolver el problema, pero dificultan la implementación en tiempo real.

3. Solución propuesta

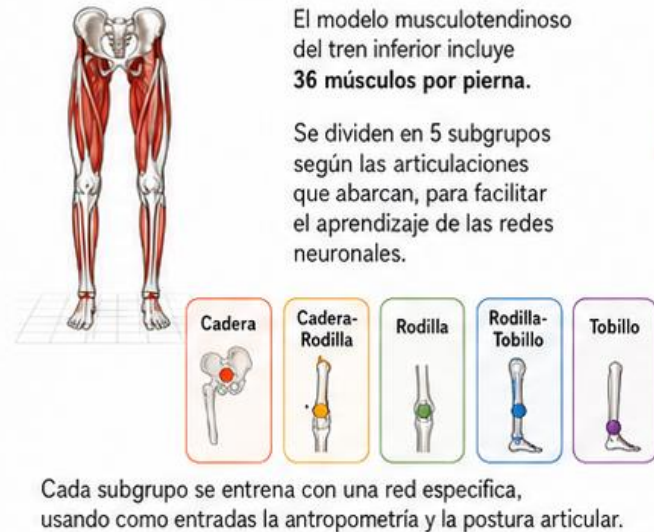
Redes neuronales *feed-forward*

- Se proponen como solución al cuello de botella computacional asociado al cálculo directo del *wrapping*.
- Se eligen redes *feed-forward* porque el problema es puramente geométrico.



Propuesta: utilizar **FFNN** como modelo sustituto del *wrapping* para resolver el **cuello de botella computacional** y facilitar su uso en **tiempo real**.

1. División muscular en subgrupos



El modelo musculotendinoso del tren inferior incluye **36 músculos por pierna**.

Se dividen en 5 subgrupos según las articulaciones que abarcan, para facilitar el aprendizaje de las redes neuronales.

Cadera, Cadera-Rodilla, Rodilla, Rodilla-Tobillo, Tobillo

Cada subgrupo se entrena con una red específica, usando como entradas la antropometría y la postura articular.

2. Generación de configuraciones

Factores de escala antropométricos

Configuraciones articulares

1000 modelos escalados

500 posturas articulares

$1000 \times 500 = 500.000$ posiciones

3. Resolución con OpenSim

Cada posición se evalúa en OpenSim para obtener las variables musculotendinosas.

Variables obtenidas

Longitud musculotendinosa (μ^{MT})

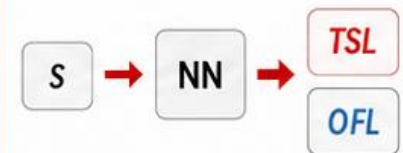
Brazos de momento (r)

OpenSim

Esto produce la base de datos que utilizarán las redes neuronales.

Red paralela para parámetros dependientes de la antropometría

En paralelo, se entrena una sexta red para estimar parámetros que dependen solo de la antropometría del sujeto.



- **TSL**: Longitud del tendón en reposo
- **OFL**: Longitud óptima de la fibra

Permite obtener parámetros individuales del sujeto, independientes de la postura.

4. Base de datos generada

Se obtienen 500.000 pares de entrada-salida.

Entradas: ángulos articulares (q) + factores de escala (s)										Salidas: μ^{MT}, r	
q_1	q_2	...	q_n	s_1	s_2	...	s_m			μ^{MT}	r
q_1^1	q_2^1	...	q_n^1	s_1^1	s_2^1	...	s_m^1			μ_1^{MT}	r_1
q_1^2	q_2^2	...	q_n^2	s_1^2	s_2^2	...	s_m^2			μ_2^{MT}	r_2
...	...	...	...	...	...	...	...			...	...
q_1^N	q_2^N	...	q_n^N	s_1^N	s_2^N	...	s_m^N			μ_N^{MT}	r_N

