

Jabalina Sensorizada con IMU para Análisis del Lanzamiento y Biofeedback en Deportistas

Autor:
Santiago Pérez Delgado

Tutores:
Daniel Dopico Dopico
Florian Michaud

Máster en Informática Industrial y Robótica
Universidad de A Coruña
Septiembre 2025



1. Antecedentes

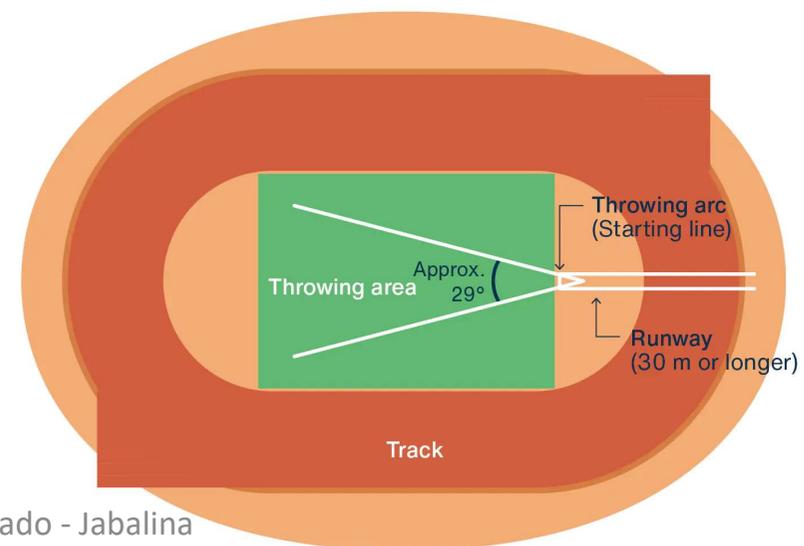


1.1 Lanzamiento de Jabalina

- Disciplina originada en la Antigua Grecia, actualmente olímpica.
- Consiste en lanzar la jabalina lo más lejos posible tras una carrera.
- Jabalinas de 600–800 g, compuestas por cabeza, asta y encordado

