

# **ESCUELA POLITÉCNICA NACIONAL**

**FACULTAD DE INGENIERÍA MECÁNICA**

**TÉCNICAS BASADAS EN SENSORES INERCIALES PARA EL  
SEGUIMIENTO DE LA ACTIVIDAD FÍSICA**

**VALIDACIÓN DE LAS TÉCNICAS BASADAS EN SENSORES  
INERCIALES PARA EL SEGUIMIENTO DE LA ACTIVIDAD  
FÍSICA**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO  
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERA EN  
MECÁNICA**

**NATHALY YESENIA CUMBAJIN DIAZ**

**nathaly.cumbajin@epn.edu.ec**

**DIRECTOR: WILLIAM RICARDO VENEGAS TORO**

**william.venegas@epn.edu.ec**

**Quito, agosto 2023**

## **CERTIFICACIONES**

Yo, NATHALY YESENIA CUMBAJIN DIAZ declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

---

**NATHALY YESENIA CUMBAJIN DIAZ**

Certifico que el presente trabajo de integración curricular fue desarrollado por NATHALY YESENIA CUMBAJIN DIAZ, bajo mi supervisión.

---

**WILLIAM VENEGAS  
DIRECTOR**

## **DECLARACIÓN DE AUTORÍA**

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

NATHALY CUMBAJIN DIAZ

WILLIAM VENEGAS TORO

## **DEDICATORIA**

A mis padres, por todo su apoyo y amor incondicional durante todo este periodo académico.  
Gracias por sus palabras alentadoras siempre que lo necesitaba.

A todos que con sus palabras me dieron su apoyo y que con risas compartidas me han dado  
fuerzas para afrontar cualquier momento difícil.

## **AGRADECIMIENTO**

Quiero expresar mi gratitud a toda mi familia por todo su apoyo constante y sabios consejos.

Agradezco sinceramente a mi tutor por su guía en la creación de este trabajo de integración curricular.

Por último, quiero agradecer a todos aquellos que contribuyeron de alguna forma a la realización de este trabajo de integración curricular.

# ÍNDICE DE CONTENIDO

CERTIFICACIONES .....	I
DECLARACIÓN DE AUTORÍA .....	II
DEDICATORIA .....	III
AGRADECIMIENTO .....	IV
ÍNDICE DE CONTENIDO .....	V
ÍNDICE DE FIGURAS.....	VIII
ÍNDICE DE TABLAS.....	X
RESUMEN.....	XI
ABSTRACT .....	XII
1 INTRODUCCIÓN.....	1
1.1 Objetivo general .....	2
1.2 Objetivos específicos.....	2
1.3 Alcance.....	2
1.4 Marco teórico.....	2
1.4.1 Actividad física .....	2
1.4.2 Métodos para cuantificar el movimiento humano .....	2
1.4.3 Técnicas de movimiento .....	3
1.4.4 Análisis del movimiento clínico e instrumental .....	3
1.4.5 Anatomía funcional de la extremidad superior .....	4
1.4.5.1 Hombro .....	4
1.4.5.1 Articulación del codo .....	5
1.4.6 Método de Euler.....	6
1.4.7 Método de cuaterniones.....	8
1.4.8 Sistemas con sensores inerciales .....	9
1.4.8.1 Usos prácticos de sensores inerciales en el área de salud.....	10
1.4.8.2 IMUs.....	10
1.4.8.3 EMG .....	11
1.4.8.4 Fotogrametría.....	11

1.4.9 OpenSim.....	12
2 METODOLOGÍA.....	14
2.1 Desarrollo de programa para el tratamiento de datos en la calibración .....	15
2.2 Calibración.....	19
2.2.1 Materiales.....	19
2.2.1.1 Marcadores .....	19
2.2.1.2 Cinta adhesiva.....	20
2.2.1.2 Sensores inerciales IMUS.....	20
2.2.1.3 Método de fotogrametría .....	20
2.2.2 Instrumentación.....	20
2.2.3 Captura de datos por fotogrametría e IMUs.....	21
2.2.4 Tratamiento de datos en Programa Matlab.....	21
2.2.4.1 Calibración del eje X.....	21
2.2.4.2 Calibración del eje Y .....	22
2.2.4.3 Calibración del eje Z.....	22
2.2.5 Simulación en OpenSim .....	22
2.3 Validación .....	22
2.3.1 Desarrollo de protocolo de pruebas.....	24
2.3.1.1 Muestra del experimento .....	24
2.3.1.2 Materiales.....	25
2.3.1.3 Instrumentación .....	25
2.3.1.3.1 Ubicación de sensores y marcadores en el sujeto .....	25
2.3.1.4 Posición y orientación de sujeto.....	26
2.3.2 Tratamiento de datos y procedimiento.....	27
2.3.2.1 Calibración del experimento registros IMUs.....	28
2.3.2.2 Simulación en OpenSim IMUs .....	28
2.3.2.3 Calibración del experimento registro de fotogrametría.....	31
2.3.2.4 Simulación en OpenSim fotogrametría .....	31
2.3.2.5 Validación de fotogrametría e IMUs.....	36
3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES.....	39
3.1 Análisis de resultados.....	39
3.1.1 Resultados de la calibración .....	39

3.1.1.1 Calibración en X .....	39
3.1.1.2 Calibración Y .....	41
3.1.1.3 Calibración Z .....	42
3.1.1 Resultados de la experimentación .....	44
3.1.2 Resultados de la validación .....	45
3.2 Conclusiones .....	48
3.3 Recomendaciones .....	49
4 REFERENCIAS BIBLIOGRÁFICAS .....	50
5 ANEXOS .....	52
ANEXO I .....	53
ANEXO II .....	54
ANEXO III .....	57
ANEXO IV .....	61



## ÍNDICE DE FIGURAS

<b>Figura 1.</b>	Modelo cinemático del brazo (Lobato, 2016).....	4
<b>Figura 2.</b>	Representación de extremidad superior en 3D .....	4
<b>Figura 3.</b>	Articulaciones de la cintura escapular: 1) glenohumeral, 2) suprahumeral, 3) acromioclavicular, 4) escapulocostal, 5) esternoclavicular, 6) esternocostal, 7) costovertebral 5	
<b>Figura 4.</b>	Articulación del codo.....	6
<b>Figura 5.</b>	Movimientos realizados por la extremidad superior.....	6
<b>Figura 6.</b>	Representación vectorial de la rotación de un punto P .....	7
<b>Figura 7.</b>	Diagrama del tratamiento de datos de sensores inerciales (Cuadros, 2020).....	11
<b>Figura 8.</b>	Esquema de registro de datos por fotogrametría (Instituto de biomecánica de Valencia, s/f).....	12
<b>Figura 9.</b>	Modelo musculoesquelético en plataforma OpenSim.....	13
<b>Figura 10.</b>	Esquema de metodología. ....	15
<b>Figura 11.</b>	Ejes de rotación de IMUs (Cuadros, 2020).....	19
<b>Figura 12.</b>	Instrumentación para la calibración (Fuente: propia).....	20
<b>Figura 13.</b>	Mecanismo rotativo, IMU y marcadores. (Fuente: propia) .....	21
<b>Figura 14.</b>	Ejes de la articulación del codo según la ISB. ( <i>Descripción de los modelos implementados</i> , 2020) .....	23
<b>Figura 15.</b>	Ejes de articulación de codo reemplazando IC. ( <i>Descripción de los modelos implementados</i> , 2020) .....	24
<b>Figura 16.</b>	Esquema de programa en Matlab de calibración en IMUs .....	28
<b>Figura 17.</b>	Archivo Traker_imu_cuat.sto.....	30
<b>Figura 18.</b>	Modelo de IMUs en OpenSim .....	30
<b>Figura 19.</b>	Esquema de programa en Matlab de calibración en fotogrametría.....	31
<b>Figura 20.</b>	Ventana de OpenSim para escalar los registros de la experimentación .....	35
<b>Figura 21.</b>	Modelo de fotogrametría con OpenSim.....	35
<b>Figura 22.</b>	Modelo escalado en OpenSim .....	36
<b>Figura 23.</b>	Posición angular Pitch calibración en X.....	39
<b>Figura 24.</b>	Posición angular Roll calibración en X. ....	40
<b>Figura 25.</b>	Posición angular Yaw calibración en X.....	40

<b>Figura 26.</b>	Posición angular Pitch calibración en Y.....	41
<b>Figura 27.</b>	Posición angular Roll calibración en Y. ....	41
<b>Figura 28.</b>	Posición angular Yaw calibración en Y.....	42
<b>Figura 29.</b>	Posición angular Pitch calibración en Z.....	42
<b>Figura 30.</b>	Posición angular Roll calibración en Z.....	43
<b>Figura 31.</b>	Posición angular Yaw calibración en Z.....	43
<b>Figura 32.</b>	Cinemática del brazo según fotogrametría y OpenSim (rojo=fotogrametría, azul=OpenSim).....	44
<b>Figura 33.</b>	Posición angular componente principal brazo. ....	46
<b>Figura 34.</b>	Posición angular en el componente principal en el antebrazo.....	47

## ÍNDICE DE TABLAS

<b>Tabla 1.</b>	Sensores inerciales y características .....	9
<b>Tabla 2.</b>	Ejemplos de usos prácticos de sensores inerciales. ....	10
<b>Tabla 3.</b>	Relación de variables de IMUs y fotogrametría calibración eje X .....	21
<b>Tabla 4.</b>	Relación de variables de IMUs y fotogrametría calibración eje Y .....	22
<b>Tabla 5.</b>	Relación de variables de IMUs y fotogrametría calibración eje Z .....	22
<b>Tabla 6.</b>	Ubicación de los sensores y marcadores.....	25
<b>Tabla 7.</b>	Posición y orientación del sujeto .....	27
<b>Tabla 8.</b>	Ángulos de Euler y ejes de rotación.....	44

## RESUMEN

En la actualidad existen varios métodos para capturar el movimiento de actividades físicas, que son costosos y de complicada movilidad. Por este motivo nace la necesidad de buscar alternativas que ofrezcan los mismos resultados, pero que no sean costosos e imprácticos. Es así como con este trabajo de integración curricular se pretende determinar la fiabilidad del uso de sensores inerciales IMUs, para el seguimiento de la actividad física. Ya que los beneficios que brindan con respecto a otros métodos son altos.

Para desarrollar la metodología se realiza la experimentación de un modelo biomecánico del codo utilizando el sistema de fotogrametría y sensores IMUs. En donde se detalla la importancia de la calibración, el tratamiento de datos con la creación de programas en Matlab para posteriormente validar ambas técnicas. Adicionalmente se compara la cinemática obtenida con los registros de fotogrametría y OpenSim, mediante una simulación. Además, se creó programas en Matlab detallando la transformación de los resultados de la experimentación de Euler a cuaterniones ya que es el formato aceptable en OpenSim.

Al realizar la validación de las técnicas se obtienen resultados de las posiciones angulares en el plano principal del brazo con un CMC (Comparative Measurement Close-range) de 0.91 y del antebrazo con un CMC de 0.96 y cuyas curvas indican la misma tendencia. Lo que confirma la fiabilidad de utilizar los sensores IMUs como técnica de captura de movimiento en donde se analicen modelos simples.

**PALABRAS CLAVE:** IMUs, fotogrametría, OpenSim, Matlab, modelo biomecánico del codo.

## ABSTRACT

Currently there are several methods to capture the movement of physical activities, which are expensive and difficult to move. For this reason, the need arises to look for alternatives that offer the same results, but that are not expensive and impractical. This is how this curricular integration project aims to determine the reliability of the use of IMUs inertial sensors for monitoring physical activity. Since the benefits they provide compared to other methods are high.

To develop the methodology, experimentation is carried out on a biomechanical model of the elbow using the photogrammetry system and IMUs sensors. Where the importance of calibration is detailed, the data treatment with the creation of scripts in Matlab to later validate both techniques. Additionally, the kinematics obtained with the photogrammetry and OpenSim records are compared, through a simulation. In addition, scripts were created in Matlab detailing the transformation of the results of Euler's experimentation to quaternions since it is the acceptable format in OpenSim.

When performing the validation of the techniques, results of the angular positions in the main plane of the arm are obtained with a CMC (Comparative Measurement Close-range) of 0.91 and of the forearm with a CMC of 0.96 and whose curves indicate the same tendency. This confirms the reliability of using IMUs sensors as a motion capture technique where simple models are analyzed.

**KEYWORDS:** IMUs, photogrammetry, Matlab, OpenSim, biomechanical model of the elbow.

# 1 INTRODUCCIÓN

Actualmente existen varios métodos para registrar el movimiento de actividades físicas que son utilizados en áreas de la salud, deporte, entretenimiento e industria, que son relativamente costosos y de complicada movilidad. Es por esta razón que nace la necesidad de buscar alternativas viables que ofrezcan los mismos resultados pero que no sean costosos e imprácticos. Por varios años se han utilizado el método MOCAP (captura de movimiento) para el diagnóstico de condiciones físicas en medicina, para la implementación de técnicas de rehabilitación que no comprometan el movimiento. De la misma forma en el área deportiva se ha utilizado fotogrametría para la optimización de movimientos de deportistas y la recuperación de lesiones. Sin embargo, en ambos ejemplos se requería de un gran espacio para la realización de estudios que contenga cámaras de video calibradas y fijas para registrar los movimientos.

El método para cuantificar la actividad física es conocido como MOCAP, este método consiste en la utilización de varias cámaras alrededor del sujeto de estudio. A pesar de su utilidad este método es costoso y con restringida movilidad, lo que lo hace impráctico para su uso directo en distintas áreas de estudio. Por estos motivos se pretende usar sensores inerciales IMUs por ser una alternativa prometedora, ya que al contar con adaptaciones de acelerómetros y giroscopios ofrecen resultados de velocidad angular y aceleración lineal adicional, que contribuyen a la creación de modelos matemáticos para la resolución de problemas y finalmente obtener una solución confiable dependiendo de los requerimientos de cada área.

Para validar la eficacia del uso de los sensores IMUs al reemplazar otros sistemas es necesario la creación de un modelo biomecánico que permita la obtención de registros tanto en IMUs como en Fotogrametría para comparar los resultados. Además, se analiza la cinemática de los sensores IMUs al usar la metodología de Euler con ayuda de un programa desarrollado en Matlab.

Se ha mencionado que el uso de los sensores IMUs proporciona varias ventajas al momento de comparar con otros métodos para capturar el movimiento. Los sensores IMUs son de fácil portabilidad y tienen precios asequibles, por lo que sería de gran utilidad en diferentes campos de aplicación, es así como con el presente trabajo de integración curricular se pretende determinar la fiabilidad y comparabilidad de los sensores IMUs con un sistema de fotogrametría

## 1.1 Objetivo general

Determinar la fiabilidad de las técnicas basadas en sensores inerciales IMUs para el seguimiento de la actividad física.

## **1.2 Objetivos específicos**

1. Realizar investigaciones sobre las técnicas de captura de movimiento en diferentes sectores como de la salud, ergonomía y deporte, para definir las ventajas y desventajas de cada técnica.
2. Establecer metodología de Euler para el análisis de la cinemática de los IMUs.
3. Desarrollar un modelo biomecánico que permita determinar la viabilidad de usar sensores IMUs para la captura de movimiento.
4. Desarrollar un programa en el software Matlab para el tratamiento de datos y comparación de los registros de los sensores IMUs con fotogrametría.

## **1.3 Alcance**

El presente trabajo de integración curricular se centrará en la validación de las técnicas basadas en sensores inerciales IMUs para el seguimiento de la actividad física mediante un modelo biomecánico del codo, además se enfocará en el desarrollo de algoritmos en Matlab para contrastar los registros de los sensores IMUs y fotogrametría. No se considerarán otras técnicas de captura de movimiento.

## **1.4 Marco teórico**

### **1.4.1 Actividad física**

La actividad física es definida como todo movimiento que es producido por el cuerpo por medio de los músculos esqueléticos al consumir energía. Estos movimientos pueden ser desde el descanso, desplazamiento, ejercicio, actividades de recreación, entre otros (Ainsworth, s/f).

La actividad física moderada e intensa de manera regular permite al individuo mantener su estado de salud. Sin embargo, según la organización mundial de la salud los niveles de actividad física realizada por la población adulta no son suficientes para más de una cuarta parte, lo que conllevaría consecuencias negativas afectando a los sistemas de salud de los países (Organización Mundial de la Salud, 2022).

### **1.4.2 Métodos para cuantificar el movimiento humano**

El método de laboratorio que permite rastrear el movimiento humano es el MOCAP cuya

técnica se basa en capturar el movimiento mediante video. En el MOCAP se utilizan cámaras de alta velocidad que se ubican alrededor del objetivo, las cuales producen coordenadas de posición tridimensionales con la correcta calibración. Para la obtención de las coordenadas es necesario usar marcadores ya sean reflectantes o LED infrarrojos.

Existen los métodos de radiografía estereoscópica y las resonancias magnéticas en donde no es necesario el uso de marcadores y se obtienen descripciones de gran precisión del movimiento óseo, sin embargo, los sujetos estudiados se someten a altos niveles de radiación limitando el uso de estos métodos (*MOCAP, como funciona la captura de movimiento y que aplicaciones tiene para la industria, s/f*).

Otra alternativa es el método de captura de movimiento en video sin uso de marcadores, en donde es necesario el uso de cámaras de video de alta velocidad para el registro del modelo en forma de imágenes o fotogramas, aunque en este método existen limitaciones de precisión y medición.

Las diferentes desventajas que ofrece el MOCAP como el costo, precisión, portabilidad, medición, entre otros, imposibilitan aplicar este método como uno clínicamente viable para capturar el movimiento. Por lo cual el desarrollo de tecnologías alternativas a permitido el uso de los IMUs que están compuestos de acelerómetros y giroscopios, facilitando la obtención de datos de velocidad angular y aceleración lineal. Además, por su portabilidad se lo utiliza en varios sectores en donde se analice la actividad física (Mcginnis, 2013).

### **1.4.3 Técnicas de movimiento**

Las técnicas de movimiento se encuentran clasificadas en dos tipos el método creativo y el método imitativo, el preferido es el método imitativo por el bajo desgaste energético al tratarse de series repetitivas de movimiento. Se conoce que el movimiento corporal se interrelaciona con parámetros dinámicos, rítmicos y espaciales para lograr coordinar los movimientos (Gomeñuka & Cabral, 2008). Existen algunas actividades que requieren de secuencias ordenadas para lograr el movimiento adecuado como lo son los movimientos de flexo-extensión, de balanceos, impulsos, movimientos de ondas, entre otros.

### **1.4.4 Análisis del movimiento clínico e instrumental**

Para la mejor decisión a la hora de un diagnóstico clínico, los médicos utilizan el análisis instrumental que facilita determinar la terapia adecuada, además de optimizar el conocimiento sobre un sistema biológico específico (Chaler et al., 2005). El análisis instrumental le facilita al profesional de la salud estudiar el sistema biológico a partir de puntos fisiológicos y anatómicos.