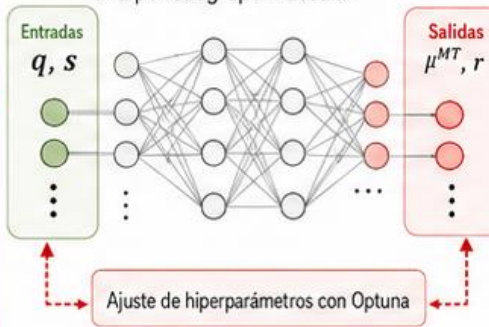
$N = 500.000$ posiciones

5. División del dataset



6. Entrenamiento de las redes

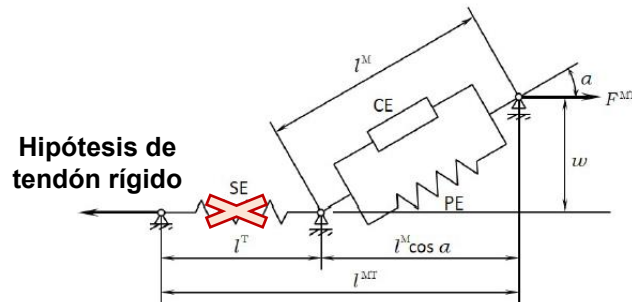
Se entrenan 5 redes feed-forward, una por subgrupo muscular.



La división en subgrupos facilita el aprendizaje de la relación entre la antropometría, los ángulos articulares y las variables musculotendinosas.