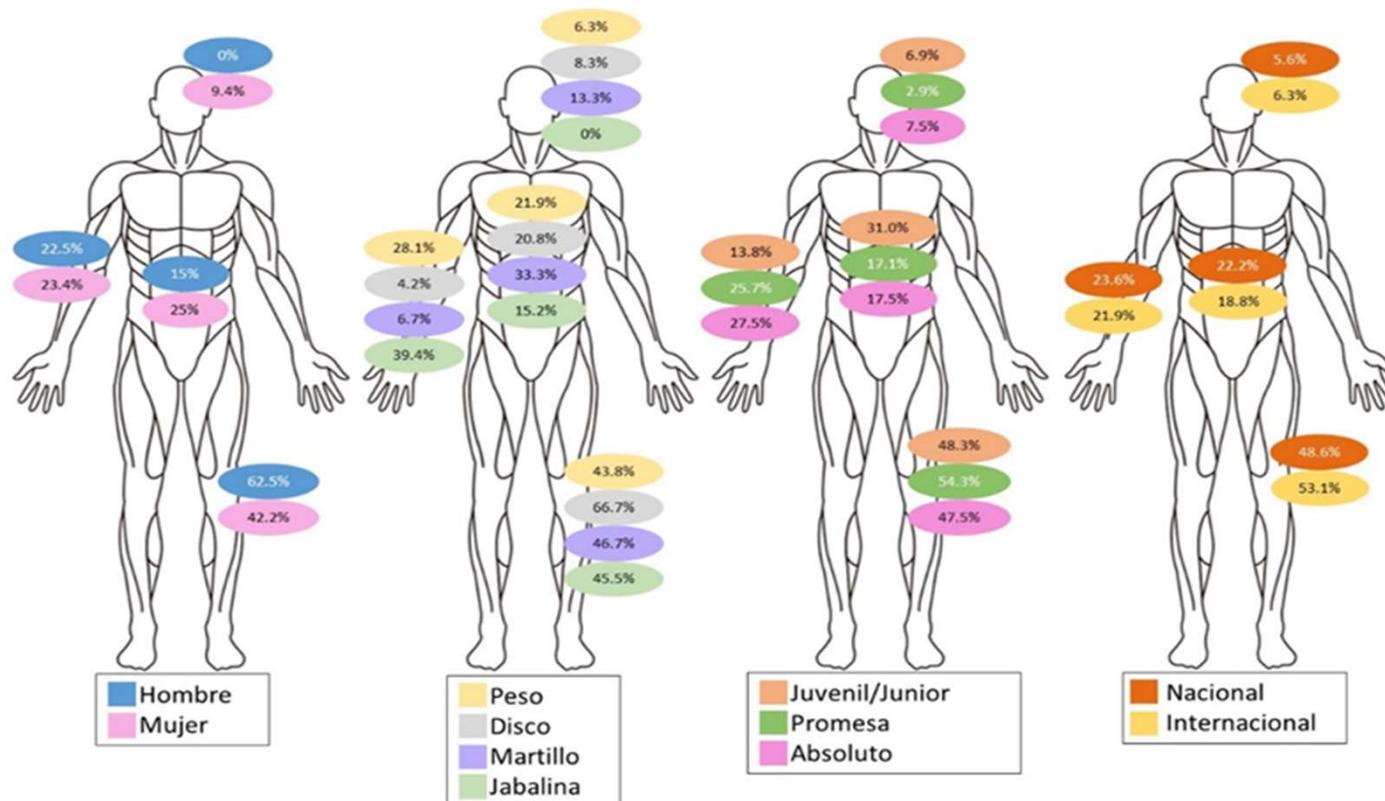


An athletic field and the javelin throw's throwing area



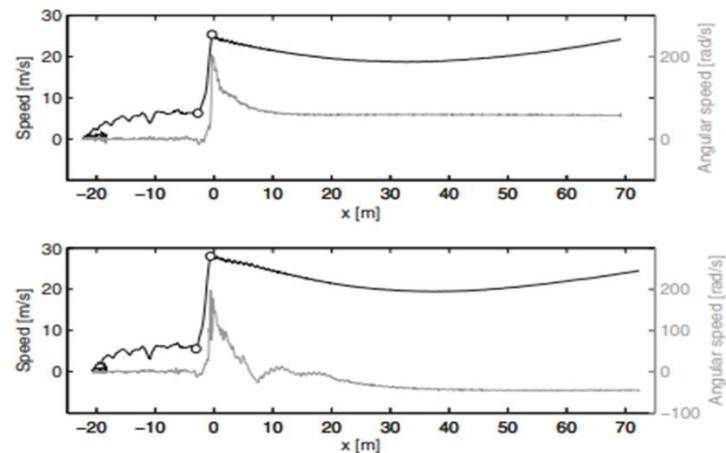
1.2 Lesiones en el lanzamiento de Jabalina

- Deporte de alta intensidad con elevado riesgo de lesión.
- Movimientos explosivos → sobrecargas y problemas en tendones.