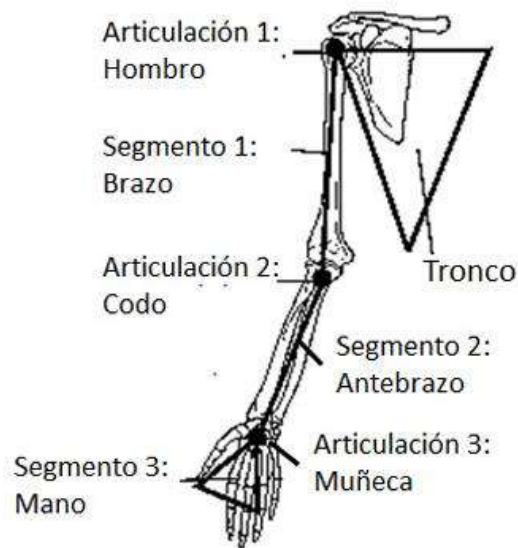
En el área de rehabilitación el análisis instrumental se basa en diversos parámetros como;



el tipo de instrumentación, el modelamiento matemático, algoritmos y el procesamiento de datos. Además, los resultados del análisis instrumental no deben ser diferentes a los resultados del análisis clínico (Garofalo, s/f).

### 1.4.5 Anatomía funcional de la extremidad superior

Los elementos principales de la extremidad superior son: clavícula, húmero, escápula, antebrazo y brazo. A continuación, se indica un modelo cinemático del brazo y articulaciones importantes en la figura 1.



**Figura 1.** Modelo cinemático del brazo (Lobato, 2016)

En la figura 2 se muestra los elementos principales de la extremidad superior.

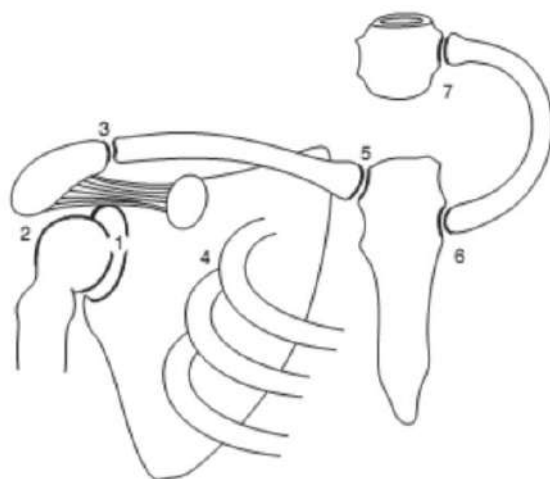


**Figura 2.** Representación de extremidad superior en 3D

#### 1.4.5.1 Hombro

Es importante analizar la cintura escapular que se encuentra formada por siete articulaciones como lo son la unión suprahumeral, la articulación acromioclavicular, la articulación acromioclavicular, escapulocostal, esternoclavicular, esternocostal y costovertebral (Laguna, 2023).

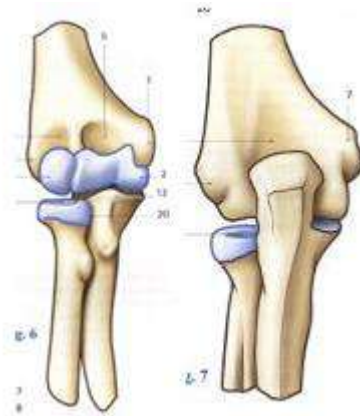
El movimiento de giro de la clavícula alrededor del manubrio esternón forma la articulación esternoclavicular. La estructura principal que se encarga de sostener el brazo a la pared torácica es la escápula, y esta se conecta con la articulación glenohumeral. Para el soporte de la articulación esternoclavicular actúa la clavícula (Navarro, 2023). Tanto la escápula y la clavícula se encuentran conectadas por la articulación acromioclavicular. Además, es fundamental mencionar que gracias a la presencia de ligamentos como el trapecio claviculoescapular y conoide el movimiento de la escápula es posible. A continuación, se indica la esquematización de la cintura escapular.



**Figura 3.** Articulaciones de la cintura escapular: 1) glenohumeral, 2) suprahumeral, 3) acromioclavicular, 4) escapulocostal, 5) esternoclavicular, 6) esternocostal, 7) costovertebral

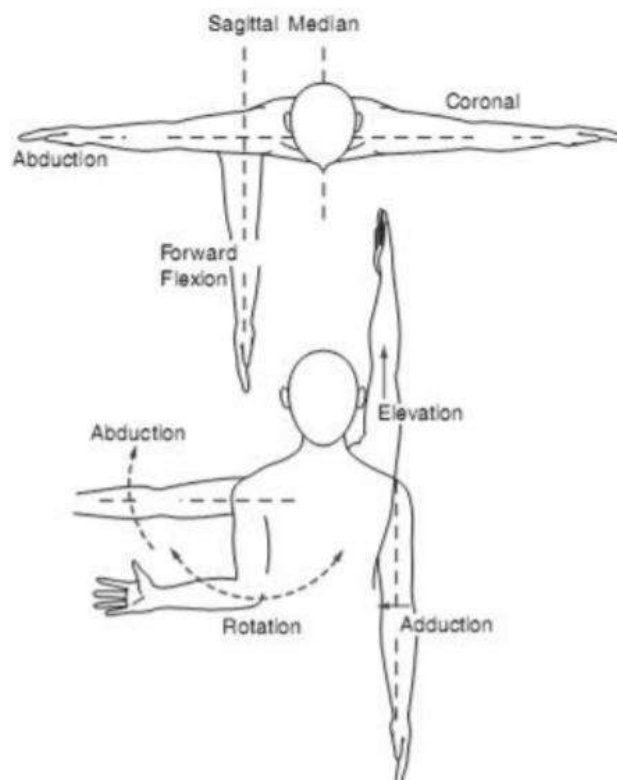
#### 1.4.5.1 Articulación del codo

La articulación del codo se encuentra localizada en el centro de la extremidad superior. La cual está formada por la unión del húmero, cúbito y radio. Gracias a esta articulación y su amplitud de movimiento de 150° a 160° son posibles los movimientos de flexión como llevar la mano hacia la cara. Además, permite realizar el movimiento de extensión con una pequeña amplitud de 5° a 10° de recurvatum hacia atrás. La siguiente figura 4 indica una representación de la articulación del codo (González, 2016).



**Figura 4.** Articulación del codo

La extremidad superior realiza los siguientes movimientos, en el plano sagital realiza flexión y extensión hacia delante y atrás respectivamente, en el plano frontal movimientos de abducción y aducción y en el plano transversal el movimiento de rotación tanto interna como externa (Garofalo, s/f). En la siguiente figura 5 se indica los movimientos realizados por la extremidad superior.

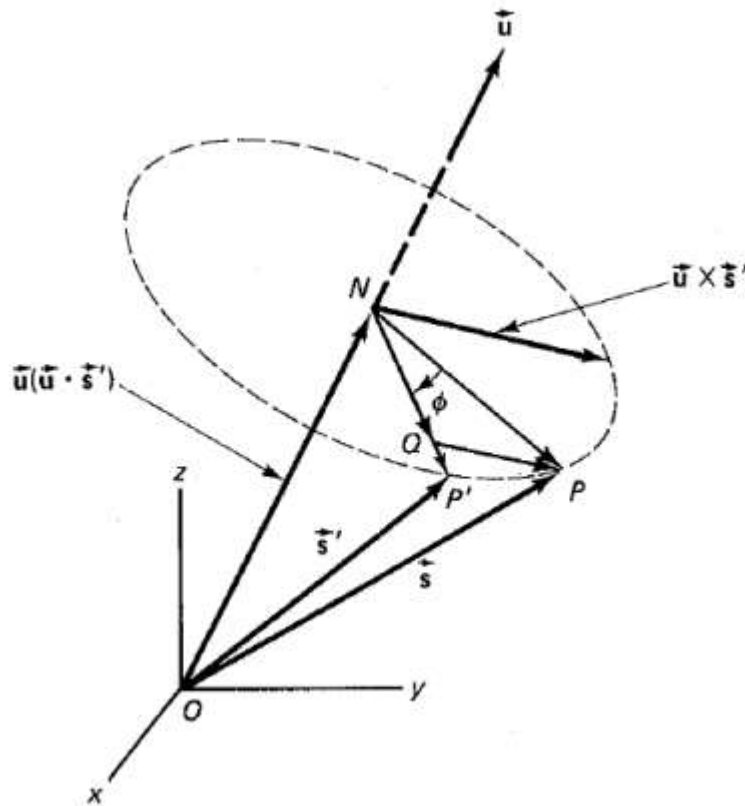


**Figura 5.** Movimientos realizados por la extremidad superior

#### 1.4.6 Método de Euler

Uno de los métodos utilizados para calcular la cinemática es el método de Euler. En donde la orientación de un objeto se encuentra definido por una matriz de transformación la cual

es recomendable que sea una función continua en el tiempo. A continuación, se indica la representación vectorial del cambio de dirección. de un punto P en un instante de tiempo (Ramírez, 2011).



**Figura 6.** Representación vectorial de la rotación de un punto P

En la figura 6 se representa el cambio de dirección de un punto P, el cual se calcula mediante la matriz de rotación. Los términos que forman esta matriz facilitan la determinación de los parámetros de Euler que posteriormente ayudaran a encontrar la cinemática de un sujeto con mayor facilidad.

$$\vec{s}' = A\vec{s} \tag{1}$$

Los elementos de la matriz de rotación son cosenos directores del eje rotacional.

$$A = [\vec{u}_\eta \quad \vec{u}_\xi \quad \vec{u}_\zeta] \tag{2}$$

$$\vec{u}_\eta = \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \end{bmatrix}, \vec{u}_\xi = \begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \end{bmatrix}, \vec{u}_\zeta = \begin{bmatrix} a_{13} \\ a_{23} \\ a_{33} \end{bmatrix} \tag{3}$$

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{4}$$

Los parámetros de Euler son cuatro y forman un vector que se muestra a continuación.

$$\vec{p} = \begin{bmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{bmatrix} \quad (5)$$

La matriz A con estos elementos y con un cálculo vectorial se puede reescribir de esta forma.

$$A = 2 \begin{bmatrix} e_0^2 + e_1^2 - \frac{1}{2} & e_1 e_2 - e_0 e_3 & e_1 e_3 + e_0 e_2 \\ e_1 e_2 + e_0 e_3 & e_0^2 + e_2^2 - \frac{1}{2} & e_2 e_3 - e_0 e_1 \\ e_1 e_3 - e_0 e_1 & e_2 e_3 + e_0 e_1 & e_0^2 + e_3^2 - \frac{1}{2} \end{bmatrix} \quad (6)$$

Con la determinación de la traza en la ecuación (7) y la matriz A (4) se establece el primer parámetro de Euler.

$$\text{tr } A = a_{11} + a_{22} + a_{33} \quad (7)$$

$$e_0^2 = \frac{\text{tr } A + 1}{4} \quad (8)$$

Al sustituir (8) en la diagonal de la matriz de A, se tiene los demás parámetros de Euler.

$$e_1^2 = \frac{1 + 2a_{11} - \text{tr } A}{4} \quad (9)$$

$$e_2^2 = \frac{1 + 2a_{22} - \text{tr } A}{4} \quad (10)$$

$$e_3^2 = \frac{1 + 2a_{33} - \text{tr } A}{4} \quad (11)$$

Se conoce que el vector (5) es continuo en el tiempo y el mismo pertenece a la matriz A que es una función continua. A partir de los parámetros se puede determinar las matrices G y L que ayudan a resolver la velocidad y aceleración angular en el sistema de coordenadas global y el sistema de coordenadas propio.

$$G = \begin{bmatrix} -e_1 & e_0 & -e_3 & e_2 \\ -e_2 & e_3 & e_0 & -e_1 \\ -e_3 & -e_2 & e_1 & e_0 \end{bmatrix} \quad (12)$$

$$L = \begin{bmatrix} -e_1 & e_0 & e_3 & -e_2 \\ -e_2 & -e_3 & e_0 & e_1 \\ -e_3 & e_2 & -e_1 & e_0 \end{bmatrix} \quad (13)$$

$$\omega_g = 2G\dot{p} \quad (14)$$

$$\omega_p = 2L\dot{p} \quad (15)$$

$$\dot{\omega}_g = 2G\ddot{p} \quad (16)$$

$$\dot{\omega}_p = 2L\ddot{p} \quad (17)$$

#### 1.4.7 Método de cuaterniones

De la misma forma que la metodología de Euler con los cuaterniones se comprende la rotación que se considera como única sobre un vector que se encuentra definida en el marco de referencia. El cuaternión es considerado como un vector tetradimensional y es necesario utilizar cuaterniones al momento de cargar los registros en OpenSim (*Algoritmo de estimación e implementación Software, s/f*).

### 1.4.8 Sistemas con sensores inerciales

Por definición los sensores inerciales son dispositivos con la capacidad de cuantificar la aceleración lineal o velocidad angular mediante acelerómetros o giroscopios respectivamente. A inicios de los años 20's se originó el desarrollo de diferentes aplicaciones de control, como sistemas de navegación y mecanismo guiados de manera automática, los cuales eran demasiado grandes y de difícil portabilidad lo que impulso al desarrollo de mecanismos de mejor movilidad (R. Martínez & Romero, 2013). Desde la aparición de un sensor inercial micro maquinado electromecánico creado por la universidad de Stanford ha facilitado el continuo desarrollo de estos mecanismos con mejores características y que se utilizan en diferentes sectores de salud y deporte. A continuación, se muestran sensores inerciales con disponibilidad en la actualidad y con características importantes.

**Tabla 1.** Sensores inerciales y características

Sensor	Ejes de medición	Rango de medición	Resolución	Voltaje de alimentación	Tamaño
<b>Acelerómetros</b>					
AIS328DQ (ST microelectronics)	3: x, y, z	± 2 g ± 6 g ± 8 g Seleccionable dinámicamente	30 µg/LSB	2.4 a 3.6 V	4 x 4 x 1.8 mm
ADXL345 (Analog devices)	3: x, y, z	± 2 g ± 6 g ± 8 g ± 16 g Seleccionable dinámicamente	1.95 mg/LSB	2.0 a 3.6 V	3 x 5 x 1 mm
<b>Giroscopios</b>					
L3G3200D (ST microelectronics)	Pitch, yaw, roll	2000 dps	3.8 dps /LSB	2.4 a 3.6 V	3 x 3.5 x 1 mm
<b>IMU's (Módulos de unidades inerciales)</b>					
LMS333D (ST microelectronics)	9 DOF x, y, z, pitch, roll, yaw, 3 ejes magnéticos	± 2 g ± 6 g ± 8 g ± 16 g ± 2 gauss ± 6 gauss ± 8 gauss ± 12 gauss ± 250 dps ± 500 dps ± 2000 dps	30 µg/LSB   30 µgauss/LSB   3.8 mdps /LSB	2.0 a 3.6 V	3.5 x 6 x 1 mm

Como se observa en la Tabla 1, los sensores inerciales presentados tienen la capacidad de medición hasta en los tres ejes de manera simultánea con alimentación de voltajes reducidos, además de la mejor precisión y eficaz portabilidad (R. Martínez & Romero, 2013).

#### 1.4.8.1 Usos prácticos de sensores inerciales en el área de salud

El uso de sensores inerciales ha facilitado al diagnóstico de varias enfermedades e incluso para rehabilitación para deportistas. En la siguiente Tabla 2 se describen algunos ejemplos de aplicación de los sensores (R. Martínez & Romero, 2013).

**Tabla 2.** Ejemplos de usos prácticos de sensores inerciales.

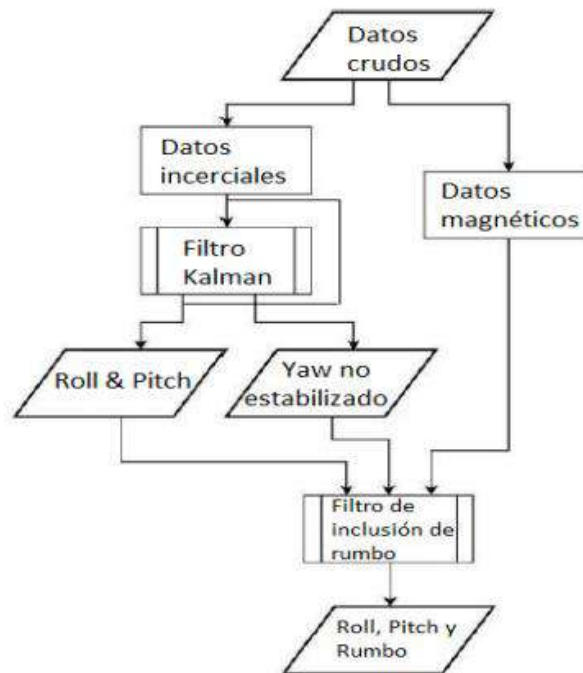
Usos prácticos	Ejemplos
Rehabilitación	<ul style="list-style-type: none"> <li>-Evaluación de la repetibilidad de los efectos del estrés en las articulaciones al marchar.</li> <li>-Valoración de la aceleración tanto en muslos y piernas en pacientes hemipléjicos.</li> <li>-Apreciación en ensayos que se han utilizado en pacientes hemipléjicos (Pozo et al., 2022).</li> <li>-Evaluaciones de consecuencias de pacientes con fracturas en el pie al usar diferentes calzados (Mayo clinic, 2022).</li> <li>-Seguimiento del ejercicio físico de pacientes que presentan enfermedades neurodegenerativas (J. Martínez, s/f).</li> </ul>
Deporte	<ul style="list-style-type: none"> <li>-Determinación de variables de marcha de alta velocidad</li> <li>-Valoración clínica del equilibrio en medicina deportiva</li> <li>-Mecanismo de evaluación del movimiento de balanceo en deportes como el golf o béisbol.</li> </ul>

#### 1.4.8.2 IMUs

Se conoce que los sensores inerciales IMUs tienen la capacidad de cuantificar la aceleración lineal y la velocidad angular, lo que es de gran ayuda en áreas como la salud y deporte. Además, los IMUs son útiles en fisioterapia para la evaluación del balanceo postural y cuya inconsistencia sería señal de enfermedades neurodegenerativas, otro uso sería la evaluación de movimientos bajo el agua. También se determina la cinemática y el comportamiento angular vertebral. El uso de IMU permite retroalimentar actividades que se realizan en rehabilitación conjuntamente el control postural y un correcto rango de movimiento (Castellanos et al., 2020). Los sensores inerciales IMUs se utilizan en el presente documento para la obtención de los datos del movimiento para la calibración y validación en modelos biomecánicos.

Existen algunas ventajas de los sensores inerciales IMUs integrados que permiten el

procesamiento de datos para la obtención de diferentes parámetros como el roll, pitch y yaw, es decir alabeo, elevación y dirección respectivamente. A continuación, se indica un diagrama de representación del tratamiento de datos (Cuadros, 2020).