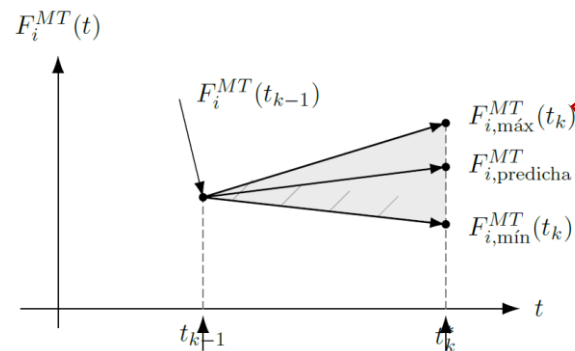
4.3 Límites de generación de fuerza

Modelo muscular de Hill

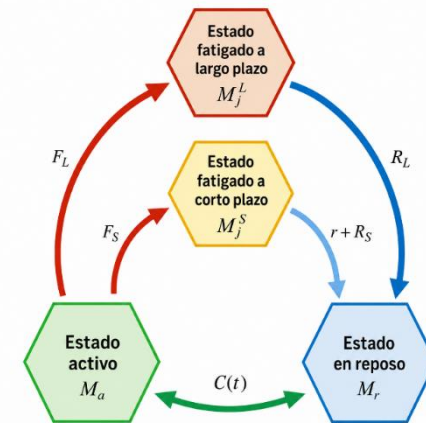


$$l^{MT} = l^T + l^M \cos(\alpha)$$

$$F^M = F_0^M \cos(\alpha) [af_l(l^M)f_v(v^M) + f_{PE}(l^M)]$$

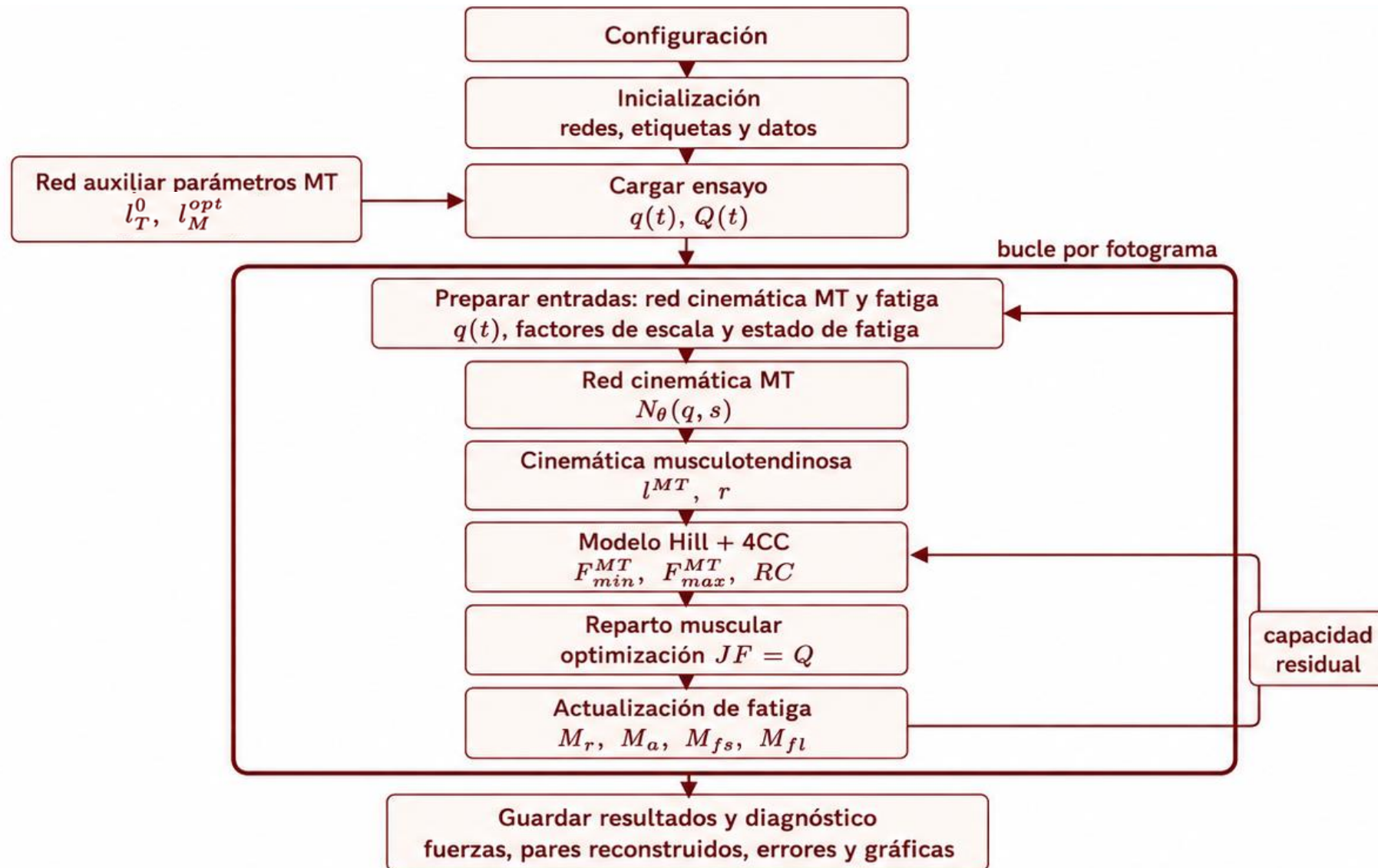


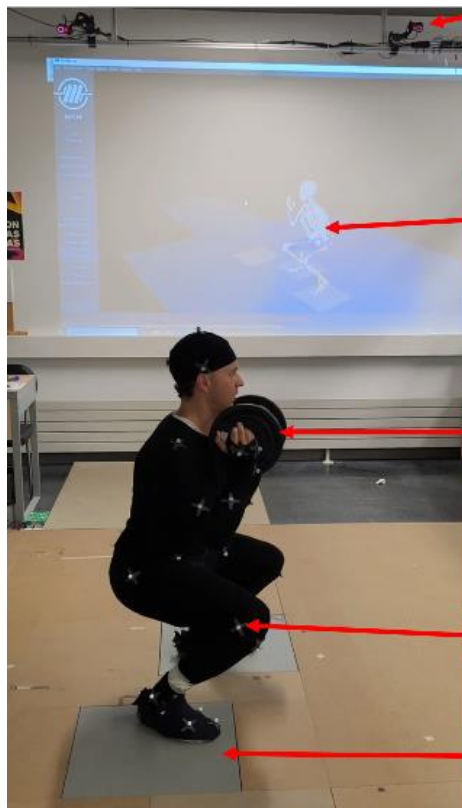
Modelo de fatiga muscular 4CC