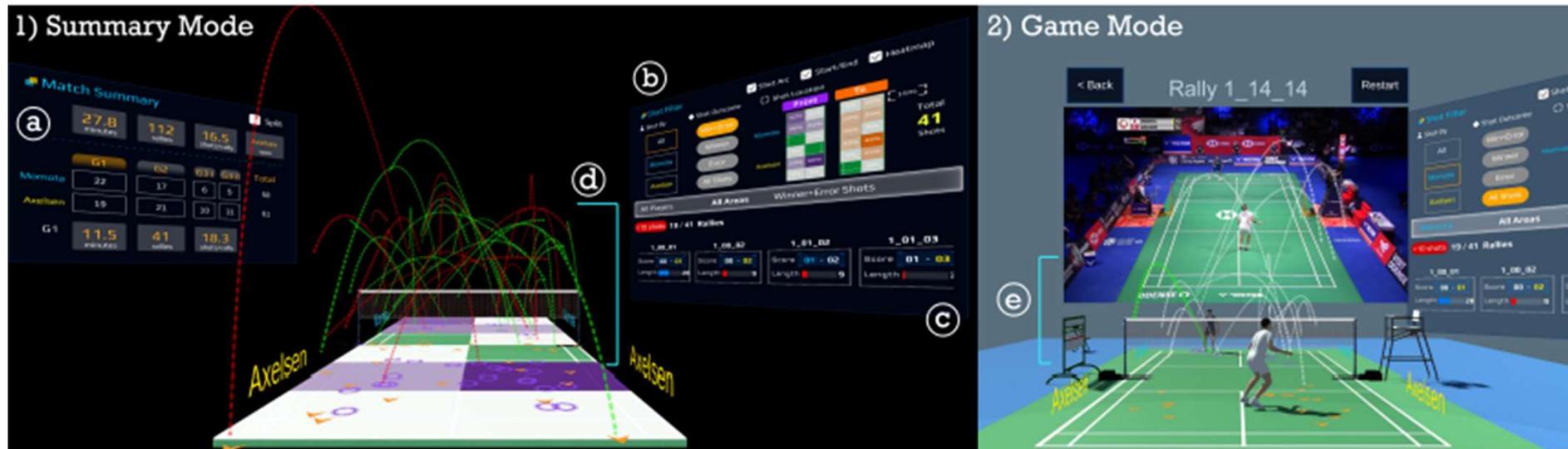
1.3 Adquisición de datos en el deporte

- Tecnologías más usadas:
- GPS (espacios abiertos).
- Sistemas ópticos (espacios cerrados).
- IMUs → alternativa económica y versátil.
- Aplicaciones actuales: análisis técnico y estudios de carga.



1.4 Simuladores en el deporte

- Principalmente basados en realidad virtual.
- Permiten reproducir situaciones reales y practicar en condiciones similares a la competición.



2. Requisitos de Diseño



2. Requisitos de Diseño

- Bajo coste.
- Fácil de usar.
- Adaptable y replicable.
- Ligero y de dimensiones reducidas.

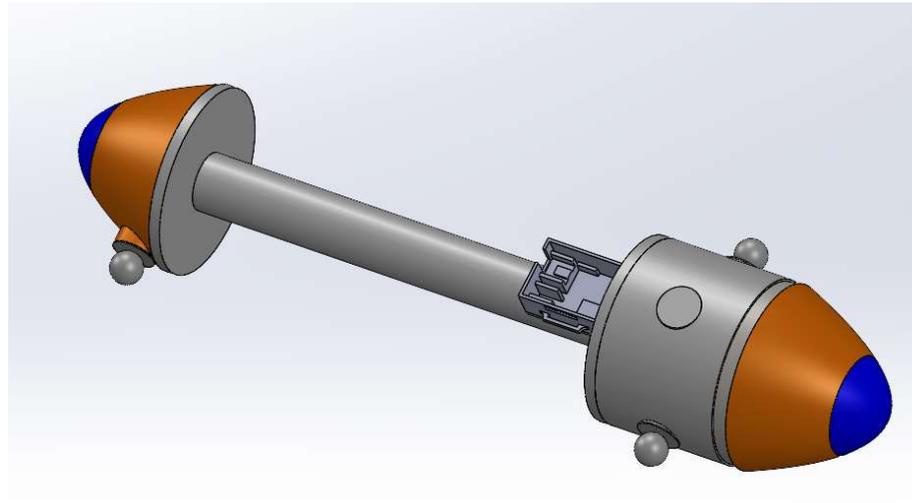


2.1 Metodología



2.1.1 Sensorización de la Jabalina con IMU

- Tamaño reducido: no altera las características de la jabalina.
- Componentes: acelerómetro, giroscopio y magnetómetro.
- Registra aceleraciones, orientaciones y velocidades angulares → derivan velocidades lineales.