**Figura 7.** Diagrama del tratamiento de datos de sensores inerciales (Cuadros, 2020).

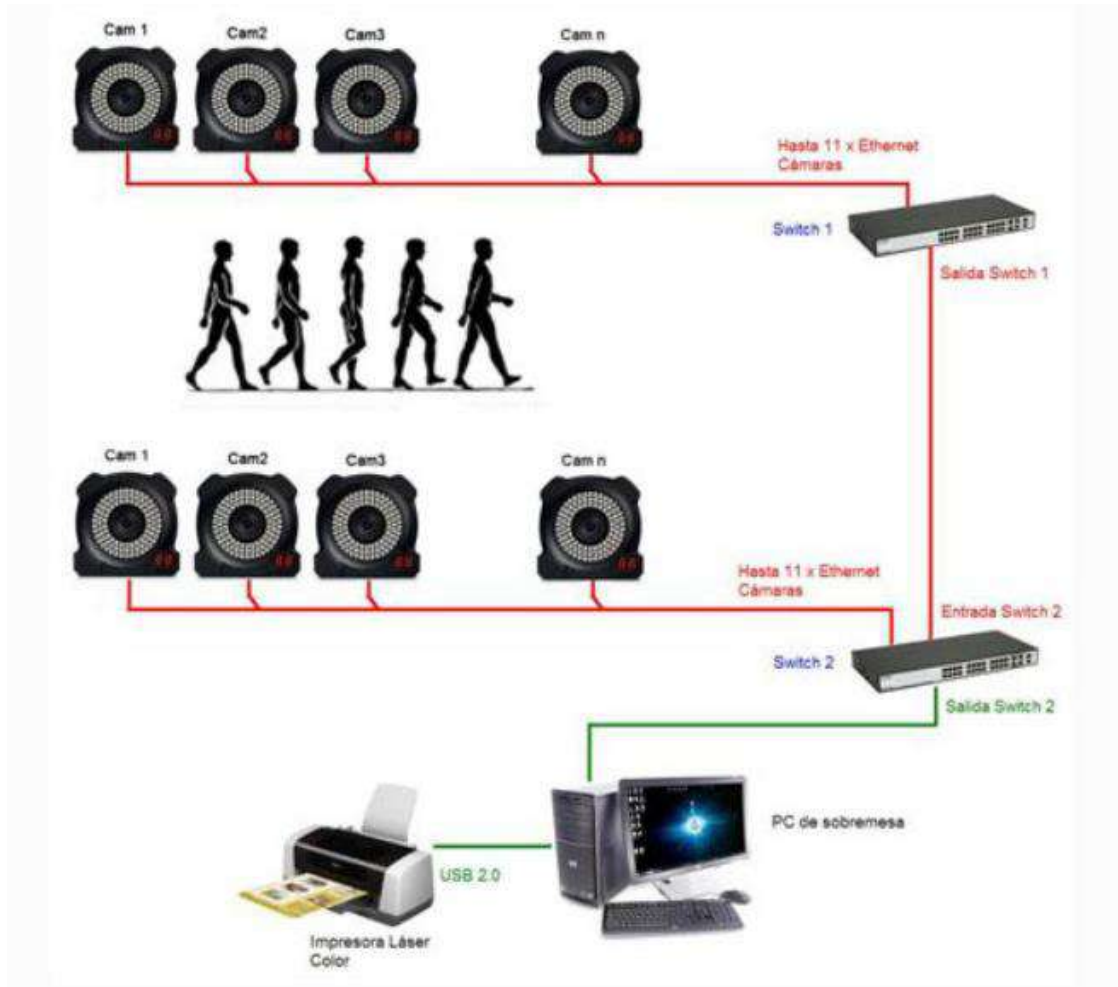
#### 1.4.8.3 EMG

EMG es una técnica utilizada para detectar, analizar y usar señales eléctricas para la detección de contracción del músculo. En EMG se priorizan la amplitud y frecuencia de las señales (Guzmán & Méndez, 2018). Sin embargo, para el presente documento no se utilizará esta técnica, aunque se recomendaría para futuras investigaciones.

#### 1.4.8.4 Fotogrametría

La fotogrametría es una herramienta que facilita a la definición con precisión la dimensión, posición y forma de un objeto. Esto se consigue analizando las medidas recolectadas en secuencias fotográficas o de video. Otra definición pertinente para mencionar es la capacidad para obtener con precisión y confiabilidad las medidas de los objetos analizados. Los experimentos de fotogrametría se realizarán en el laboratorio de bioingeniería de la Escuela Politécnica Nacional que cuenta con el equipamiento adecuado (Trujillo, 2022).





**Figura 8.** Esquema de registro de datos por fotogrametría (Instituto de biomecánica de Valencia, s/f)

#### 1.4.9 OpenSim

OpenSim es un software de código abierto que se utiliza para el modelado neuromusculoesquelético, simulación y análisis. OpenSim se encuentra escrito en ANSI C++, y en java la interfaz gráfica de usuario. Lo que facilita la ejecución en distintos sistemas operativos de uso común.

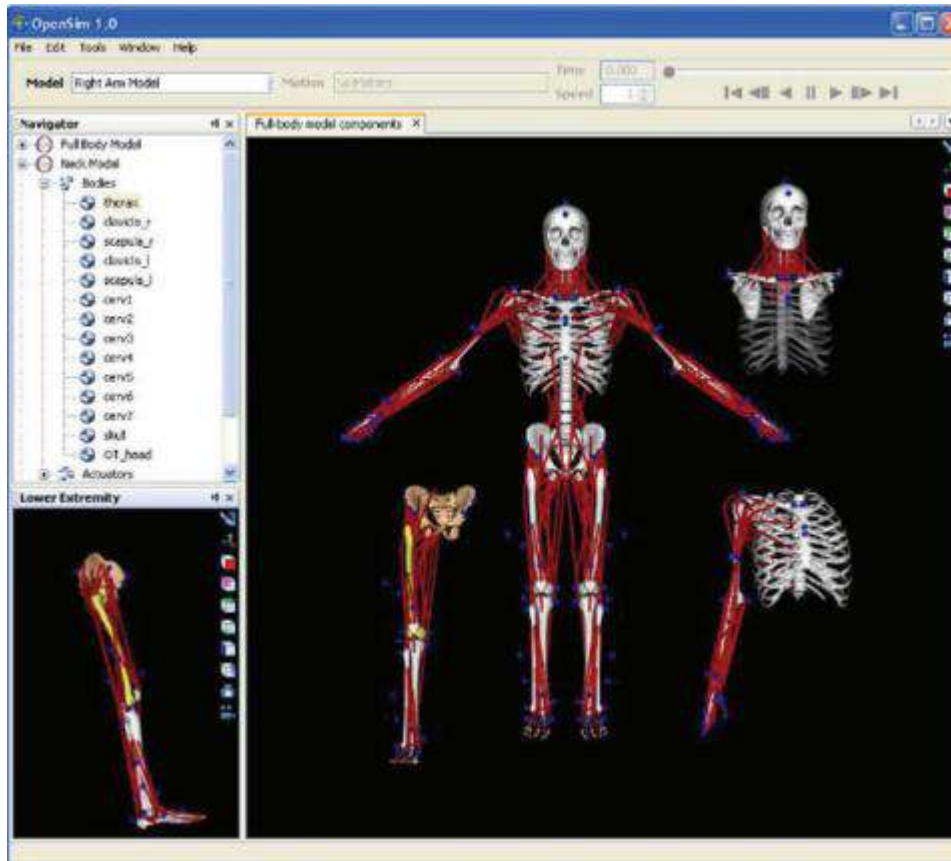
En OpenSim se desarrollan y analizan modelos musculoesqueléticos además de simulaciones.

Algunas características mencionables de este software son:

La captura fotográfica de modelos musculoesqueléticos y la creación de películas de animación.

Obtención de gráficos de los resultados de los análisis realizados.

La realización de un análisis cinemático inverso para el cálculo de ángulos a partir de la posición de los marcadores utilizados y el análisis de modelamientos dinámicos. A continuación, se indica la estructura de un modelo musculoesquelético (Freire, 2019).

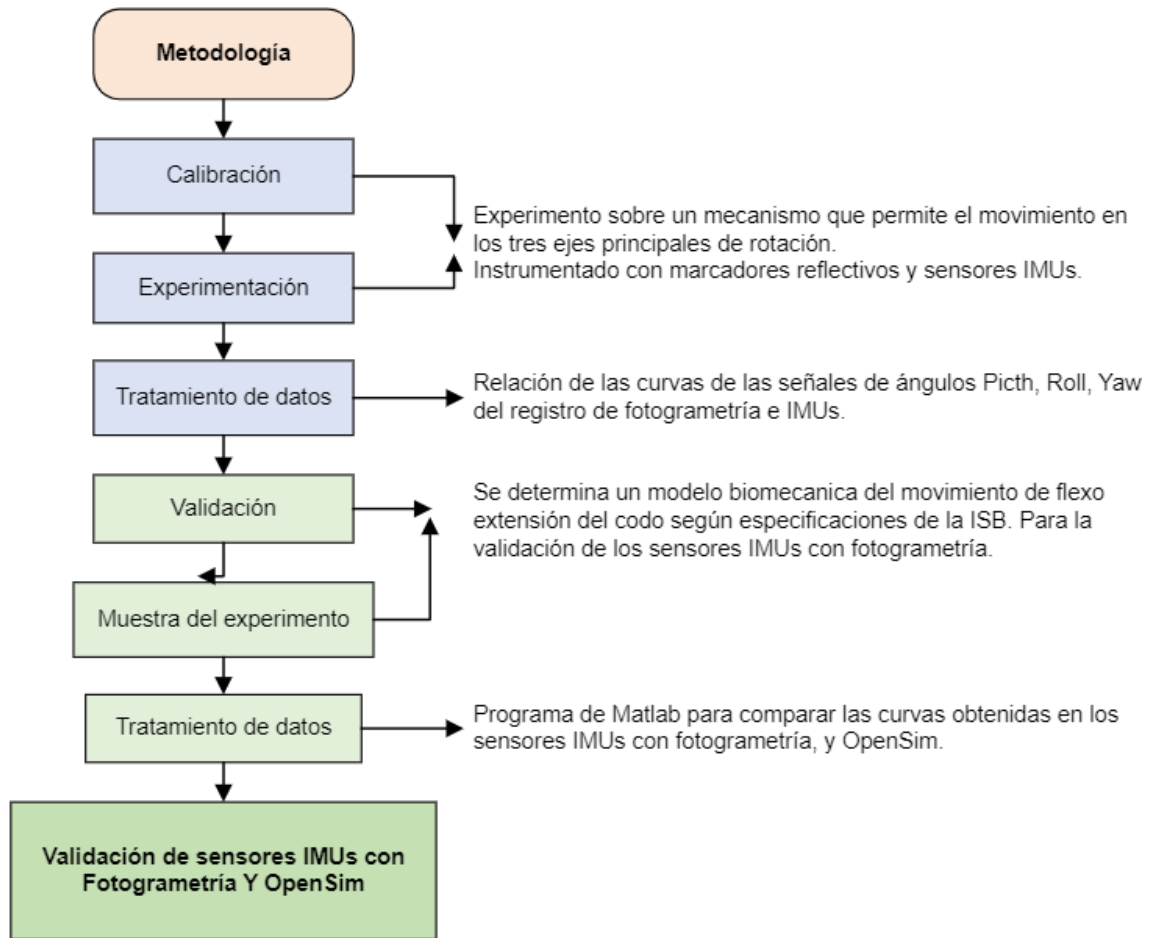


**Figura 9.** Modelo musculoesquelético en plataforma OpenSim

## 2 METODOLOGÍA

Para desarrollar la metodología para la validación de técnicas basadas en sensores inerciales IMUs para la actividad física es necesario dividir en dos secciones importantes tanto la calibración como la validación, los experimentos se realizan en el laboratorio de bioingeniería de la Escuela Politécnica Nacional. Para la calibración se instrumenta con marcadores y con sensores IMUs a un mecanismo que permite el movimiento de rotación de los tres ejes. Se realiza el movimiento en los ejes coordenados locales del IMUs y sus respectivas rotaciones. Posteriormente se realiza el tratamiento de datos con los registros de fotogrametría e IMUs con ayuda de un programa desarrollado en Matlab en donde se comparan las señales de los ángulos Pitch, Roll, Yaw.

Para la segunda fase de la metodología es necesaria la validación de los sensores IMUs con los de fotogrametría y adicionalmente con una simulación en OpenSim. Por lo que se determina un modelo biomecánico del movimiento de flexo extensión del codo con especificaciones para la ubicación de los marcadores y sensores según la ISB. Posteriormente al igual que la calibración para el tratamiento de datos se utiliza un programa de Matlab (especificado en la sección de validación) y OpenSim. Finalmente, con las curvas obtenidas se compara la fiabilidad de la utilización de los sensores IMUs en los diferentes campos de aplicación como reemplazo de la fotogrametría. A continuación, se indica el esquema gráfico de la metodología.



**Figura 10.** Esquema de metodología.

## 2.1 Desarrollo de programa para el tratamiento de datos en la calibración

Se establece un programa en Matlab para el tratamiento de datos tanto para el eje X, Y, Z. En donde se comparan las curvas obtenidas en fotogrametría e IMUs. A continuación, se describe brevemente la estructura del programa.

1. Para el inicio del programa realizado en Matlab es necesario cargar los registros de calibración tanto de los sensores IMUs como los de fotogrametría.

```

clear;clc;
% Cargar registros de calibración IMUs
datos_imus_cal=xlswread('GIRO_ORIENTACION_X_Plot_and_Store_Rep_1.3.xlsx');
datos=dlmread('GIRO_ORIENTACION_X.txt',' ',1,0);
  
```

2. Posteriormente es necesaria la filtración de aquellas celdas que contengan singularidades como espacios vacíos, 0 y 180 de valor de los datos del registro del sensor IMUs, Además es necesario el desplazamiento del tiempo tomando en cuenta las celdas que contenían cero en el registro IMUs. Se determina el número de filas, y se linealiza el tiempo con pasos iguales (Ferrer, 2015).

```

%Filtro las celdas en cero para el registro IMUs
posZero=find(datos_imus_cal(:,6)==0);
datos_imus_cal(posZero(1:end),:)=[];

[ni nj nk]=size(datos_imus_cal);
for i=1:2
    tiempo(:,i)=datos_imus_cal(:,5+(i-1)*10); %Señal de tiempo
    imu_p_cal(:,i)=datos_imus_cal(:,6+(i-1)*10); %Señal de ángulo Pitch
    imu_r_cal(:,i)=datos_imus_cal(:,8+(i-1)*10); %Señal de ángulo Roll
    imu_y_cal(:,i)=datos_imus_cal(:,10+(i-1)*10); %Señal de ángulo Yaw
end
posZero=find(nanmean(imu_p_cal,2)==180);
imu_p_cal(posZero(1:end),:)=[];
posZero=find(nanmean(imu_r_cal,2)==0);
imu_r_cal(posZero(1:end),:)=[];
posZero=find(nanmean(imu_y_cal,2)==180);
imu_y_cal(posZero(1:end),:)=[];
tiempo(posZero(1:end),:)=[];
t=tiempo(:,1)-tiempo(1,1); %Desplazo el tiempo tomando en cuenta los las celdas
IMUs en cero,
n=max(size(t)); %Halla el numero de filas
time_IMUs=t(end)*(0:n-1)'/n; %Vuelvo a linealizar el tiempo con pasos iguales

```

3. Es indispensable la linealización para la disminución de singularidades en las curvas de los ángulos. Con ayuda de la función desarrollada Filtro\_singularidades (Anexo I) se filtran los datos irregulares obtenidos de la experimentación.

```

[imu_p_cal]=Filtro_singularidades(imu_p_cal,'MAX');
[imu_r_cal]=Filtro_singularidades(imu_r_cal,'0');
[imu_y_cal]=Filtro_singularidades(imu_y_cal,'MAX');
[imu_r_cal]=Filtro_singularidades(imu_r_cal,'MAX');

imu_e(:,1)=imu_p_cal; %imu Euler
imu_e(:,2)=imu_r_cal;
imu_e(:,3)=imu_y_cal;

Phi_imu=permute(imu_e,[1 3 2]);
Phi_imu(:,[1 2],:)=Phi_imu(:,[1 2],:);

```

4. El sistema IMUs genera una singularidad al arranque por lo que es necesario dividir este registro en 1 segundo, además se realiza el desplazamiento tomando en cuenta las celdas del registro en cero y se halla el desfase para poder lograr la sincronización con el sistema de fotogrametría.

```

%El sistema IMUs genera una singularidad al arranque por lo que se corta
%este registro en 1 segundo
corte=find(time_IMUs<1);
Phi_imu(corte,:)=[];
time_IMUs(corte)=[];
t_imu=time_IMUs(:,1)-time_IMUs(1,1); %Desplazo el tiempo tomando en cuenta los las
celdas IMUs en cero,
%Se halla el desfase para poder sincronizar con el otro sistema
pos_desfase_imu=desfase_H(Phi_imu(:,1),t_imu);
offset_imu=nanmean(Phi_imu(1:pos_desfase_imu/3,1));

```

Con el adecuado tratamiento de los datos se puede generar gráficas de los resultados, los cuales pueden ser contrastados posteriormente con las curvas de fotogrametría.

5. Para el análisis de la fotogrametría se cargan los registros.
6. Es necesario hallar el número de filas y el número de frecuencia para determinar el

vector tiempo, el que debe ser linealizado en pasos iguales. Además, es fundamental desplazar el tiempo tomando en cuenta las celdas en cero del registro IMUs.