$$M_a + M_r + M_f^L + M_f^S = 100\%$$

$$RC_i = 1 - \frac{M_{f,i}^S + M_{f,i}^L}{100}$$





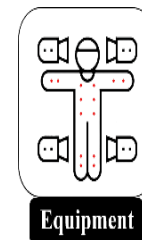
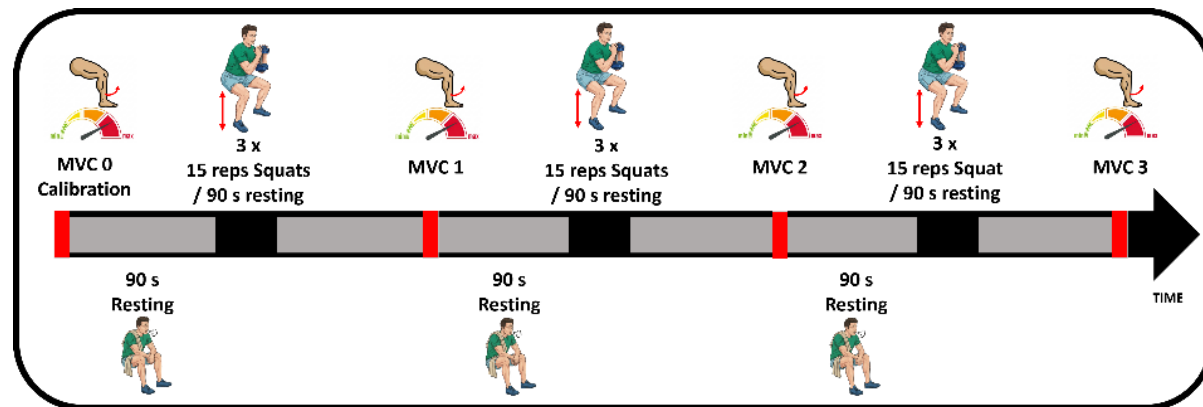
Cámaras infrarojos

Dinámica inversa
en tiempo real

Pesa

Marcadores ópticos

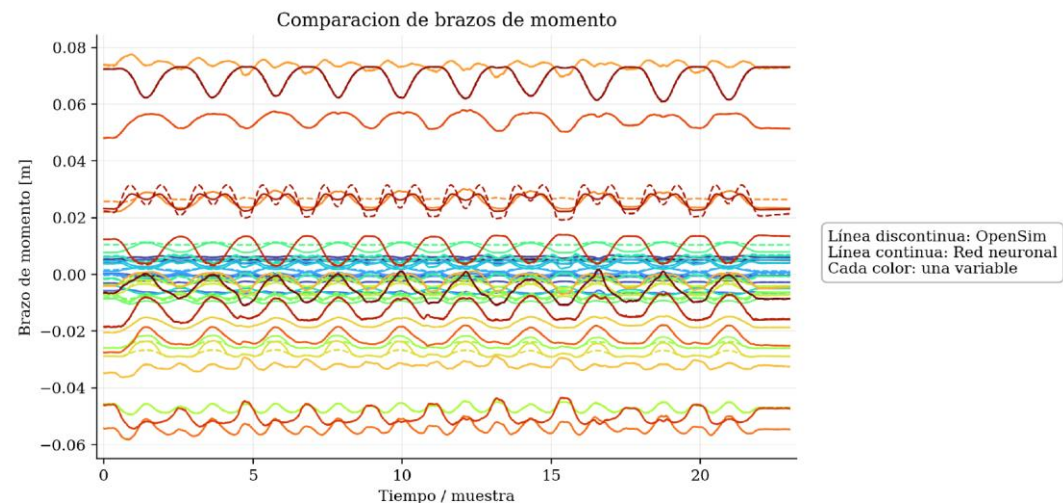
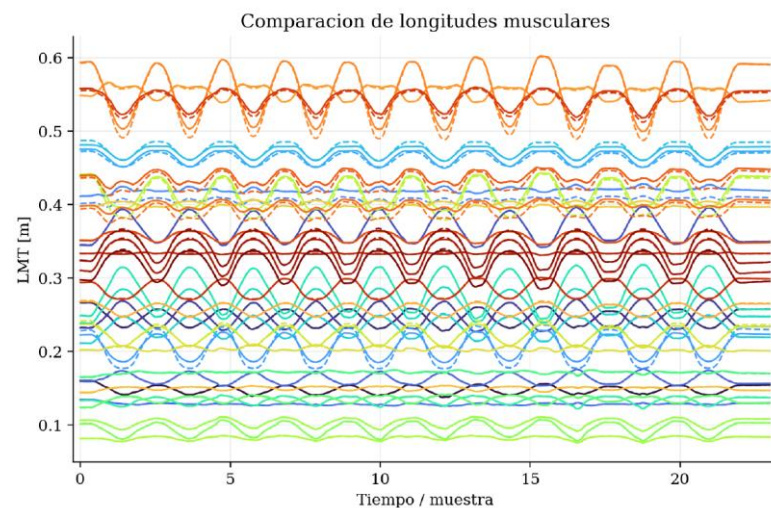
Placa de fuerza