2.1.2 Desarrollo de la simulación de lanzamiento

- Inicialmente implementado en **MATLAB** para validar el modelo.
- A partir de un vector de estados de entrada se calcula el alcance del lanzamiento

$$\mathbf{q} = \begin{bmatrix} r_{0x} & r_{0y} & r_{0z} & \gamma & \beta & \alpha & v_{0x} & v_{0y} & v_{0z} & \omega_{x_loc} \\ \omega_{y_loc} & \omega_{z_loc} & & & & & & & & \end{bmatrix}^T$$

Conversión y corrección de orientaciones

- Los cuaterniones se transforman en ángulos de Euler (α , β , γ) mediante ejes seguidores.

$$\mathbf{y} = \begin{bmatrix} r_{0x} & r_{0y} & r_{0z} & c_0 & c_1 & c_2 & c_3 & v_{0x} & v_{0y} & v_{0z} \\ \omega_{x_loc} & \omega_{y_loc} & \omega_{z_loc} & & & & & & & \end{bmatrix}^T$$

2.1.2 Desarrollo de la simulación de lanzamiento

- Se aplica la **regla trapezoidal** hasta que los resultados convergen.

$$y_{k+1}^* = y_k + \Delta t f(y_k, t_k).$$

- El criterio de error: la norma de la diferencia entre el resultado actual y los dos anteriores.

$$y_{k+1} = y_k + \frac{\Delta t}{2} [f(y_k, t_k) + f(y_{k+1}^*, t_{k+1})]$$

2.1.2 Desarrollo de la simulación de lanzamiento

Integración de estados

- La posición se obtiene con un **integrador paso a paso**.
- La predicción inicial se hace mediante un esquema **Forward-Euler**.

$$\mathbf{q} = \begin{bmatrix} r_{0x} & r_{0y} & r_{0z} & \gamma & \beta & \alpha & v_{0x} & v_{0y} & v_{0z} & \omega_{x,loc} \\ \omega_{y,loc} & \omega_{z,loc} & & & & & & & & \end{bmatrix}^T$$

2.1.2 Desarrollo de la simulación de lanzamiento

Se calcula la derivada del vector de estados en cada paso:

- La derivada de la posición proviene de las velocidades.
- La derivada de las orientaciones se resuelve con un sistema de ecuaciones diferenciales.

$$\begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} -\sin \beta & 0 & 1 \\ \sin \alpha \cos \beta & \cos \alpha & 0 \\ \cos \alpha \cos \beta & -\sin \alpha & 0 \end{bmatrix} \begin{bmatrix} \dot{\gamma} \\ \dot{\beta} \\ \dot{\alpha} \end{bmatrix}$$

Caso especial: si $\beta \approx 0$

si $|\cos(\alpha)| \geq \frac{\sqrt{2}}{2} \rightarrow \dot{\beta} = \frac{\dot{\omega}_y}{\cos \alpha}$

$\dot{\beta} = \frac{\dot{\omega}_z}{\sin \alpha}$

2.1.2 Desarrollo de la simulación de lanzamiento

Aceleraciones

- Aceleraciones lineales: deducidas de la **Segunda Ley de Newton**, considerando únicamente la gravedad.

$$m \mathbf{a} = \mathbf{F} \longrightarrow \begin{bmatrix} \dot{v}_{0x} \\ \dot{v}_{0y} \\ \dot{v}_{0z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}$$

- Aceleraciones angulares: obtenidas de la dinámica de la jabalina modelada como un sólido rígido sin torques externos.

$$\dot{\boldsymbol{\omega}} = -\mathbf{I}^{-1}(\boldsymbol{\omega} \times (\mathbf{I}\boldsymbol{\omega}))$$

2.1.2 Desarrollo de la simulación de lanzamiento

- En cada iteración se calcula la posición de los extremos de la jabalina.

$$p_{\text{extremo}} = p_{\text{cm}} \pm \mathbf{R} \cdot \mathbf{d}$$

- Si alguno de los puntos alcanza valores negativos en el eje vertical, se interpreta como contacto con el suelo → finalización de la simulación.

$$\mathbf{R} = R(z, \gamma)R(y, \beta)R(x, \alpha)$$

2.1.2 Desarrollo de la simulación de lanzamiento

- Código portado a **C++** con cambios mínimos de sintaxis.
- Uso de la librería **Eigen** para cálculos matriciales y vectoriales.
- Validación en **Qt Creator** mediante una pestaña específica de comprobación.