7. Se determinan las matrices de posiciones para el cálculo cinemático y la postura estática de referencia.

```
%ANALISIS POR FOTOGRAMETRIA
n=max(size(datos)); %Halla el numero de frames o filas

fps=200; %frames por segundo o frecuencia en Hz
t=(0:(n-1))/fps; %Halla el vector de tiempo, linealizar el tiempo con pasos
iguales
corte=find(t<1);
datos(corte,:)=[];
t(corte)=[];
t_fot=t(:,1)-t(1,1); %Desplazo el tiempo tomando en cuenta los las celdas IMUs en
cero,
n=max(size(datos)); %Halla el numero de frames o filas
%Matrices de posiciones
MC = calcM(datos,[1:3]); %[m] % cálculo cinemático
MC0=MC(1:500,:,:)
MC0=nanmean(MC0,1); %Postura estatica de referencia
MC0(1, :, 4)=(MC0(1, :, 1)+MC0(1, :, 2)+MC0(1, :, 3))/3;
```

8. Se realiza el cálculo cinemático con los datos y se determina las posiciones relativas. El cálculo cinemático MC se almacena en una variable auxiliar con subíndice 1 para todas las posiciones que se encuentran localizadas en el centro de los marcadores. De la misma forma se halla la nueva posición relativa de las coordenadas en movimiento y tiempo con respecto al punto de referencia y a la vez se realiza la transformación de las coordenadas.

```
%Se determina las posiciones relativas
MC1=MC; %Almaceno en una variable auxiliar con indice sub 1 para todas las
posiciones que van en el centro de los tres marcadores
for i=1:n
    for j=1:3;
        MC1(i, :, j)=(MC(i, :, j)-MC0(:, :, 4));
    end
end

for j=1:3;
    MC01(1, :, j)=(MC0(1, :, j)-MC0(:, :, 4));
end

[omegac ,svoc,RGv,sGv,M,RG] = finitosolid(MC1(:, :, [1:3]),MC01(:, :, [1:3])); %
Cinematica de X

for i=1:n
    XYZ(i, :) = convertirrotacion(omegac(i, :), 'rodrigues', 'xyz');
end
```

9. Se define la variable principal del par que será fi. Además, se considera el desfase horizontal.



```

%Definimos la variable principal del par que será fi
Fi_fot=180*XYZ(:, :)/pi;

pos_desfase_fot=desfase_H(Fi_fot(:, :),1),t_fot);

offset_fot=nanmean(Fi_fot(1:pos_desfase_fot/3, :,1));
offset=abs(offset_imu-offset_fot);

%Desfase horizontal
t1=t_imu-t_imu(pos_desfase_imu);
t2=t_fot-t_fot(pos_desfase_fot);

t_imus=t1-min(t2);
t_fot=t2-min(t2);

if t_imus(1)<0
posneg=find(t_imus<0);
t_imus(posneg, :, :)=[];

Phi_imu(posneg, :, :)=[];
Fi_imu=Phi_imu;
t_imu=t_imus-t_imus(1);
else
dt_imus=t_imus(2)-t_imus(1);
tr(:,1)=0:dt_imus:t_imus(1);
n=max(size(tr));
t_imu=[tr;t_imus];
fi_offset=offset_imu.*ones(n,3,2);
Fi_imu=[fi_offset;Phi_imu];
end

fdamat = suavizado_bspline(Fi_fot, t_fot, 200, 4,t_imu);
fi_fot(:, :,1)=fdamat(:, :,1);
Dfi_fot(:, :,1)=fdamat(:, :,2);
D2fi_fot(:, :,1)=fdamat(:, :,3);

```

10. Se establece el código para representar gráficamente las curvas comparativas de los ángulos de Euler de los registros de IMUs y de fotogrametría.

```

%:::::Validación:::::

%'X_Pitch_offset'
Xc(:,1,1)=Fi_imu(:,1,1);
Xc(:,1,2)=fi_fot(:,1);

[Max Min Rango Media Mediana SEM DesvStan Varianza Sig_F F_test Sig_Q Q_test
Percentil IQR fxi xi Icc CMC CMCm cmc Pearson]=ICorre_sesiones(2, 'single',Xc);
disp(['X_Pitch_offset CMC = ',num2str(sqrt(abs(cmc))])

figure(1)
hold on
plot(t_imu,Fi_imu(:,1), 'Color',[0.6 0.6 0.6], 'LineWidth',2)
plot(t_imu,fi_fot(:,1), 'k-.', 'LineWidth',2)
ax = gca;
ax.FontName='Times New Roman';
ax.FontSize=16;
ax.GridLineStyle = '-';
ax.GridColor = [0 0 0];
grid on;box on;
xx = ax.XLim;
yy = ax.YLim;
xlabel('Tiempo ( s )', 'fontsize', 18);
ylabel('Posición angular Pitch ( ° )', 'fontsize', 18);
legend('IMUs', 'Fotogrametría', 'Location', 'northeast', 'NumColumns',1, 'fontsize', 14, 'FontName', 'Times New Roman', 'Position', [0.20 0.2 0.1 0.2]);
legend('boxoff')
text(0.5,yy(1)+3, ['offset ', num2str(offset(1))], 'fontsize', 12)
text(10,yy(1)+3, ['CMC ', num2str(sqrt(abs(cmc)))] , 'fontsize', 12)
print('-djpeg', '-r500', '-painters', ['X_Pitch_offset', '.jpg']);

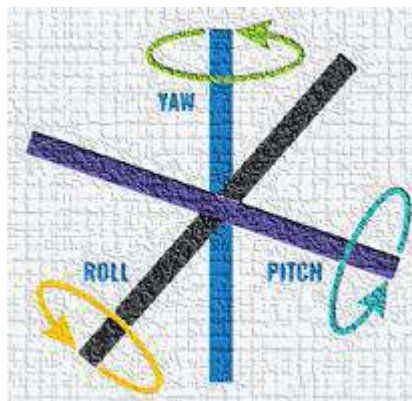
```

De la misma forma se realiza en los ángulos Yaw y Roll.

## 2.2 Calibración

La calibración es fundamental para asegurar la precisión y fiabilidad de los datos obtenidos de los sensores IMUs, mientras que la fotogrametría se basa en la captura de imágenes de un objeto que está en movimiento mediante cámaras para convertirlas en datos tridimensionales. Es esencial comparar y validar los resultados obtenidos mediante la calibración de los sensores IMUs y la fotogrametría para garantizar la exactitud y confiabilidad de los datos, lo que es crucial para la aplicación en diversos campos (Samatas & Pachidis, 2022).

Para realizar la calibración se establecen 3 movimientos angulares, tanto en el eje X, Y, Z. Y con ayuda de marcadores reflectivos y un sensor IMU se fijan adecuadamente sobre un mecanismo que permite el movimiento rotacional.



**Figura 11.** Ejes de rotación de IMUs (Cuadros, 2020)

Se conoce que los IMUs cuentan con tres ejes de rotación como se observa en la figura 11, que además son llamados como ejes de giro, que son: el eje X (eje de cabeceo), el eje Y (eje de balanceo) y el eje Z (yaw). Los ejes mencionados son usados para la medición de la orientación y el desplazamiento de un elemento en un espacio tridimensional.

### 2.2.1 Materiales

Los materiales descritos a continuación son esenciales al momento de utilizar los sistemas de fotogrametría e IMUs del laboratorio de bioingeniería de la facultad de ingeniería mecánica de la EPN.

#### 2.2.1.1 Marcadores

Los marcadores utilizados tienen forma esférica y son hechos de materiales reflectantes que facilitan la detección de la posición y movimiento en el espacio tridimensional del objeto analizado por las videocámaras del sistema de fotogrametría. Los marcadores utilizados son tres y con ayuda de una estructura de sujeción se ubican como se muestra en la figura 12.



### 2.2.1.2 Cinta adhesiva

La cinta adhesiva es ideal para la fijación de los marcadores e IMUs en el mecanismo de rotación durante el proceso de calibración.

### 2.2.1.2 Sensores inerciales IMUS

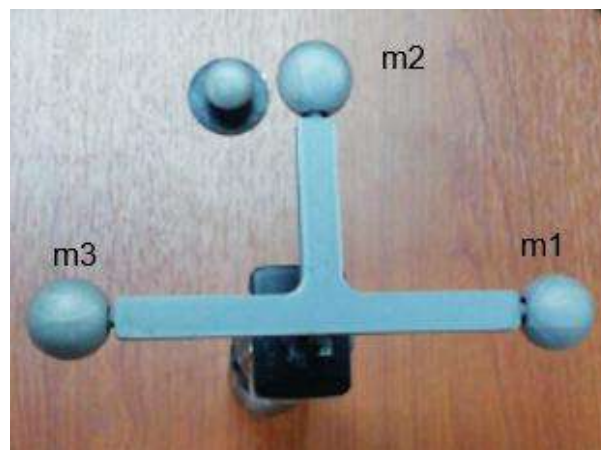
Como sea mencionado para la obtención de datos confiables es pertinente realizar la calibración de los sensores, es así como se utiliza solamente un sensor IMU ubicado por debajo de los marcadores reflectantes, debido a que el espacio del mecanismo de rotación cuenta con espacio limitado.

### 2.2.1.3 Método de fotogrametría

El laboratorio de bioingeniería cuenta con el sistema de fotogrametría que se ha utilizado para la experimentación. El sistema usa el software Kinescan instalado en un ordenador que facilita la obtención de los registros de los marcadores reflectantes por medio de un sistema interconectadas de cámaras alrededor de la habitación.

## 2.2.2 Instrumentación

La instrumentación se realiza sobre un mecanismo que permite el movimiento de rotación en los tres ejes principales de rotación. A continuación, se indica la posición de los marcadores reflectivos utilizados.



**Figura 12.** Instrumentación para la calibración (Fuente: propia)

Se observa en la figura 12 que el sensor IMU se ubica por encima del mecanismo de rotación, y por encima de este los tres marcadores reflectantes formando un triángulo tal como se indica en la siguiente figura 13.



**Figura 13.** Mecanismo rotativo, IMU y marcadores. (Fuente: propia)

Los marcadores y el sensor IMU se fija en el mecanismo con ayuda de cinta adhesiva.

### 2.2.3 Captura de datos por fotogrametría e IMUs

Para la captura de datos es necesario realizar tres movimientos, el primer movimiento se realiza alrededor del x al rotar el mecanismo en un rango de  $170^\circ$ , de la misma forma alrededor del eje Y o Z, el rango de movimiento en este eje es de  $180^\circ$ . Los movimientos se repiten durante 5 ciclos. Al inicio de cada toma de datos el mecanismo permanece estático durante 5 segundos.

### 2.2.4 Tratamiento de datos en Programa Matlab

El programa de Matlab utilizado se detalla en la sección 2.1. para los tres movimientos en los ejes de rotación comparando los sensores IMUs y fotogrametría.

#### 2.2.4.1 Calibración del eje X

Del registro de calibración del eje x del IMU se obtienen las señales de los ángulos pitch, roll, yaw, y de los datos de fotogrametría se obtiene los componentes del ángulo  $\varphi$ . Cuyas curvas semejantes relacionadas se indican en la siguiente tabla 3.

**Tabla 3.** Relación de variables de IMUs y fotogrametría calibración eje X

Señales de ángulos (IMUs)	Señales del ángulo $\varphi$
pitch	1
roll	2
yaw	3

Como se observa en la tabla 3 se relacionan los ángulos pitch, roll, yaw con las coordenadas del ángulo  $\varphi$  1, 2, 3 respectivamente.

#### 2.2.4.2 Calibración del eje Y

De la misma forma que se realizó en el X se realiza en el eje Y. En la tabla 4 se indica la relación de variables de los ángulos del IMU y fotogrametría.

**Tabla 4.** Relación de variables de IMUs y fotogrametría calibración eje Y

Señales de ángulos (IMUs)	Señales del ángulo $\varphi$
pitch	1
roll	2
yaw	3

#### 2.2.4.3 Calibración del eje Z

Asimismo, se realiza el tratamiento de los registros de calibración en el eje Z.

**Tabla 5.** Relación de variables de IMUs y fotogrametría calibración eje Z

Señales de ángulos (IMUs)	Señales del ángulo $\varphi$
pitch	1
roll	2
yaw	3

### 2.2.5 Simulación en OpenSim

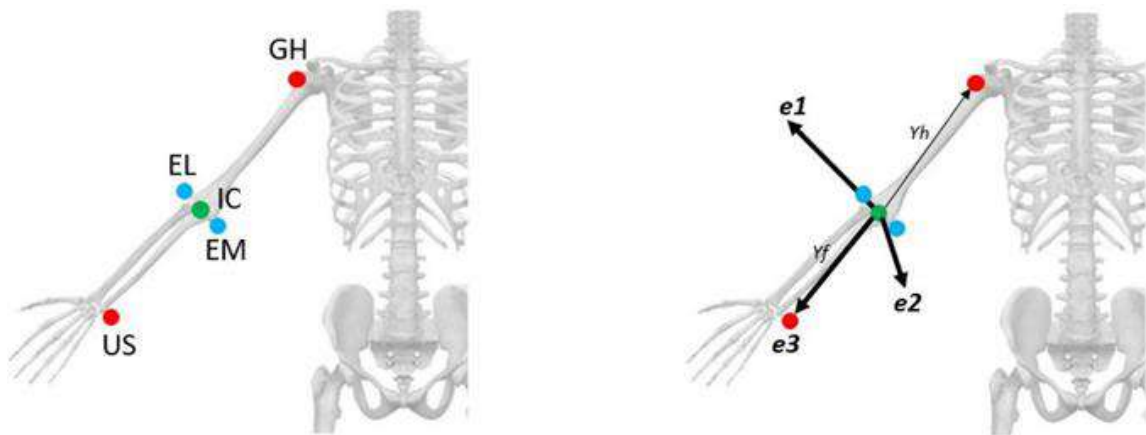
Para la simulación de los registros recolectados de calibración de IMUs y fotogrametría en OpenSim es necesario cambiar el formato para que sea compatible con el software. Los datos recolectados de IMUs se obtienen en formato .xlsx (archivo de Excel), mientras que los registros de fotogrametría se capturan en formato .txt (archivo de texto).

Los formatos aceptables en el software OpenSim son: trc, sto, mot, csv, c3d, xml, vtp entre otros.

## 2.3 Validación

Para la validación de las técnicas utilizadas se determina un modelo biomecánico de la parte superior específicamente del codo, por lo que es necesario la implementación del sistema de coordenadas de esta articulación según la sociedad internacional de Biomecánica (ISB) (*Descripción de los modelos implementados*, 2020).

Según la ISB los ejes para la articulación del codo se posicionan como se indica en la siguiente figura 14.



**Figura 14.** Ejes de la articulación del codo según la ISB. (*Descripción de los modelos implementados*, 2020)

Donde:

GH = articulación glenohumeral se reemplaza por el punto acromioclavicular.

EL= epicóndilo lateral

EM= epicóndilo medial

IC= Mitad de EL y EM

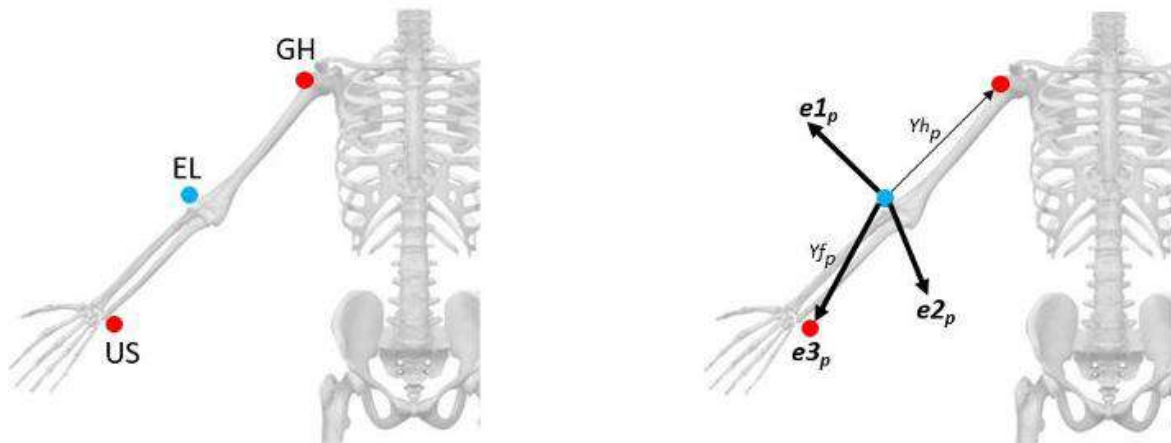
US= estiloides cubital

Además,  $e1$  se establece la actividad de flexo extensión las cuales son positivas y negativas respectivamente. El vector  $Y_h$  se determina por el producto vectorial entre la línea interconectada GH-IC y el vector  $Y_f$  que es la conexión lineal de US-IC.

$e3$  se encuentra definido como el movimiento de pronación supinación donde los movimientos son positivos y negativos respectivamente. El vector  $Y_f$  es definido por la conexión de US-IC.

Mientras que el  $e2$  es determinado por el producto vectorial entre  $e1$  y  $e3$ .

Sin embargo, la aplicación directa de la solución de la ISB puede dificultar la visibilidad de algunos marcadores que se encuentran ocultos por lo que se recomienda en caso de presentar este error reemplazar IC por EL (*Descripción de los modelos implementados*, 2020). Tal como se indica en la siguiente figura 15.



**Figura 15.** Ejes de articulación de codo reemplazando IC. (*Descripción de los modelos implementados*, 2020)

### 2.3.1 Desarrollo de protocolo de pruebas

Se pretende evaluar la posición de los marcadores y los sensores IMUs para validar el modelo biomecánico de la articulación del codo. La localización de los sensores y marcadores dependen de las ubicaciones anatómicas propuestas por la ISB (Wu et al., 2005).

#### 2.3.1.1 Muestra del experimento

Según la ISB se recomienda utilizar 14 sujetos de prueba homogéneos siete mujeres y siete hombres de edades similares (Instituto Biomecánico de Valencia, 2020). Edades desde los 20 a los 35 años. Además, los sujetos estudiados no deben presentar ningún padecimiento o enfermedad que limite el movimiento de flexo extensión del codo.

Se tienen varias recomendaciones antes de iniciar con la experimentación:

- a) Los sujetos de estudio deben vestir camisas sin mangas para la ubicación de los sensores de forma adecuada. Es importante sujetar el cabello antes de iniciar el experimento.
- b) Los sujetos de estudio deben estar informados del tipo de experimentación que se realizará y el objetivo de este, además es esencial que los sujetos firmen el consentimiento.

c) Los sujetos deben proporcionar su peso, su edad, su sexo y talla antes de la instrumentación.

*2.3.1.2 Materiales*

Los materiales, equipos y softwares utilizados en la experimentación son:

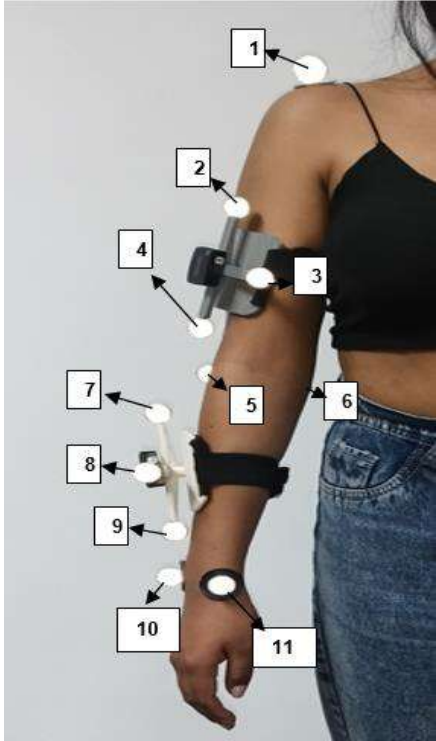
- Marcadores reflectivos, cinta adhesiva
- Instrumentos de sujeción de marcadores
- Sensores IMUs
- Sistema de fotogrametría y software Kinescan
- Software OpenSim
- Cámara de celular

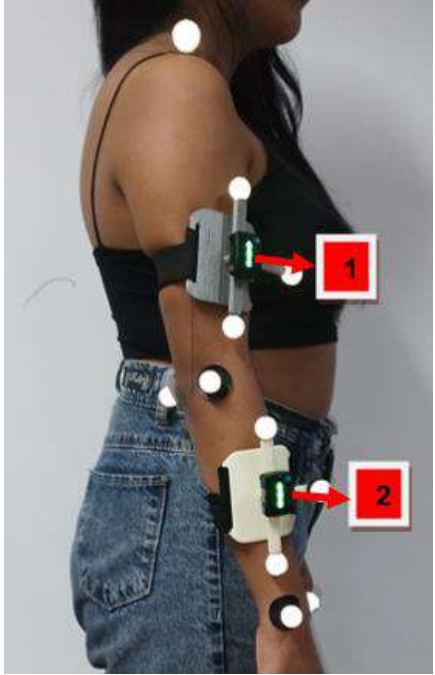
*2.3.1.3 Instrumentación*

*2.3.1.3.1 Ubicación de sensores y marcadores en el sujeto*

Los marcadores utilizados se ubican de acuerdo con las recomendaciones de la ISB.

**Tabla 6.** Ubicación de los sensores y marcadores

Marcadores	Localización	Fotografía
M1	M1 se ubica en el acromio.	
M2, M3, M4	Se ubican en la parte proximal del brazo formando un triángulo.	
M5, M6	M5 se ubica en el epicóndilo lateral y M6 se ubica en el epicóndilo lateral.	
M7, M8, M9	Parte distal del antebrazo formando un triángulo.	
M10, M11	Se ubican cerca de la muñeca M10 en la estiloides cubital y M11 en la estiloides radial.	