Equipment

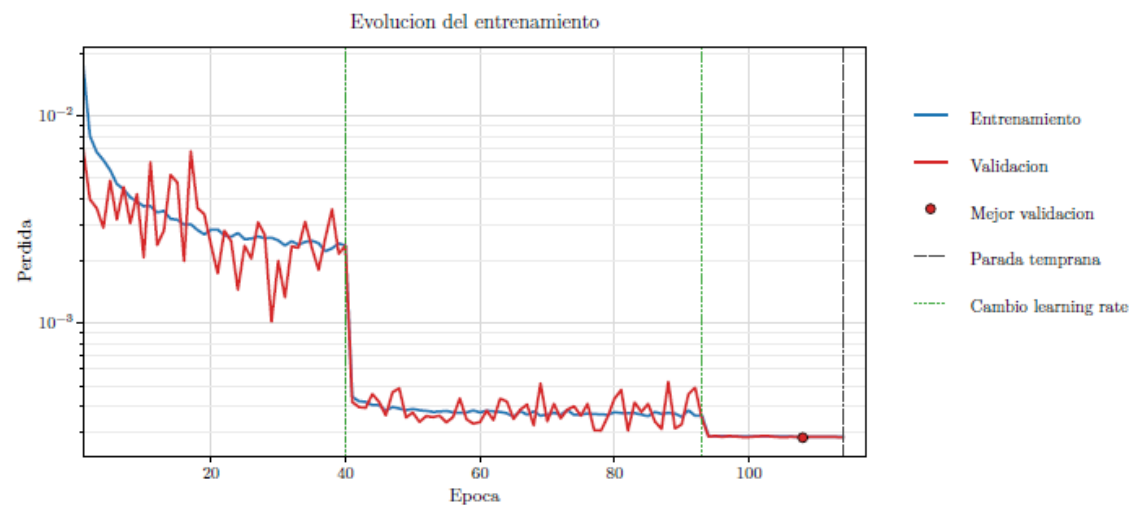
18 camaras infrarrojas (100 Hz)
37 marcadores ópticos
2 placas de fuerza (100 Hz)
1 célula de carga (100 Hz)
1 pesa