IMU MODO PRUEBAS

Orientación y velocidad de salida

Roll	<input type="text"/>	Xj	<input type="text"/>	Vx	<input type="text"/>
Pitch	<input type="text"/>	Yj	<input type="text"/>	Vy	<input type="text"/>
Yaw	<input type="text"/>	Zj	<input type="text"/>	Vz	<input type="text"/>

Calcular

Resultados simulación

X

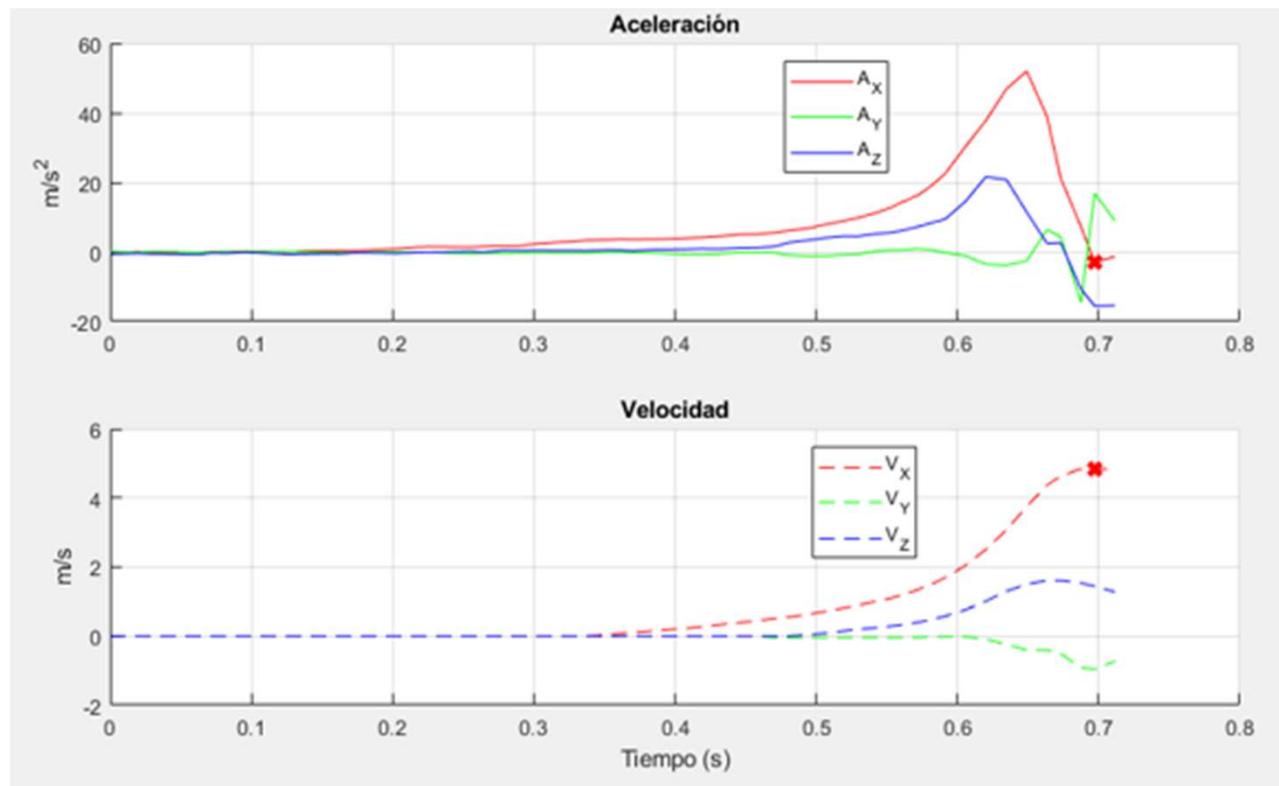
Y

Total

Validez del Lanzamiento

2.1.3 Detección de lanzamiento y visualización de resultados

- Estrategia automática: se detecta la suelta cuando la aceleración en eje X llega a cero.
- Se considera la altura del atleta y se valida si la jabalina cae con la punta.