Sensores	Localización	Fotografía
IMU1	IMU1 se ubica en la parte proximal del brazo.	
IMU2	IMU2 se ubica en la parte distal del antebrazo.	

De la misma forma para los sensores IMUs, para la medición del movimiento de flexo extensión del codo es recomendable la colocación de dos sensores IMUs tanto en la parte distal del antebrazo y en la parte proximal del brazo. Los criterios utilizados para la ubicación son:


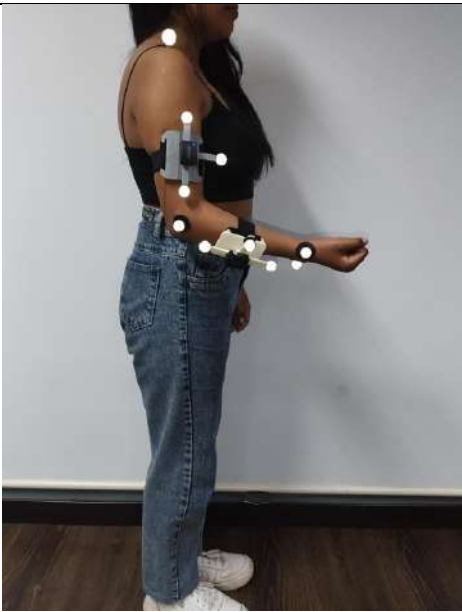
En la parte distal del antebrazo, la ubicación del sensor es en el tercio distal del antebrazo, en la línea media sagital del antebrazo, con el eje x en dirección distal, el eje Y en dirección medial y el eje z en dirección dorsal.

En la parte proximal del brazo, el sensor es ubicado en la cara lateral del brazo a una distancia aproximada de un tercio de la longitud del brazo medida desde el epicóndilo lateral del húmero hasta el acromio, el eje X en dirección medial, el eje Y en dirección distal y el Z en dirección posterior.

#### *2.3.1.4 Posición y orientación de sujeto*

La experimentación se divide en dos fases, la de calibración y movimiento de flexo extensión del codo. En la siguiente tabla 7 se indica la posición y orientación del sujeto.

**Tabla 7.** Posición y orientación del sujeto

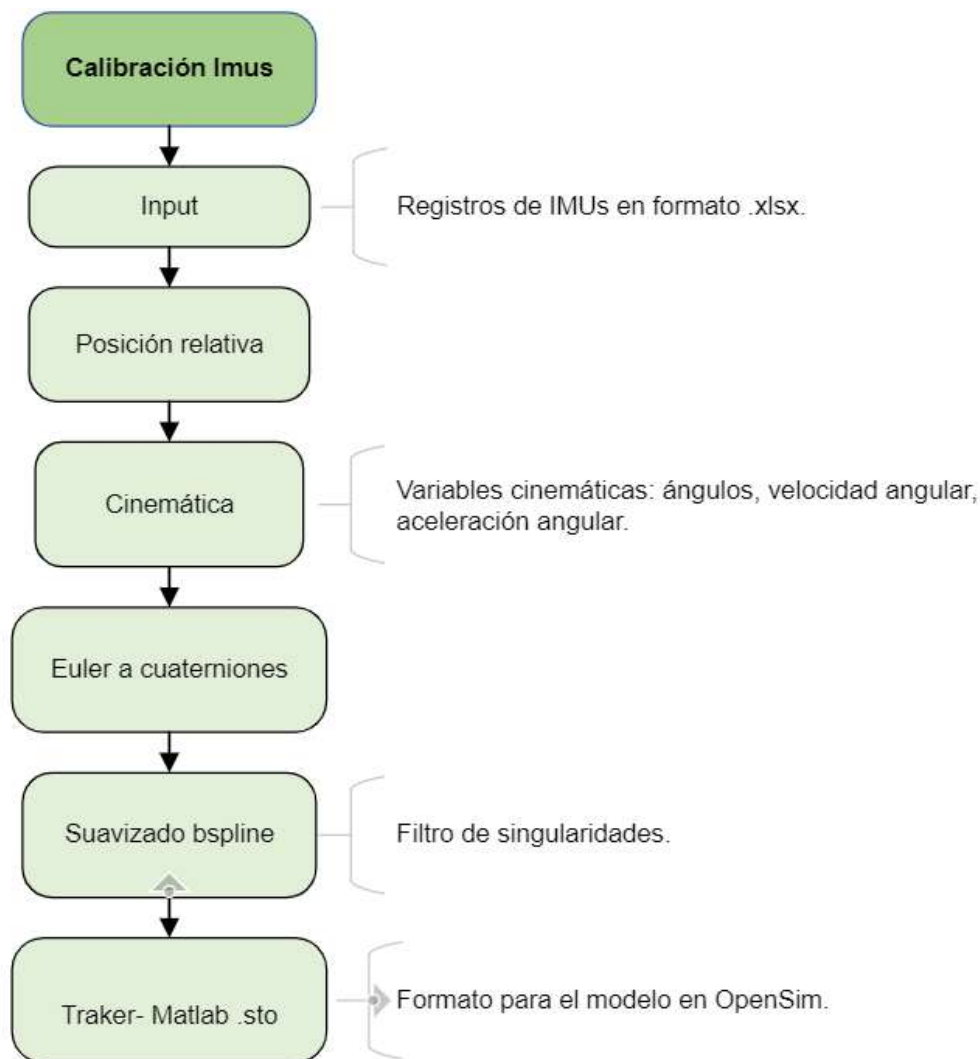
Movimiento	Posición y orientación	Fotografía
Calibración	<p>Duración 5 segundos</p> <p>Sujeto permanece de pie, con los pies juntos y verificando los tres ejes coordenados del sistema.</p>	
Posición de referencia	<p>El sujeto permanece de pie, con el brazo y el antebrazo creando un ángulo recto. El sujeto debe mantenerse inmóvil mientras se inicia el movimiento y durante la calibración.</p>	
Flexo extensión de codo	<p><i>Inicio de la experimentación</i></p> <p>Duración de 60 segundos</p> <p>Sujeto permanece de pie, el brazo derecho flexiona el codo a 90° al iniciar y con antebrazo en posición neutral. Realizando movimiento de extensión máxima y de la misma forma de flexión máxima. Se repite el procedimiento durante varios ciclos hasta finalizar el registro.</p> <p>Se recomienda replicar el registro al menos dos veces para la correcta obtención de registros.</p>	

### 2.3.2 Tratamiento de datos y procedimiento



### 2.3.2.1 Calibración del experimento registros IMUs

Como se indicó en la tabla 7 antes de la experimentación es necesario realizar el proceso de calibración para verificar el funcionamiento adecuado de los sensores para el posterior tratamiento de datos obtenidos. A continuación, se indica un esquema gráfico para resumir el programa desarrollado en Matlab para el tratamiento de datos de la calibración de los sensores IMUs en el experimento y preparación de archivos para la simulación en OpenSim. El programa de Matlab completo se encuentra en el Anexo II.



**Figura 16.** Esquema de programa en Matlab de calibración en IMUs

El filtro de singularidades es un logaritmo desarrollado en Matlab que permite reducir irregularidades en los datos, la programación completa se encuentra en el Anexo I.

### 2.3.2.2 Simulación en OpenSim IMUs

Para realizar la simulación en OpenSim es necesario utilizar los datos de los registros IMUs lo cuales deben ser cargados en formatos admisibles en el software, es así como se desarrolla un script en Matlab para poder cargar los archivos en OpenSim. Es

recomendable transformar los registros de IMUs a cuaterniones para poder realizar la simulación en OpenSim. A continuación, se indica parte del script de Matlab del anexo II desarrollado para la preparación de archivos para cargar en OpenSim en un formato aceptable.

Primero los registros son transformados en cuaterniones y son almacenados en estas variables variables\_imu. Los sensores IMUs utilizados son nombrados r\_humerus\_imu y r\_ulna\_radius\_hand\_imu. Como se indica a continuación.

```
dat_imu_cuat(:,1)=t_imu;
dat_imu_cuat(:, [2 3 4 5])=Q1;
dat_imu_cuat(:, [6 7 8 9])=Q2;

save resultados.mat dat_fot_euler dat_imu_euler dat_fot_cuat dat_imu_cuat

[ni nj]=size(dat_imu_cuat);

Variables_imu(1,:)={'DataRate=74.000000',' ',' '};
Variables_imu(2,:)={'DataType=Quaternion',' ',' '};
Variables_imu(3,:)={'version=3',' ',' '};
Variables_imu(4,:)={'OpenSimVersion=4.1',' ',' '};
Variables_imu(5,:)={'endheader',' ',' '};
Variables_imu(6,:)={'time','r_humerus_imu','r_ulna_radius_hand_imu'};
```

Después de generar el archivo en Matlab se lo guarda con el nombre Traker\_imu\_cuat.sto. Para poder cargar los registros en OpenSim para realizar la simulación. Se reorganiza el registro con los datos de fotogramas y tiempo como se muestra a continuación.

```
for i=1:ni
    for j=1
        Variables_imu{i+6,j}=dat_imu_cuat(i,j);
    end
end

for i=1:ni
    Variables_imu{i+6,2}=[num2str(dat_imu_cuat(i,2)), ',', num2str(dat_imu_cuat(i,
3)), ',', num2str(dat_imu_cuat(i,4)), ',', num2str(dat_imu_cuat(i,5))];
    Variables_imu{i+6,3}=[num2str(dat_imu_cuat(i,6)), ',', num2str(dat_imu_cuat(i,
7)), ',', num2str(dat_imu_cuat(i,8)), ',', num2str(dat_imu_cuat(i,9))];
end

diary Traker_imu_cuat.sto
fprintf([Variables_imu{1,1}, '\n']);
fprintf([Variables_imu{2,1}, '\n']);
fprintf([Variables_imu{3,1}, '\n']);
fprintf([Variables_imu{4,1}, '\n']);
fprintf([Variables_imu{5,1}, '\n']);
fprintf([Variables_imu{6,1}, '\t', Variables_imu{6,2}, '\t', Variables_imu{
6,3}, '\t', '\n']);
for i=7:n+6
    fprintf([num2str(Variables_imu{i,1}), '\t', Variables_imu{i,2}, '\t',
Variables_imu{i,3}, '\t', '\n']);
end
diary off
```

El archivo Tracker\_imu\_cuat.sto tiene la siguiente configuración en donde se representa las columnas del tiempo y de los sensores IMUs r\_humerus\_imu y r\_ulna\_radius\_hand\_imu. Como se observa en la siguiente figura 17.

```

CA:\Users\willi\Documents\MATLAB\IMUs\Experimento\Tracker_imu_cuat.sto - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Complementos  Pestañas ?
Tracker_imu_cuat.sto
Tracker_imu_cuat.sto
1  DataRate=74.000000
2  DataType=Quaternion
3  version=3
4  OpenSimVersion=4.1
5  endheader
6  time      r_humerus_imu      r_ulna_radius_hand_imu
7  0      0.00061975,0.99951,0.021875,0.022457      -0.00064311,0.99951,0.021741,0.022435
8  0.013497      0.00033799,0.999,-0.031228,-0.032105      0.00078882,0.999,-0.031064,-0.03207
9  0.026994      0.025079,0.99617,-0.01214,-0.011064      0.029017,0.9962,-0.011794,-0.011069
10 0.040491      0.0574,0.99488,0.0066872,0.0097224      0.057325,0.99491,0.0072165,0.0096799
11 0.053988      0.083933,0.99398,0.023117,0.030129      0.083745,0.99397,0.02304,0.030054
12 0.067485      0.11441,0.99834,0.043057,0.050035      0.1143,0.99835,0.043955,0.049935
13 0.080982      0.14313,0.99834,0.06036,0.069327      0.14302,0.99835,0.061441,0.069209
14 0.094479      0.17203,0.99465,0.076923,0.087852      0.17189,0.99467,0.078187,0.087767
15 0.10798      0.20108,0.95353,0.09264,0.10563      0.20093,0.95328,0.094085,0.10551
16 0.12147      0.23029,0.91258,0.10741,0.12244      0.23012,0.91207,0.10904,0.12234
17 0.13497      0.25964,0.8719,0.12115,0.13824      0.25944,0.8711,0.12295,0.13817
18 0.14847      0.28908,0.83155,0.13376,0.15296      0.28887,0.83048,0.13576,0.15294
19 0.16196      0.31859,0.79163,0.14518,0.16651      0.31835,0.79026,0.14737,0.16637
20 0.17546      0.34811,0.75222,0.15536,0.17885      0.34785,0.75055,0.15775,0.17903
21 0.18896      0.37758,0.71399,0.16422,0.18994      0.3773,0.71142,0.16605,0.19028
22 0.20245      0.40696,0.67524,0.17178,0.19975      0.40664,0.67294,0.17462,0.20027
23 0.21595      0.43616,0.63783,0.17795,0.20828      0.43581,0.6352,0.18109,0.209
24 0.22945      0.46511,0.60125,0.18287,0.2156      0.46474,0.59826,0.18613,0.21641
25 0.24295      0.49375,0.56555,0.18623,0.22161      0.49334,0.56221,0.19006,0.22255
26 0.25644      0.52198,0.53092,0.18881,0.22641      0.52155,0.52709,0.19214,0.22728
27 0.26994      0.54975,0.4971,0.18881,0.2296      0.54928,0.49298,0.19422,0.23092
28 0.28344      0.57697,0.46445,0.18706,0.232      0.57647,0.45992,0.19196,0.23277
29 0.29693      0.60357,0.43293,0.18392,0.23177      0.60304,0.42798,0.18875,0.23462
30 0.31043      0.62949,0.40258,0.17959,0.22979      0.62852,0.39719,0.18452,0.23238

```

Figura 17. Archivo Tracker\_imu\_cuat.sto

El modelo de OpenSim tiene dos sensores IMUs en el brazo y en el antebrazo. Con este modelo desarrollado en OpenSim se obtiene la cinemática de los experimentos. Sin embargo, se tiene un inconveniente ya que los sensores en OpenSim no se encuentran orientados adecuadamente al segmento anatómico lo que imposibilita la obtención de graficas coherentes. En la siguiente figura 18 se indica el modelo desarrollado en OpenSim.

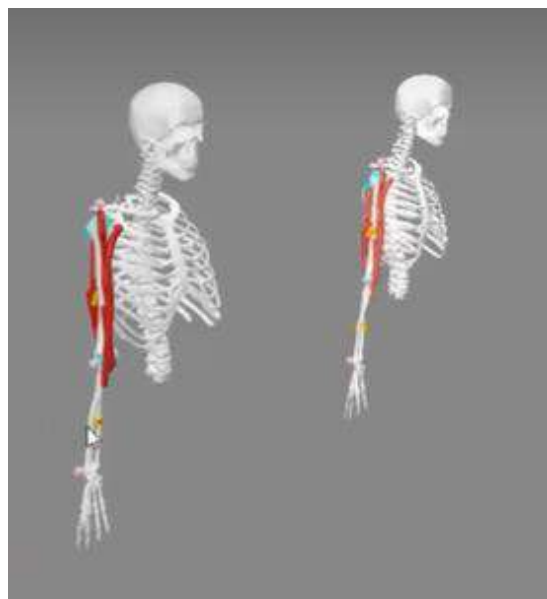
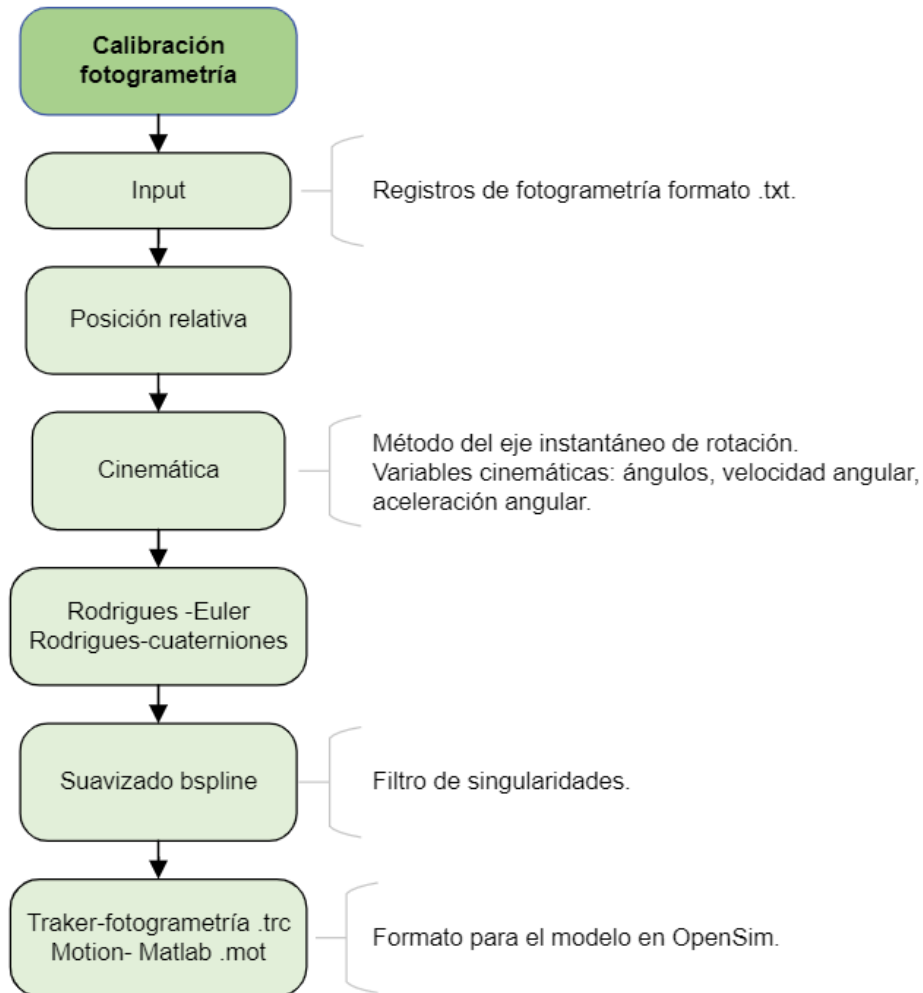


Figura 18. Modelo de IMUs en OpenSim

### 2.3.2.3 Calibración del experimento registro de fotogrametría

El siguiente esquema representa el tratamiento de datos de la calibración en fotogrametría similar al del IMUs. La programación completa se encuentra en el Anexo III.



**Figura 19.** Esquema de programa en Matlab de calibración en fotogrametría

Para determinar la cinemática y vectores de rodrigues de los registros de fotogrametría se utiliza la metodología detallada en el trabajo de titulación del “estudio cinemático de la biomecánica de la pierna con aplicación de fotogrametría” (Trujillo, 2022) de Trujillo.

### 2.3.2.4 Simulación en OpenSim fotogrametría

El modelo replicado en OpenSim tiene 3 marcadores como se indica en la figura 21, en el acromio, codo y muñeca externos a diferencia con el experimento desarrollado con el sistema de fotogrametría que se tiene once marcadores como se indica en la tabla 6. Con ayuda de OpenSim y Matlab se determina la cinemática de ambos modelos y se comparan gráficamente.