5. Análisis de resultados

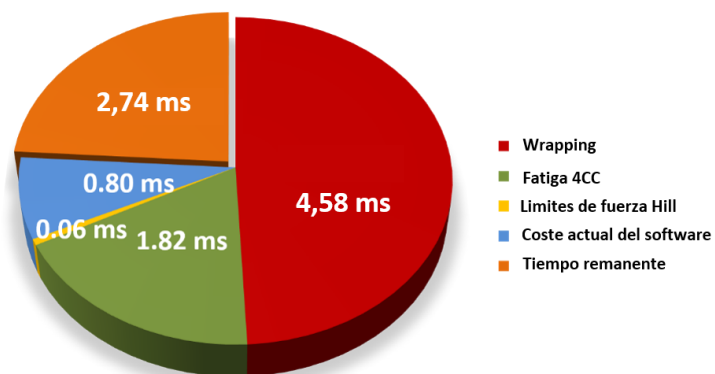
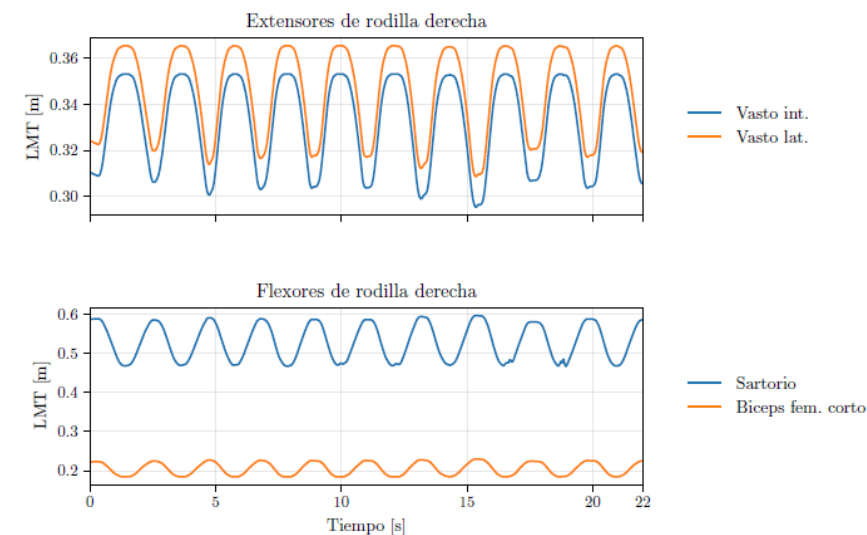
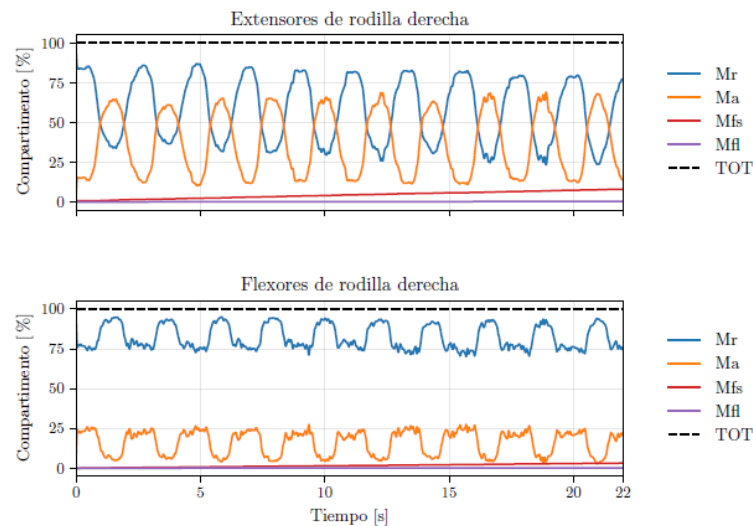
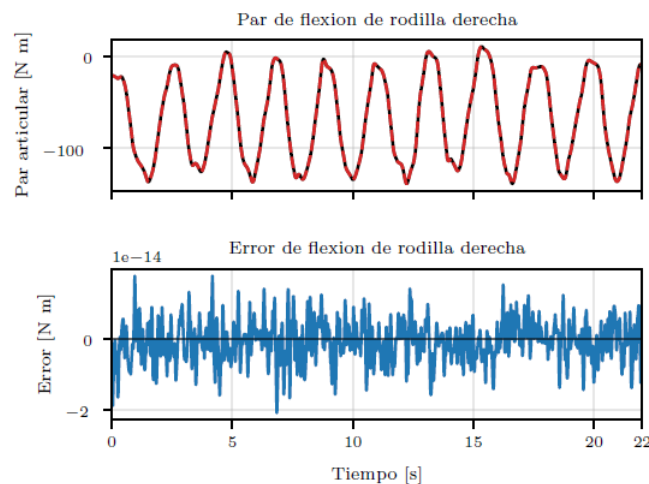


Metricas resumen de validacion para las salidas de la red neuronal.

Group	Vars.	Mean MAE (m)	Mean RMSE (m)	Mean Corr.
MTU length	36	1.83e-04	2.67e-04	0.999982
Moment arm	102	7.98e-05	1.44e-04	0.999799
Global	138	1.07e-04	1.76e-04	0.999847



6. Análisis de resultados



Component	Time (ms)	Percentage
Wrapping	4,58 ms	49.2%
Fatiga 4CC	1.82 ms	18.2%
Límites de fuerza de Hill	0.06 ms	0.6%
Coste actual del software	0.80 ms	8.0%
Tiempo restante	2,74 ms	24.0%
TOTAL	10.00 ms	100.0%