2.1.4 Interfaz Gráfica

- Datos simplificados y útiles para el atleta.
- Botón de alineación del sistema de referencia con la pista.



2.1.4 Interfaz Gráfica

- Procedimiento de uso:
 1. Encender IMU y calibrar.
 2. Ejecutar software.
 3. Crear/cargar perfil de atleta.
 4. Alinear sistema de referencia.
 5. Realizar lanzamiento.



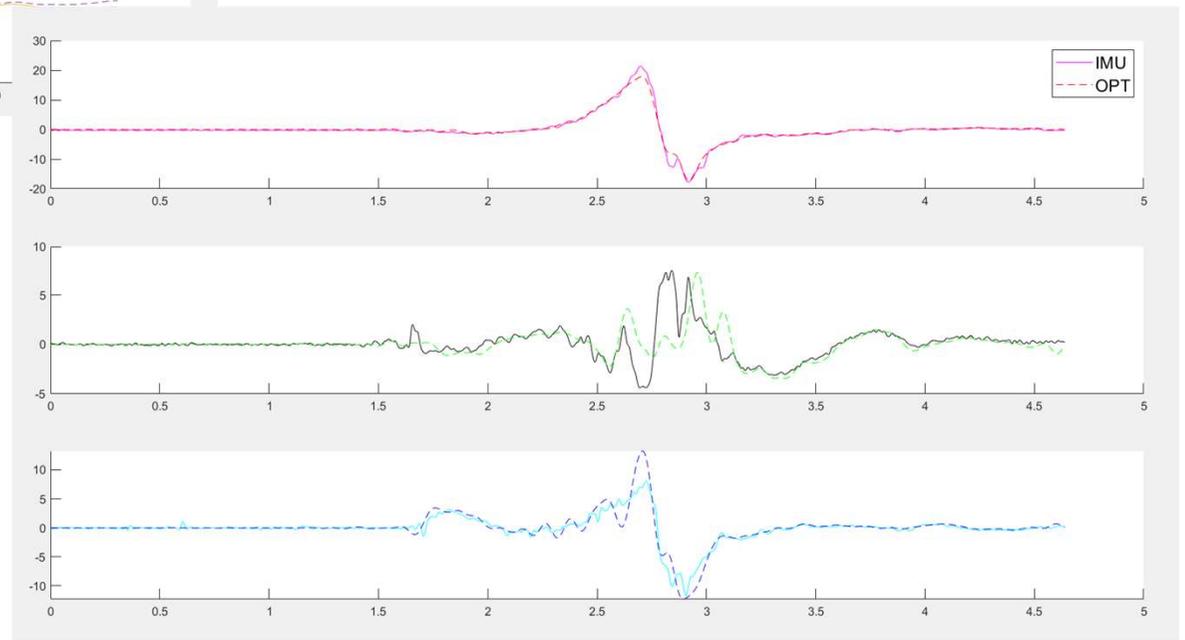
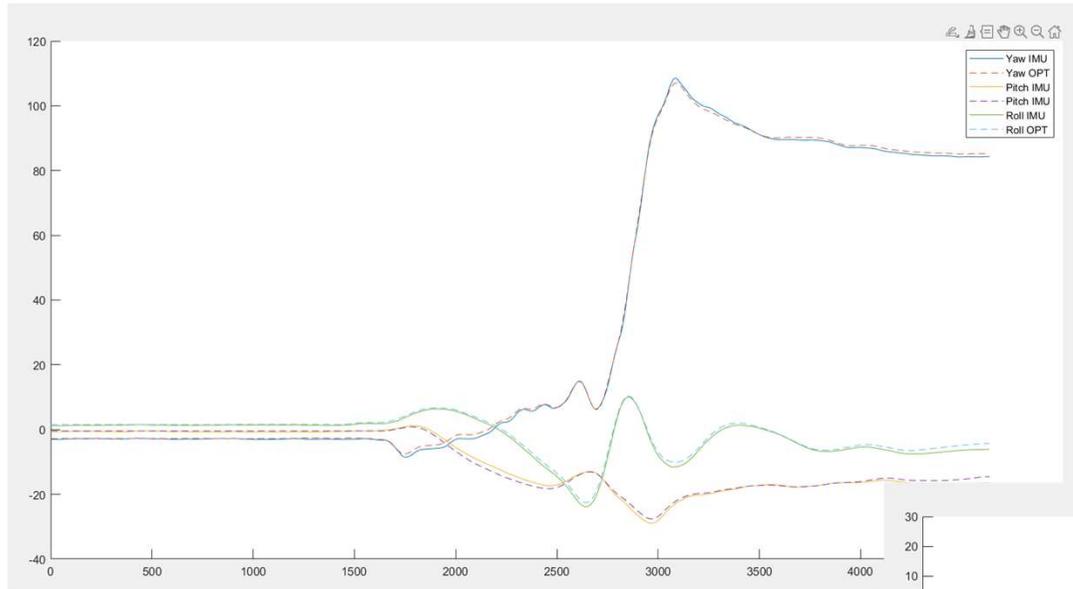
3. Análisis de las soluciones