Los datos de los registros de fotogrametría son tratados con el programa desarrollado en Matlab detallado en el anexo III y para realizar la simulación en OpenSim es importante

tener archivos con formatos compatibles. De esta manera es necesario almacenar en una variable de cuaterniones los registros de fotogrametría y de movimiento que se utilizará para OpenSim. A continuación, se indica el script de Matlab utilizado para el tratamiento de datos de fotogrametría para transformarlos en formatos admisibles para realizar la simulación en OpenSim.

```

dat_fot_cuat(:,1)=t_fot;
dat_fot_cuat(:, [2:5])=q1;
dat_fot_cuat(:, [6:9])=q2;

data_motion(:,1)=t_fot;
data_motion(:,2)=fi_b;
data_motion(:,3)=fi_ab;

clear dat_fot_euler
dat_fot_euler(:,1)=t_fot;
dat_fot_euler(:, [2 3 4])=180*XYZ_b(:,:)/pi;
dat_fot_euler(:, [5 6 7])=180*XYZ_ab(:,:)/pi;

save resultados.mat dat_fot_euler dat_imu_euler dat_fot_cuat dat_imu_cuat

```

Además, es necesario realizar el suavizado para reducir las irregularidades.

```

fdamat = suavizado_bspline(data_motion(:, [2 3]), t, 200, 6);
fi_fot(:, :, 1)=fdamat(:, :, 1);
Dfi_fot(:, :, 1)=fdamat(:, :, 2);
D2fi_fot(:, :, 1)=fdamat(:, :, 3);

data_motion(:, [2 3])=fi_fot;
dlmwrite('data_motion.txt', data_motion, 'delimiter', ' ')

```

Para cargar los registros de calibración de fotogrametría en OpenSim es necesario utilizar el formato adecuado en .trc. Se indica el formato y el número de marcadores que se definen en OpenSim y están etiquetados como r\_acromion, r\_humerus\_epicondyle, r\_radius\_styloid. Se define el encabezado que se utiliza en OpenSim para facilitar la carga de los registros de fotogrametría.

```

Variables_traker(1,:)={'PathFileType', '4', '(X/Y/Z)', 'Traker_fotogrametria.';
trc', ',', ',', ',', ',', ',', ',', ','};
Variables_traker(2,:)={
'DataRate', 'CameraRate', 'NumFrames', 'NumMarkers', 'Units', 'OrigDataRate', 'OrigDataS;
tartFrame', 'OrigNumFrames', ',', ',', ','};
Variables_traker(3,:)=(num2str(fps), num2str(fps), num2str(n), '3', 'mm', num2str(
fps), '1', num2str(n), ',', ',', ','};
Variables_traker(4,:)={
'Frame#', 'Time', 'r_acromion', ',', ',', 'r_humerus_epicondyle', ',', ',', 'r_radius_styloid'
, ',', ','};
Variables_traker(5,:)={ ',', ',', 'X1', 'Y1', 'Z1', 'X2', 'Y2', 'Z2', 'X3', 'Y3', 'Z3'};

```

El primer marcador cuenta con tres componentes X1, Y1, Z1, las coordenadas son espaciales, el segundo marcador X2, Y2, Z2 y el marcador 3 tiene X3, Y3, Z3. Se reorganiza el registro con los datos de fotogramas, tiempo y componentes de los marcadores. A



continuación, se indica como se almacena la variable de cuaterniones.

```

for i=1:ni
    for j=1:nj
        Variables_traker{i+5,j}=data(i,j);
    end
end

diary Traker_fotogrametria.trc
fprintf([Variables_traker{1,1}, '\t', Variables_traker{1,2}, '\t', Variables_traker{
1,3}, '\t', Variables_traker{1,4}, '\t', '\n']);
fprintf([Variables_traker{2,1}, '\t', Variables_traker{2,2}, '\t', Variables_traker{
2,3}, '\t', Variables_traker{2,4}, '\t', Variables_traker{2,5}, '\t', Variables_traker{
2,6}, '\t', Variables_traker{2,7}, '\t', Variables_traker{2,8}, '\t', '\n']);
fprintf([Variables_traker{3,1}, '\t', Variables_traker{3,2}, '\t', Variables_traker{
3,3}, '\t', Variables_traker{3,4}, '\t', Variables_traker{3,5}, '\t', Variables_traker{
3,6}, '\t', Variables_traker{3,7}, '\t', Variables_traker{3,8}, '\t', '\n']);
fprintf([Variables_traker{4,1}, '\t', Variables_traker{4,2}, '\t', Variables_traker{
4,3}, '\t', Variables_traker{4,4}, '\t', Variables_traker{4,5}, '\t', Variables_traker{
4,6}, '\t', Variables_traker{4,7}, '\t', Variables_traker{4,8}, '\t', Variables_traker{
4,9}, '\t', '\n']);
fprintf([Variables_traker{5,1}, '\t', Variables_traker{5,2}, '\t', Variables_traker{
5,3}, '\t', Variables_traker{5,4}, '\t', Variables_traker{5,5}, '\t', Variables_traker{
5,6}, '\t', Variables_traker{5,7}, '\t', Variables_traker{5,8}, '\t', Variables_traker{
5,9}, '\t', Variables_traker{5,10}, '\t', Variables_traker{5,11}, '\t', '\n']);
for i=6:n+5
    fprintf([num2str(Variables_traker{i,1}), '\t', num2str(Variables_traker{i,
2}), '\t', num2str(Variables_traker{i,3}), '\t', num2str(Variables_traker{i,4}), '\t',
num2str(Variables_traker{i,5}), '\t', num2str(Variables_traker{i,6}), '\t', num2str
(Variables_traker{i,7}), '\t', num2str(Variables_traker{i,8}), '\t', num2str
(Variables_traker{i,9}), '\t', num2str(Variables_traker{i,10}), '\t', num2str
(Variables_traker{i,11}), '\t', '\n']);
end
diary off

```

A continuación, se indica como se almacena en una variable de movimiento para la generación de los archivos. Se compara la cinemática calculada en Matlab con la generada en OpenSim.

```

[ni nj]=size(data_motion);

Variables_motion(1,:)={'Motion_fot_matlab', '', ''};
Variables_motion(2,:)={'version=1', '', ''};
Variables_motion(3,:)={'nRows=', num2str(n), '', ''};
Variables_motion(4,:)={'nColumns=3', '', ''};
Variables_motion(5,:)={'inDegrees=yes', '', ''};
Variables_motion(6,:)={' ', '', ''};
Variables_motion(7,:)={'Units are S.I. units (second, meters, Newtons,
...)', '', ''};
Variables_motion(8,:)={'Angles are in degrees.', '', ''};
Variables_motion(9,:)={' ', '', ''};
Variables_motion(10,:)={'endheader', '', ''};
Variables_motion(11,:)={'time', 'r_shoulder_elev', 'r_elbow_flex'};

```

```

for i=1:ni
    for j=1:nj
        Variables_motion{i+11,j}=data_motion(i,j);
    end
end

diary Motion_fot_Matlab.mot
fprintf([Variables_motion{1,1}, '\t', '\n']);
fprintf([Variables_motion{2,1}, '\t', '\n']);
fprintf([Variables_motion{3,1}, '\t', '\n']);
fprintf([Variables_motion{4,1}, '\t', '\n']);
fprintf([Variables_motion{5,1}, '\t', '\n']);
fprintf([Variables_motion{6,1}, '\t', '\n']);
fprintf([Variables_motion{7,1}, '\t', '\n']);
fprintf([Variables_motion{8,1}, '\t', '\n']);
fprintf([Variables_motion{9,1}, '\t', '\n']);
fprintf([Variables_motion{10,1}, '\t', '\n']);
fprintf([Variables_motion{11,1}, '\t', Variables_motion{11,2}, '\t', Variables_motion{
11,3}, '\t', '\n']);
for i=12:n+11
    fprintf([num2str(Variables_motion{i,1}), '\t', num2str(Variables_motion{i,
2}), '\t', num2str(Variables_motion{i,3}), '\t', '\n']);
end
diary off

```

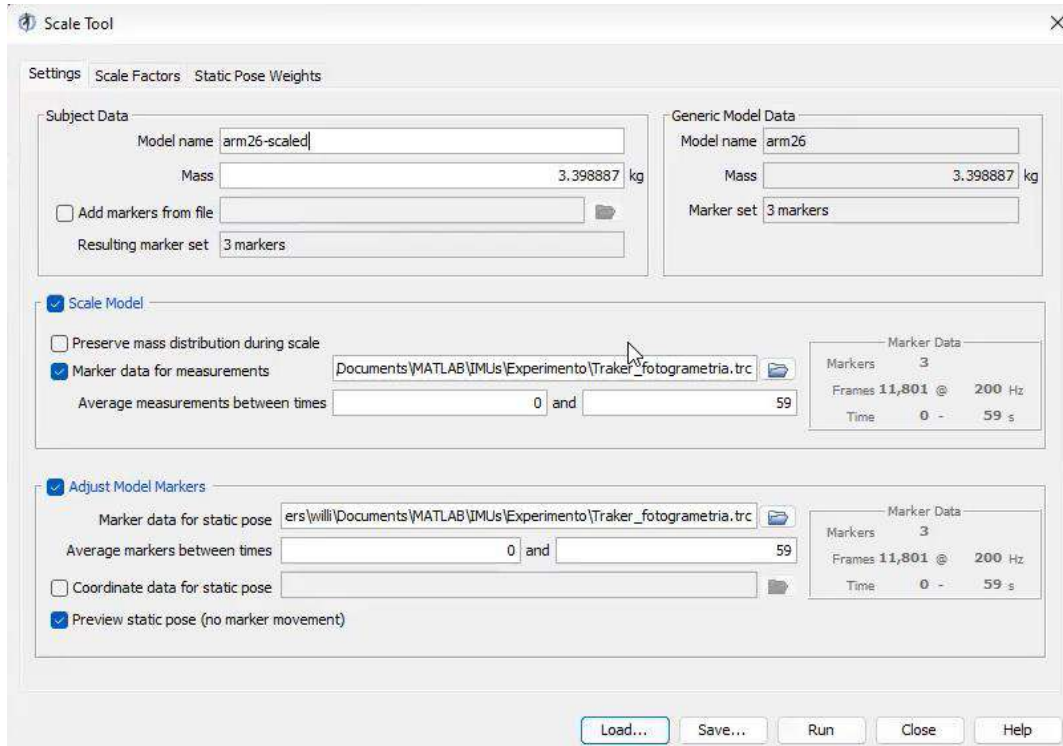
Con esta parte del programa se generan los archivos para la simulación en OpenSim.

```

filename = ['data.xlsx'];
xlswrite(filename, Variables_motion, 'motion', ['A1:', Tabla_exel(round((ncol-fix
(ncol)) * numel(Tabla_exel))), num2str(ni)]);

```

Se reorganiza y se genera el archivo para cargar en OpenSim en formato .trc y .mot. En OpenSim es necesario realizar el proceso de escalado y se cargan los registros generados de fotogrametría.



**Figura 20.** Ventana de OpenSim para escalar los registros de la experimentación

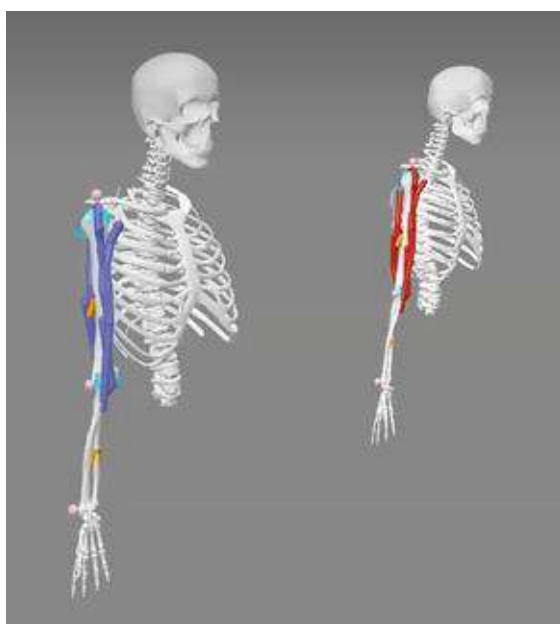
Se determina la cinemática con dos formas diferentes. El modelo que se trabaja en Matlab requiere de más marcadores para representar un sólido rígido (mínimo 3), que a diferencia en OpenSim se puede trabajar con modelos de cinemática rígida simple. El modelo se lo puede reajustar de acuerdo con dimensiones reales de experimentación.



**Figura 21.** Modelo de fotogrametría con OpenSim

El modelo se lo puede reajustar de acuerdo con dimensiones reales de la experimentación como se muestra en la figura 22. Al ejecutar el archivo en formato .trc se genera un nuevo archivo que se lo nombra Motion\_OpenSim\_traker.mot y se vuelve a ejecutar.





**Figura 22.** Modelo escalado en OpenSim

### 2.3.2.5 Validación de fotogrametría e IMUs

Para la validación de las técnicas de captura de movimiento de fotogrametría e IMUs se utiliza un script desarrollado en Matlab en donde se determina la posición angular del brazo y antebrazo. A continuación, se indica el script utilizado.

Se requiere cargar los datos de los registros de tiempo y los ángulos obtenidos de fotogrametría e IMUs. Además, se utiliza la función `desfase_H` en donde se determina puntos en donde su valor sea cero o sea inferior al umbral y se los eliminan para obtener datos similares en ambos registros detalles de la programación se indican en el anexo IV.

```
clear;clc;
load('resultados.mat')
t_imu=dat_imu_euler(:,1);
t_fot=dat_fot_euler(:,1);
fi_imu=dat_imu_euler(:,2:7);
Fi_fot=dat_fot_euler(:,2:7);
pos_desfase_imu=desfase_H(fi_imu(:,2),t_imu);
pos_desfase_fot=desfase_H(Fi_fot(:,2),t_fot);
```

Se determina el desfase horizontal de los datos de IMUs y fotogrametría.

```
%Desfase horizontal
t1=t_imu-t_imu(pos_desfase_imu);
t2=t_fot-t_fot(pos_desfase_fot);

t_imus=t1-min(t2);
t_fot=t2-min(t2);

if t_imus(1)<0
posneg=find(t_imus<0);
```

```

t_imus(posneg, :, :) = [];
fi_imu(posneg, :, :) = [];
Fi_imu=fi_imu;
t_imu=t_imus-t_imus(1);
else
dt_imus=t_imus(2)-t_imus(1);
tr(:,1)=0:dt_imus:t_imus(1);
n=max(size(tr));
t_imu=[tr;t_imus];
fi_offset=offset_imu.*ones(n,3,2);
Fi_imu=[fi_offset;fi_imu];
end

```

Se realiza el suavizado de los datos.

```

fdamat = suavizado_bspline(Fi_fot, t_fot, 200, 4,t_imu);
fi_fot(:, :, 1)=fdamat(:, :, 1);
Dfi_fot(:, :, 1)=fdamat(:, :, 2);
D2fi_fot(:, :, 1)=fdamat(:, :, 3);

```

Se determina los datos necesarios para determinar la posición angular del brazo y se realiza el gráfico editando la configuración desde el nombramiento de ejes, título, leyendas, estilo de curvas y el tipo de letra.

```

%Validar
Xc(:,1,1)=Fi_imu(:,2,1);
Xc(:,1,2)=fi_fot(:,2);

[Max Min Rango Media Mediana SEM DesvStan Varianza Sig_F F_test Sig_Q Q_test
Percentil IQR fxi xi Icc CMC CMCm cmc Pearson]=ICorre_sesiones(2, 'single',Xc);
disp(['Componente principal CMC = ', num2str(sqrt(abs(cmc)))])

%Brazo
figure(1)
hold on
plot(t_imu,fi_fot(:,2), 'Color',[0.6 0.6 0.6], 'LineWidth',2)
plot(t_imu,Fi_imu(:,2), 'k-.', 'LineWidth',2)
ax = gca;
ax.FontName='Times New Roman';
ax.FontSize=16;
ax.GridLineStyle = '-';
ax.GridColor = [0 0 0];
grid on;box on;
xx = ax.XLim;
yy = ax.YLim;
xlabel('Tiempo ( s )', 'fontsize', 18);
ylabel('Posición angular componente principal ( ° )', 'fontsize', 18);
legend('Fotogrametria', 'IMUs', 'Location', 'northeast', 'NumColumns', 1, 'fontsize', 14, 'FontName', 'Times New Roman', 'Position', [0.40 0.7 0.1 0.2]);
legend('boxoff')
text(40,yy(1)+130, ['CMC ', num2str(sqrt(abs(cmc)))], 'fontsize', 12)
print('-djpeg', '-r500', '-painters', ['Validar_brazo', '.jpg']);

```

De la misma forma con el antebrazo.

```

%Antebrazo
figure(2)
hold on
plot(t_imu,fi_fot(:,5),'Color',[0.6 0.6 0.6],'LineWidth',2)
plot(t_imu,Fi_imu(:,5),'k-','LineWidth',2)
ax = gca;
ax.FontName='Times New Roman';
ax.FontSize=16;
ax.GridLineStyle = '-';
ax.GridColor = [0 0 0];
grid on;box on;
xx = ax.XLim;
yy = ax.YLim;
xlabel('Tiempo ( s )','fontsize', 18);
ylabel('Posición angular componente principal ( ° )','fontsize', 18);
legend('Fotogrametria','IMUs','Location','northeast','NumColumns',1,'fontsize', 14, 'FontName','Times New Roman','Position',[0.40 0.75 0.1 0.2]);
legend('boxoff')
text(40,yy(1)+115,['CMC ',num2str(sqrt(abs(cmc)))],'fontsize', 12)
print('-djpeg','-r500','-painters', ['Validar_antebrazo','.jpg']);

```

Finalmente se obtiene curvas comparativas de la posición angular del brazo y antebrazo de los registros del experimento realizado en IMUs y fotogrametría.

### 3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

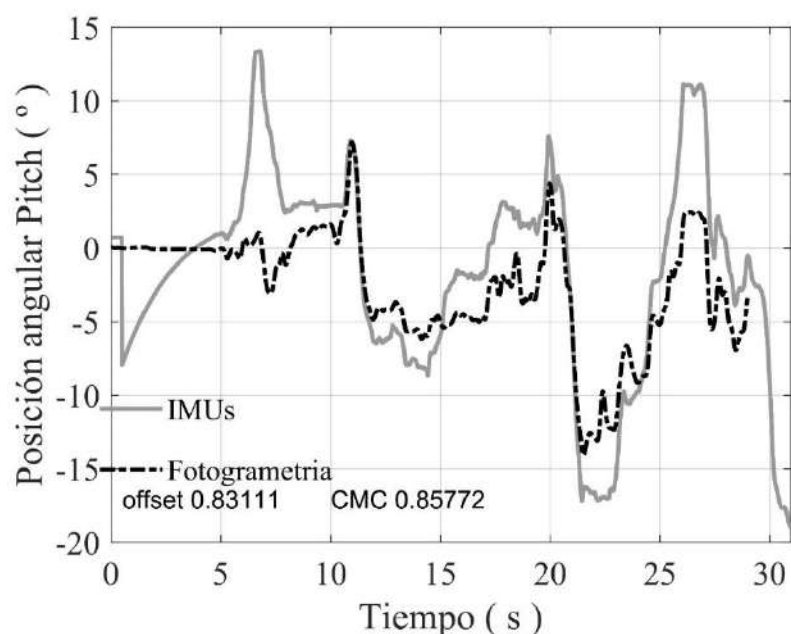
#### 3.1 Análisis de resultados

A continuación, se muestran el análisis de resultados obtenidos de la calibración y validación de la experimentación del modelo biomecánico del movimiento de flexo extensión del codo.