4.2. Conclusiones

- **La implementación de los diferentes modelos** a un único algoritmo se realizó con éxito mostrando una elevada eficiencia computacional
- Los tiempos de procesamiento obtenidos son compatibles con una futura ejecución en tiempo real
- **Las redes neuronales resultaron ser un método válido** para la resolución del problema del wrapping de forma eficiente
- El enfoque adoptado es **reproducible** y **escalable al resto del cuerpo humano** siempre que se disponga de un modelo anatómico con el que generar los datos de entrenamiento
- Trabajo aceptado para comunicación oral a **congreso internacional** (ECSS 2026 Lausanne)
- **Trabajo futuro:** validación del método y preparación de un artículo científico al respecto



CITENI



CAMPUS INDUSTRIAL



UNIVERSIDADE DA CORUÑA

Uso de un modelo simplificado de músculo para determinar la cinemática muscular en tiempo real

Autor:
Lucas Porzio González

Tutores:
Alberto Luaces Fernández
Florian Michaud

Grado en Ingeniería Mecánica

Julio 2026

Laboratorio de Ingeniería Mecánica,
CITENI, Campus Industrial de Ferrol,
Universidade da Coruña, Ferrol, España



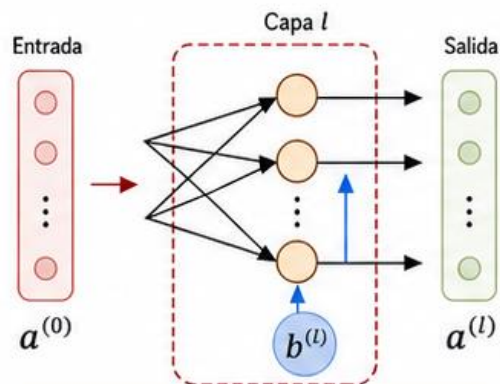
CITENI



UNIVERSIDADE DA CORUÑA

Entrenamiento de las redes

1 Propagación hacia delante



$$z^{(l)} = W^{(l)} a^{(l-1)} + b^{(l)}$$

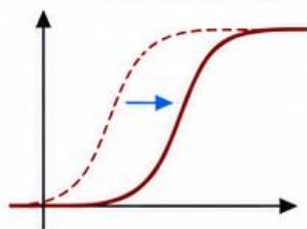
$$a^{(l)} = f(z^{(l)})$$

- $W^{(l)}$: pesos
- $b^{(l)}$: sesgos
- $f(\cdot)$: función de activación
- $a^{(l)}$: salida de la capa

2 Pesos y sesgos

W Los pesos controlan la importancia de cada entrada en la salida de la neurona.

b Los sesgos desplazan la respuesta de la neurona y permiten ajustar mejor la función aprendida.



3 Función de pérdida

Pérdida principal: MAE normalizado

$$\mathcal{L}_{MAE, norm} = \left\| \frac{y_i - \hat{y}_i}{R} \right\|_1$$



La normalización por rango evita que variables con mayor escala dominen la pérdida.

Pérdida total

$$\mathcal{L} = \mathcal{L}_{MAE, norm} + \lambda_1 \|W\|_1 + \lambda_2 \|W\|_2^2$$

4 Actualización de pesos

Descenso por gradiente

$$W^{(t+1)} = W^{(t)} - \eta \frac{\partial \mathcal{L}}{\partial W}$$



Se calcula el gradiente de la pérdida respecto a los pesos y se actualizan en la dirección que reduce el error.

Optimizador usado



Adam

Sigue esta idea general, pero adapta la actualización usando información acumulada del gradiente (momentos).

5 Control del entrenamiento

Tasa de aprendizaje

η controla el tamaño de los cambios en los pesos.

Si la pérdida de validación no mejora durante varias épocas:

Early stopping



El entrenamiento se detiene cuando la pérdida de validación no mejora durante varias épocas (paciencia).



Evita sobreajuste y ahorra cómputo innecesario.



Regularización



$L_1 (\|W\|_1)$

Penaliza la suma de valores absolutos de los pesos.

Favorece soluciones más dispersas (menos dependientes de conexiones concretas).

$L_2 (\|W\|_2^2)$

Penaliza el cuadrado de los pesos.



Evita valores excesivamente grandes y favorece una respuesta más suave y estable.



Resumen del flujo

Entrada → Capa neuronal → Predicción

Pérdida → Actualización → Control del entrenamiento

(proceso iterativo por épocas)