3. Análisis de las soluciones

- Comparación con sistemas ópticos de captura de movimiento.
- Sincronización de frecuencias → comparación de aceleraciones, velocidad, posición y orientación.
- Datos IMU almacenados para validaciones externas.

3. Análisis de las soluciones

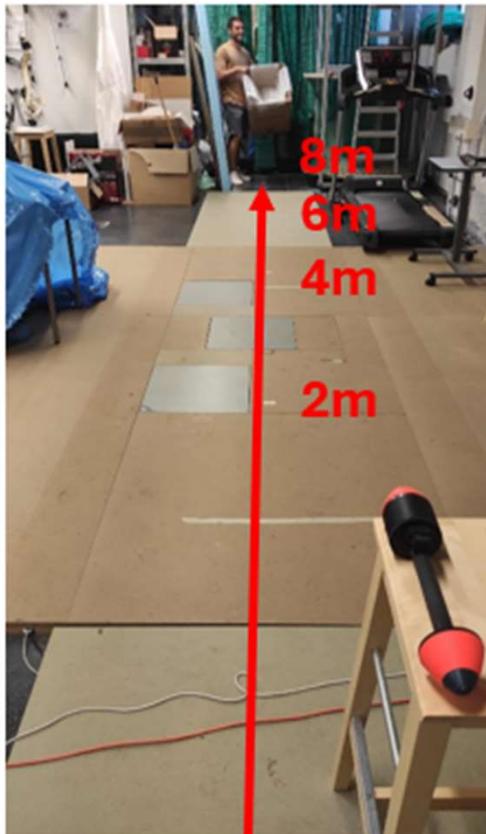


4. Resultados finales y Conclusiones



4.1 Resultados finales

- ✓ Validación experimental mediante lanzamientos a distancias conocidas.
- ✓ Detección precisa del instante de lanzamiento.



Experimental	Simulación
2.5m	2.0m
4.5m	3.4m
6.5m	5.5m
8.5m	7.4m

4.1 Resultados finales

- ✓ Validación experimental mediante lanzamientos a distancias conocidas.
- ✓ Detección precisa del instante de lanzamiento.



4.2 Conclusiones

- Simulador sencillo, rápido y efectivo.
- Interfaz gráfica clara y práctica para el atleta.
- Permite realizar múltiples lanzamientos consecutivos.
- Base sólida para continuar el desarrollo.
- Futuro: incorporar efectos aerodinámicos y mejorar la similitud de la pseudo-jabalina con la real.

Gracias por su atención

Jabalina Sensorizada con IMU
para Análisis del Lanzamiento y Biofeedback en Deportistas

Autor:
Santiago Pérez Delgado

Tutores:
Daniel Dopico Dopico
Florian Michaud