##### 3.1.1 Resultados de la calibración

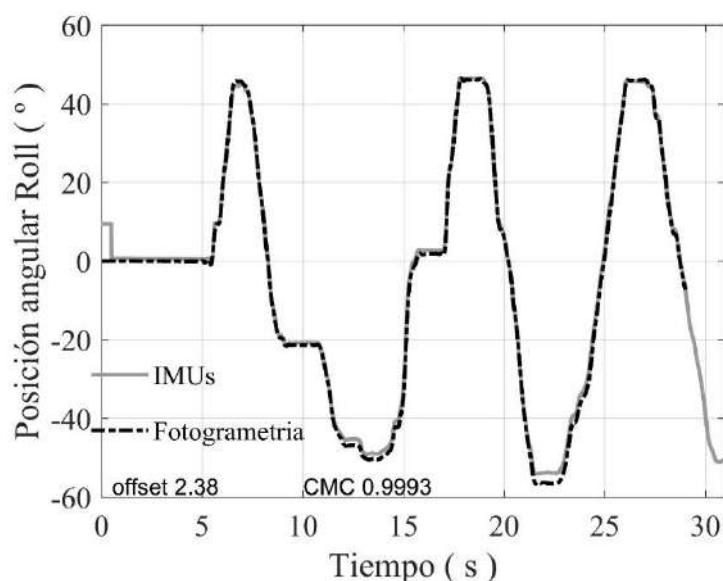
###### 3.1.1.1 Calibración en X

A continuación, se indica las curvas de los ángulos de Euler de los registros IMUs y fotogrametría en el eje X.



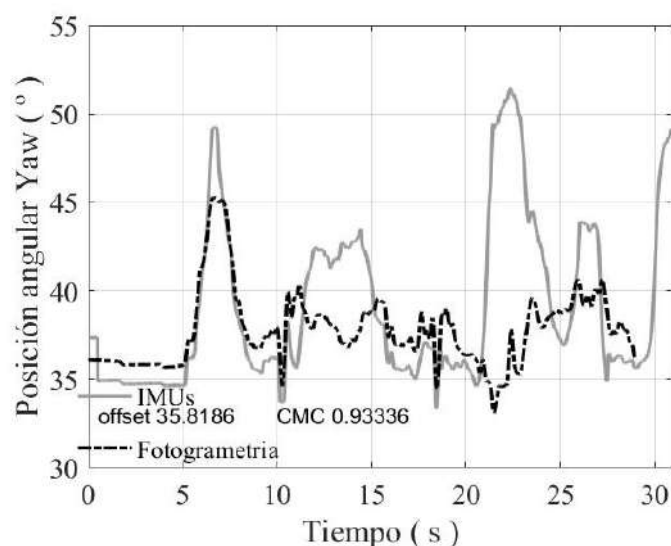
**Figura 23.** Posición angular Pitch calibración en X.

Se observa en la figura 23 que la posición angular de los registros de fotogrametría e IMUs son semejantes sin embargo existen varias irregularidades como los picos máximos además de un valor de offset de 0.83111, es decir el desplazamiento entre las curvas. Por lo que no se considera esta posición angular pitch en el eje x.



**Figura 24.** Posición angular Roll calibración en X.

Se observa que en la posición angular Roll las curvas son más semejantes tanto en los registros de IMUs y de fotogrametría por lo que se considera esta posición angular aceptable para el eje x. Aunque el offset es de 2.38 al inicio de la experimentación se debe al tiempo al tiempo de respuesta de cada sensor. El CMC (Comparative Measurement Close-range) determinado es de 0.99 que indica tan precisos y exactos son los resultados obtenidos por las técnicas de captura de movimiento el sistema de fotogrametría y sensores IMUs.



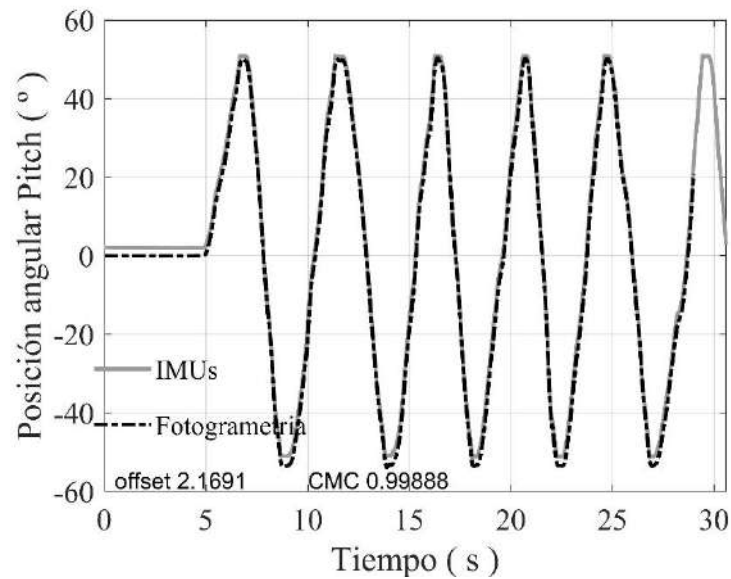
**Figura 25.** Posición angular Yaw calibración en X.

En el caso de la calibración en el eje X se observa que en la figura 24 se representan las curvas de la posición angular roll de los registros IMUs y fotogrametría de forma semejante, lo que relaciona el ángulo de Euler roll con el eje de rotación x según la teoría. Además, el offset considerado es 2.38 relativamente bajo al comparar las curvas.

Un offset demasiado grande indicaría gran desfase entre las curvas y un CMC bajo indicaría imprecisión e inexactitud.

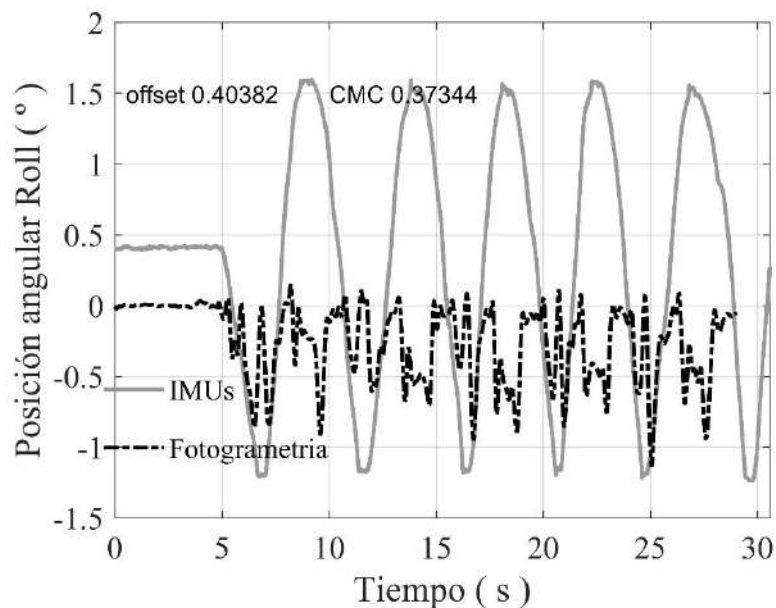
### 3.1.1.2 Calibración Y

A continuación, se indica las curvas de los ángulos de Euler de los registros IMUs y fotogrametría de calibración en el eje Y.

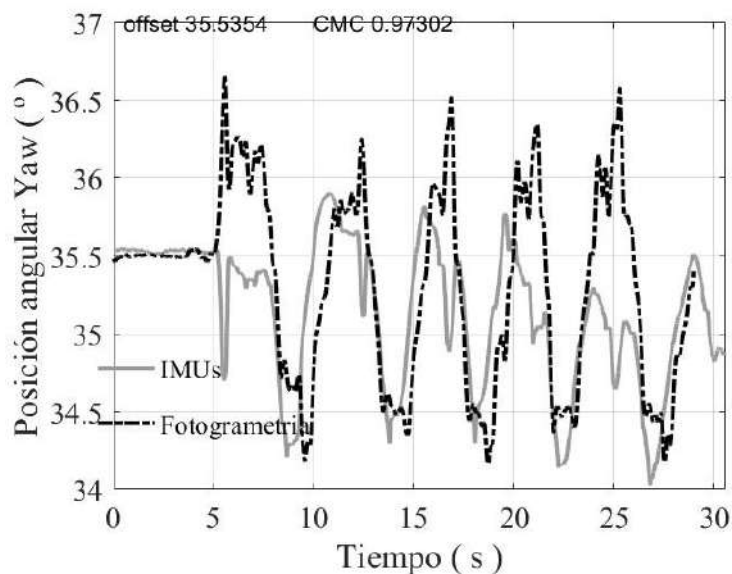


**Figura 26.** Posición angular Pitch calibración en Y.

Se observa que la posición angular Pitch de la figura 26 tanto en el sensor IMUs y de fotogrametría son semejantes. El offset es de 2.16 lo que indicaría la precisión con la que las curvas de posición angular obtenida en fotogrametría e IMUs se superponen. El CMC determinado es de 0.9988.



**Figura 27.** Posición angular Roll calibración en Y.

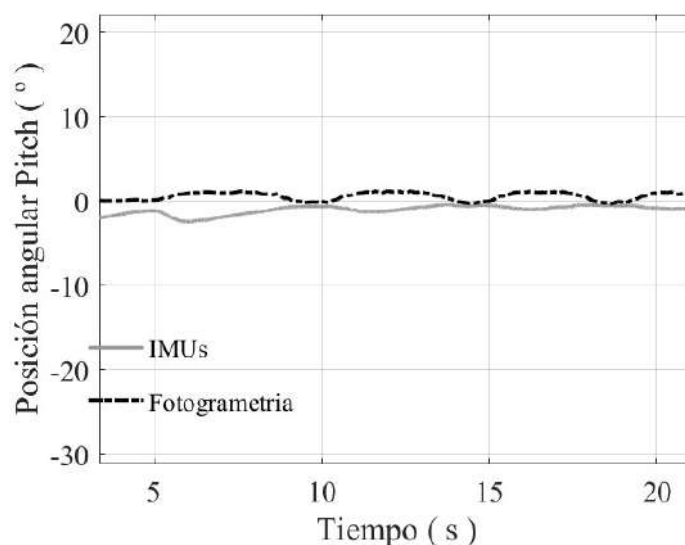


**Figura 28.** Posición angular Yaw calibración en Y.

Tanto la posición angular roll y yaw en los registros de calibración en el eje Y no coinciden lo que indicaría que no son aceptables y se considera la posición angular pitch que se encuentra en rotación en el eje Y. Aceptable con un CMC de 0.999.

### 3.1.1.3 Calibración Z

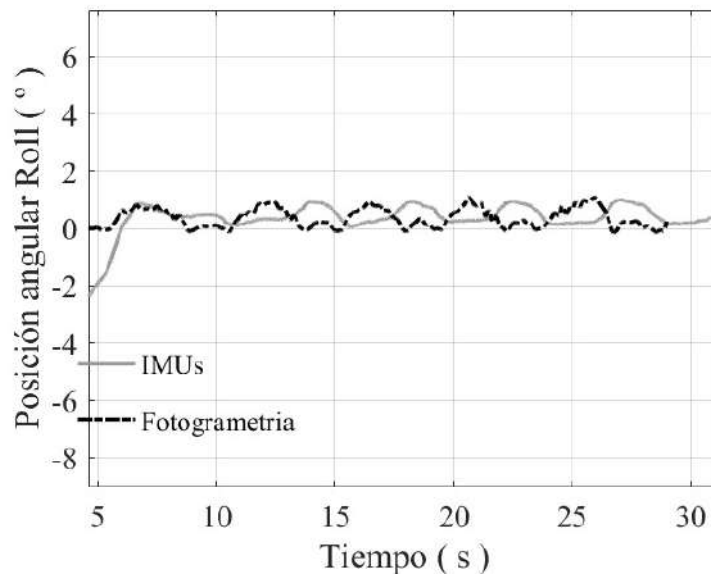
A continuación, se indica las curvas de los ángulos de Euler de los registros IMUs y fotogrametría de calibración en el eje Z.



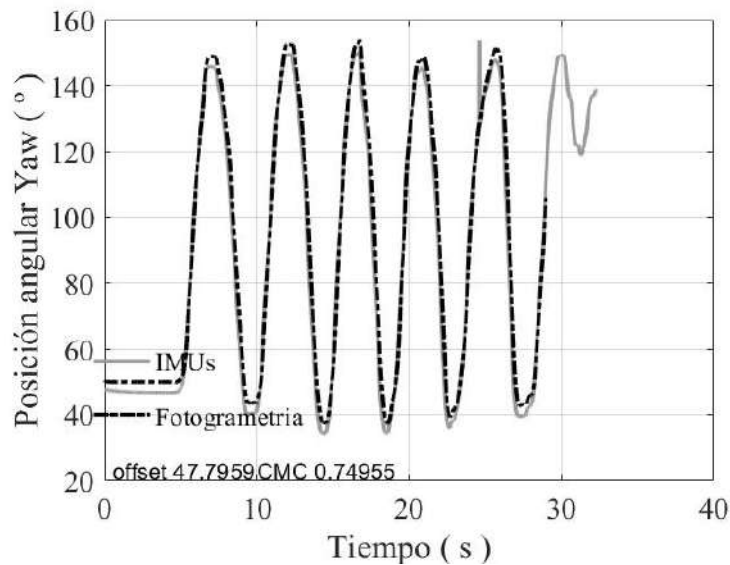
**Figura 29.** Posición angular Pitch calibración en Z.

Como se observa en la figura 29 se tiene las curvas de la posición angular pitch en el eje de rotación Z de los registros de IMUs y fotogrametría, las cuales no coinciden. Y no se

consideran.



**Figura 30.** Posición angular Roll calibración en Z.



**Figura 31.** Posición angular Yaw calibración en Z.

Las curvas de la posición angular Yaw en el registro IMUs y en el de fotogrametría coinciden en la calibración en el eje de rotación Z como se indica en la figura 31. Sin embargo, en este caso se observa que el offset es 47.7959 y el CMC 0.79955 lo que indicaría imprecisión e inexactitud.

Se conoce que al usar la metodología Euler para describir la orientación de un objeto es necesario usar los ángulos de Euler, para especificar como ha ocurrido la rotación alrededor de los ejes principales que en este caso son los ejes X, Y, Z. Los tres ángulos de Euler son el de cabeceo, alabeo y guiñada (roll, pitch, yaw respectivamente). Por esta razón al observar las figuras 24, 26, 31 se nota que las curvas son semejantes en ambos registros el de IMUs



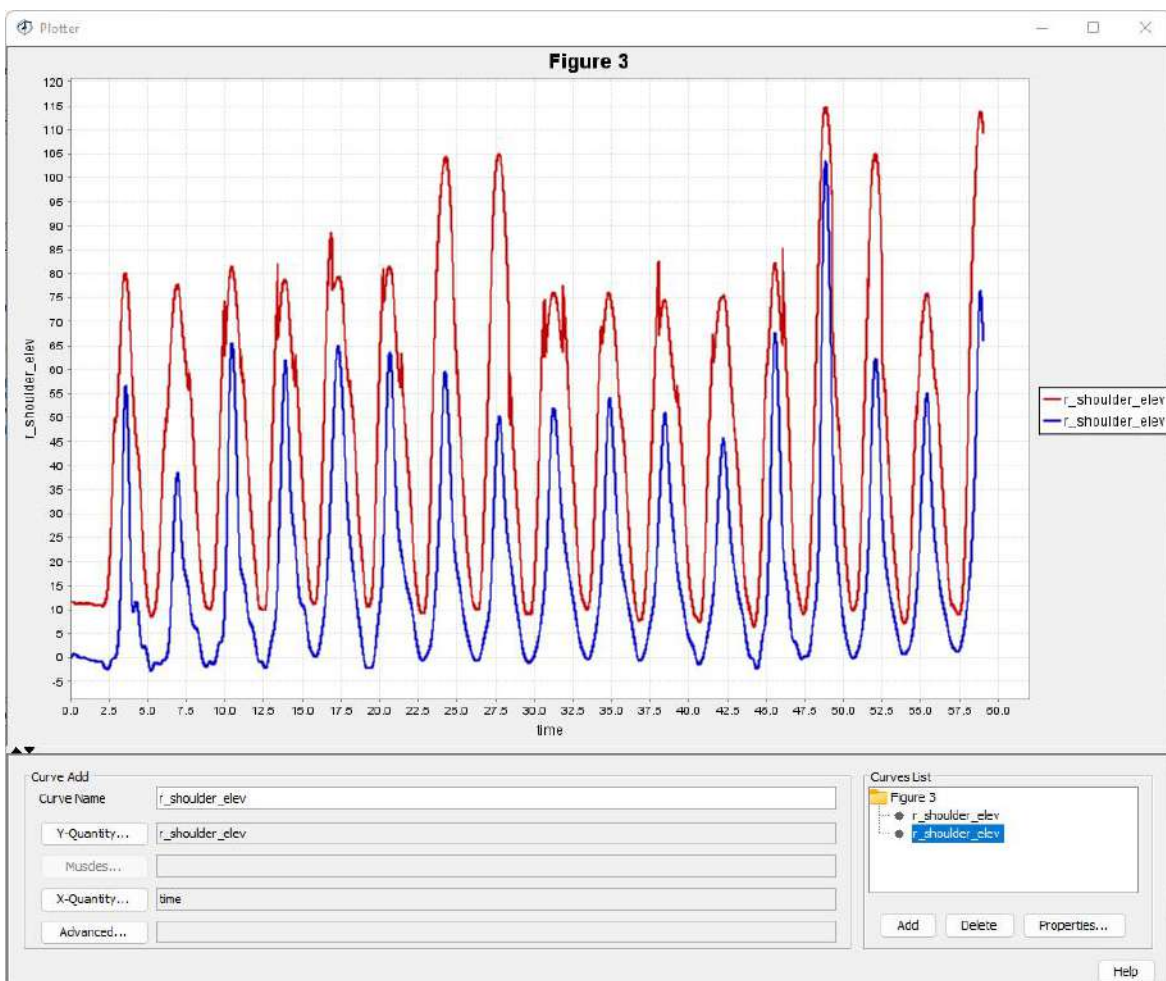
y fotogrametría lo que confirmaría el ángulo de rotación alrededor de cada eje, además, que los valores de offset y CMC son aceptables excluyendo los resultados de la figura 31. A continuación, en la siguiente tabla 8 se detallan los ángulos de Euler y el respectivo eje de rotación.

**Tabla 8.** Ángulos de Euler y ejes de rotación

Ángulos	Ejes de rotación
Roll	Eje X
Pitch	Eje Y
Yaw	Eje Z

### 3.1.1 Resultados de la experimentación

Los resultados de la simulación realizada en OpenSim del modelo con fotogrametría se indica en la siguiente figura 32. Para obtener la gráfica se carga el archivo de cinemática del brazo con respecto al tiempo generado en Matlab detalles de programación revisar el Anexo 3, juntamente con el archivo generado de la cinemática en OpenSim. El procedimiento se encuentra detallado en la sección 2.3.2.4 de la metodología.



**Figura 32.** Cinemática del brazo según fotogrametría y OpenSim (rojo=fotogrametría, azul=OpenSim)

Como se observa en la figura 32 que al comparar las curvas de la cinemática de los registros de fotogrametría con la curva del modelo de OpenSim se tiene una tendencia similar en ambas curvas sin embargo se debe considerar que el desfase que existe entre las curvas es debido a diferentes sistemas de referencia que se tiene tanto en el programa de Matlab, el sistema de fotogrametría y el OpenSim. Además, el modelo considerado en OpenSim tenía solamente 3 marcadores por lo que se recomendaría para futuras investigaciones la implementación de más marcadores en puntos anatómicos relevantes para la simulación.

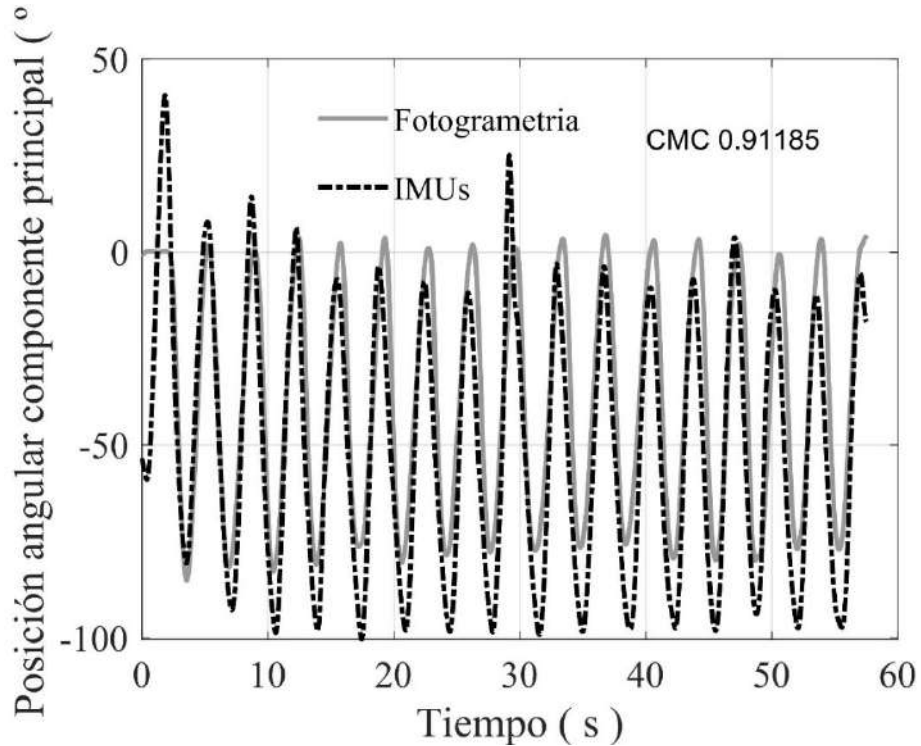
En el caso de los IMUs existen varios inconvenientes al momento de trabajar en OpenSim. Con los sensores IMUs se tiene directamente los ángulos de Euler y es necesario transformar en cuaterniones para cargar en OpenSim lo que durante ese proceso se incrementan las irregularidades que son notables al momento de realizar la simulación. Y si se utiliza el filtro de singularidades detallado en el Anexo I, existen errores de aproximaciones por que los IMUs en el OpenSim no se encuentran ubicados direccionalmente en los miembros anatómicos.

### **3.1.2 Resultados de la validación**

Para la validación de las técnicas de captura de movimiento de los sensores IMUs con fotogrametría se realiza un script en Matlab detallado en la sección 2.3.2.5 de la metodología en donde se cargan los archivos de los registros generados de la experimentación y se obtienen gráficas comparativas de las posiciones angulares del componente principal del brazo y antebrazo.

Para realizar la validación solo se considera el plano principal (ángulo axial) y no los planos secundarios, debido a que cada modelo considerado tiene su propio sistema de referencia por ejemplo en el laboratorio de fotogrametría se tiene un sistema referencia, los sensores IMUs cuentan con su propio sistema de referencia y en el caso de la simulación el OpenSim tiene otro.

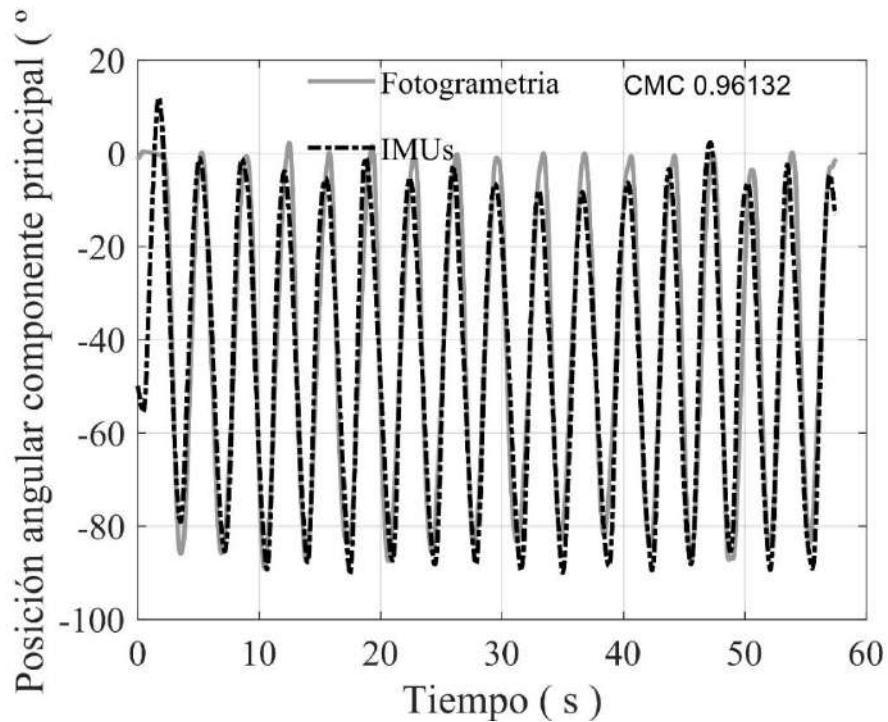
La siguiente figura 33 indica la posición angular del componente principal del brazo de los registros de fotogrametría e IMUs del experimento.



**Figura 33.** Posición angular componente principal brazo.

En la figura 33 se observa que las curvas de posiciones angulares en el componente principal obtenidas de los registros de fotogrametría e IMUs del experimento realizado son semejantes para el caso del brazo. Con un valor CMC de 0.91185 que es un valor que se encuentra dentro del parámetro aceptable ya que indica una precisión y exactitud relativamente fiable. Aunque como se observa existe el desfase entre las curvas notable en la parte inferior.

A continuación, se indica la posición angular del antebrazo de modelo cinemático en donde se considera solamente el componente principal.



**Figura 34.** Posición angular en el componente principal en el antebrazo

Como se observa en la figura 34 las curvas de posiciones angulares del antebrazo presentan mayor semejanza que el caso del brazo y el desfase es más pequeño. Ambas de fotogrametría e IMUs curvas cumplen con la misma tendencia y considerando el valor de CMC se tiene el valor de 0.96132 el cual se encuentra dentro de parámetros aceptables de exactitud y precisión.

Por lo que indicaría que en el caso de casos sencillos de estudio se puede utilizar IMUs como técnica de captura de movimiento alternativa a la de fotogrametría, sin embargo, en el caso que se requiera realizar casos de estudio de gran precisión considerar las técnicas de captura de movimiento que cumpla con los requerimientos necesarios.

## 3.2 Conclusiones

En resumen, los beneficios que brinda la utilización de los sensores IMUs con respecto a otros métodos para capturar el movimiento son altos. Ya que los sensores IMUs son portables y de fácil movilidad, haciéndolos prácticos para realizar estudios en diferentes áreas sin la necesidad de un espacio físico equipado de cámaras. Además, como se ha demostrado en este trabajo de investigación los resultados obtenidos son similares a los de fotogrametría.

La metodología utilizada para determinar la cinemática de los registros obtenidos de fotogrametría e IMUs es la de Euler. Además, es importante aclarar que en el caso de los sensores IMUs estos valores son obtenidos directamente al momento de realizar la experimentación y no requieren de tratamiento adicional. Sin embargo, para realizar la simulación en el software OpenSim se requiere transformar a cuaterniones como se indica en la sección 2.3.2.2 y 2.3.2.4 de la metodología.

El modelo biomecánico del codo desarrollado para determinar la viabilidad de las técnicas de captura de movimiento tanto el sistema de fotogrametría y los sensores IMUs, permitió obtener resultados de las posiciones angulares en el componente principal del brazo y el antebrazo, como se puede observar gráficamente en la figura 33 y 34 respectivamente, se tienen tendencias similares, además valores CMC de 0.91 en el caso del brazo y 0.96 en el antebrazo. Lo que confirma que es posible utilizar los sensores IMUs como técnica de captura de movimiento en áreas de estudio en donde se analicen modelos simples.

Para determinar la viabilidad de las técnicas de captura de movimiento se desarrolló un programa en Matlab detallado en la sección 2.3.2.5 de la metodología, en donde se comparan las posiciones angulares gráficamente del componente principal del brazo y el antebrazo del modelo biomecánico obtenidas de los registros de fotogrametría e IMUs.

Como resultado de la simulación realizada en OpenSim se observa la figura 32 en donde se comparan la curva de la cinemática del brazo de los registros de fotogrametría con la curva obtenida del modelo en OpenSim detallado en la sección 2.3.2.3 de la metodología. La figura 32 indica la misma tendencia en ambas curvas en donde se valida el sistema de fotogrametría como técnica de captura de movimiento para sistemas complejos sin considerar el desfase que existe. El desfase se produce debido a los diferentes sistemas de referencia que se tiene en fotogrametría y en OpenSim.

En conclusión, los sensores IMUs pueden ser utilizados como técnicas de captura de

movimiento en casos de estudio en donde el modelo biomecánico sea simple. Y es posible combinar sistemas de fotogrametría con sensores IMUs, para obtener mayor precisión y exactitud al momento de realizar la experimentación debido a sus diferentes enfoques. Finalmente, la elección de utilizar cualquier técnica de captura de movimiento dependerá de las necesidades de aplicación y recursos financieros.

### **3.3 Recomendaciones**

Se recomienda que para poder realizar futuras réplicas de la experimentación de modelos biomecánicos es fundamental registrar la ubicación y orientación de marcadores anatómicos y sensores IMUs adecuadamente, para poder comparar los resultados experimentales.

Para la simulación del modelo en OpenSim se recomienda utilizar más de tres marcadores, para lograr obtener mayor precisión en el movimiento. Se debe considerar las diferencias de las irregularidades del movimiento del cuerpo humano con una simulación ideal. En el caso de la simulación con IMUs se deben orientar adecuadamente al miembro anatómico. Para facilitar la simulación en OpenSim se recomienda archivos en formato .trc.

Se recomienda que antes de realizar la experimentación del modelo biomecánico, es necesario realizar la calibración de los equipos para evitar incongruencias en los resultados. Además, los sensores utilizados en el sistema de fotogrametría deben estar ubicados en zonas visibles para que las cámaras los capturen.

El sujeto que realice la experimentación debe cumplir con el protocolo de pruebas y seguir las especificaciones para la ubicación de marcadores y sensores IMUs. Es importante que el sujeto se encuentre sano y no sufra ninguna patología.

Se propone para trabajos de investigación futuros analizar modelos biomecánicos más complejos comparando el sistema de fotogrametría y sensores IMUs. E incluso se recomienda utilizar EMG como forma de validación adicional.

## 4 REFERENCIAS BIBLIOGRÁFICAS

- Ainsworth, B. (s/f). *Estilos de vida saludable*. Recuperado el 19 de julio de 2023, de <https://estilosdevidasaludable.sanidad.gob.es/actividadFisica/actividad/queEs/home.htm>
- Algoritmo de estimación e implementación Software*. (s/f). Universidad de Sevilla.
- Castellanos, J., Montealegre, L., & Martínez, B. (2020). Uso de sensores inerciales en fisioterapia: Una aproximación a procesos de evaluación del movimiento humano. *Universidad y salud*, 55–63.
- Chaler, J., Garreta, R., & Müller, B. (2005). Técnicas instrumentales de diagnóstico y evaluación en rehabilitación: estudio de la marcha instrumental. *ScienceDirect*, 39(6), 305–314. [https://doi.org/10.1016/S0048-7120\(05\)74364-4](https://doi.org/10.1016/S0048-7120(05)74364-4)
- Cuadros, S. (2020). *Estudio de viabilidad de sensores inerciales (IMUs) para el análisis de Timed Up & Go test*. Universitat Politècnica de Catalunya.
- Descripción de los modelos implementados*. (2020). Generalitat valenciana.
- Ferrer, D. (2015). Adquisición de datos IMU con un sistema embebido. En 2015.
- Freire, B. (2019). *Evaluación de datos experimentales en un modelo OpenSim del comportamiento del movimiento de flexo extensión del cuello*.
- Garofalo, P. (s/f). *Development of motion analysis protocols based on inertial sensors*.
- Gomeñuka, N., & Cabral, M. (2008, septiembre). *Las capacidades coordinativas en los alumnos del profesorado de Educación Física*. <https://www.efdeportes.com/efd124/las-capacidades-coordinativas-en-los-alumnos-del-profesorado-de-educacion-fisica.htm>
- González, I. (2016). *Movimientos de la extremidad superior y terapias de intervención en daño cerebral adquirido*. Universidad Salamanca.
- Guzmán, E., & Méndez, G. (2018). Electromiografía en las Ciencias de la Rehabilitación. *Salud Uninorte*, 34.
- Instituto Biomecánico de Valencia. (2020). *Proyecto cofinanciado por los fondos FEDER, dentro del programa operativo FEDER de la comunitat Valenciana*. Generalitat valenciana.
- Instituto de biomecánica de Valencia. (s/f). *Kinescan IBV V2014*. Recuperado el 20 de julio de 2023, de <https:// analisisbiomecanico.ibv.org/productos/tecnicas-de-registro/kinescan-ibv.html>
- Laguna, M. (2023, mayo 30). *Cintura escapular*. <https://www.kenhub.com/es/library/anatomia-es/cintura-escapular-es>
- Lobato, V. (2016). *Modelo flexible de movimiento de torso, brazo, antebrazo y muñeca*. Instituto Nacional de Astrofísica, Óptica y Electrónica.
- Martínez, J. (s/f). *Ciencias de la actividad física y el deporte*. Universidad de Sevilla.
- Martínez, R., & Romero, M. (2013). *Uso de sensores inerciales en la medición y evaluación de movimiento humano para aplicaciones en la salud*.
- Mayo clinic. (2022, abril 20). *Fractura de pie*. <https://www.mayoclinic.org/es/diseases-conditions/broken-foot/symptoms-causes/syc-20355492>

- McGinnis, R. S. (2013). *Advancing Applications of IMUs in Sports Training and Biomechanics*. *MOCAP, como funciona la captura de movimiento y que aplicaciones tiene para la industria*. (s/f). Recuperado el 20 de julio de 2023, de <https://trigital.es/mocap-como-funciona-la-captura-de-movimiento-y-que-aplicaciones-tiene-para-la-industria/>
- Navarro, B. (2023, julio 5). *Anatomía miembro superior clavícula*. <https://www.kenhub.com/es/library/anatomia-es/clavicula-anatomia>
- Organización Mundial de la Salud. (2022, octubre 5). *Actividad física*. <https://www.who.int/es/news-room/fact-sheets/detail/physical-activity>
- Pozo, R., Gómez, A., Medrano, J., Curay, P., & Abalco, D. (2022). Análisis biocinemático de la marcha en pacientes con hemiplejía. *Scielo*, 17(3).
- Ramírez, N. (2011). *Caracterización cinemática del tren superior en el saque del voleibol*.
- Samatas, G., & Pachidis, T. (2022). Inertial Measurement Units (IMUs) in Mobile Robots over the Last Five Years: A Review. En *Designs* (Vol. 6, Número 1). MDPI. <https://doi.org/10.3390/designs6010017>
- Trujillo, W. (2022). *Estudio cinemático de la biomecánica de la pierna con aplicación de fotogrametría*. Escuela Politécnica Nacional.
- Wu, G., Van der Helm, F., Veeger, D., Makhsous, M., Van Roy, P., Anglin, C., Nagels, J., Karduna, A., McQuade, K., Wang, X., Werner, F., & Buchholz, B. (2005). ISB recommendation on definitions of joint coordinate systems of various joints for the reporting of human joint motion -Part II: shoulder, elbow, wrist and hand. *ELSEVIER*, 38(5), 981–992.



## **5 ANEXOS**

ANEXO I. Filtro de singularidades

ANEXO II. Calibración IMUs

ANEXO III. Calibración fotogrametría

ANEXO IV. Función desfase\_H

# ANEXO I

## Función Filtro\_singularidades

```
function [imu_s]=Filtro_singularidades(imu_s,sing)
%imu_s=imu_p;
[ni nj]=size(imu_s);
switch sing
    case 'MAX'
        for j=1:100
            for i=1:nj
                clear pos_imus
                pos_imus=find(imu_s(:,i)==max(imu_s(:,i)));
                ns=numel(pos_imus);
                for ks=1:ns
                    if pos_imus(ks)~=ni && pos_imus(ks)~=1
                        imu_s(pos_imus(ks),i)=(imu_s(pos_imus(ks)-1,i)
+imu_s(pos_imus(ks)+1,i))/2;
                    elseif pos_imus(ks)==ni
                        imu_s(pos_imus(ks),i)=(imu_s(pos_imus(ks)-1,
i));
                    elseif pos_imus(ks)==1
                        imu_s(pos_imus(ks),i)=(imu_s(pos_imus(ks)+1,i));
                    end
                end
            end
        end
    case '0'
        for j=1:100
            for i=1:nj
                clear pos_imus
                pos_imus=find(imu_s(:,i)==0);
                ns=numel(pos_imus);
                for ks=1:ns
                    if pos_imus(ks)~=ni && pos_imus(ks)~=1
                        imu_s(pos_imus(ks),i)=(imu_s(pos_imus(ks)-1,i)
+imu_s(pos_imus(ks)+1,i))/2;
                    elseif pos_imus(ks)==ni
                        imu_s(pos_imus(ks),i)=(imu_s(pos_imus(ks)-1,
i));
                    elseif pos_imus(ks)==1
                        imu_s(pos_imus(ks),i)=(imu_s(pos_imus(ks)+1,i));
                    end
                end
            end
        end
    otherwise
        error('Wrong value for input typ')
end
```

En el anexo I se detalla la función Filtro\_singularidades que permite descartar datos irregulares de los recolectados en la experimentación.

## ANEXO II

### Calibración IMUs

```
clear;clc;
load('resultados.mat')
% Cargar registros de calibración IMUs
datos_imus=xlsread('CALIBRACION_01.xlsx');

%Filtro las celdas en cero para el registro IMUs
posZero=find(datos_imus(:,6)==0);
datos_imus(posZero(1:end),:)=[];

[ni nj nk]=size(datos_imus);
for i=1:2
tiempo(:,i)=datos_imus(:,5+(i-1)*10); %Señal de tiempo
imu_p(:,i)=datos_imus(:,6+(i-1)*10); %Señal de ángulo Pitch
imu_r(:,i)=datos_imus(:,8+(i-1)*10); %Señal de ángulo Roll
imu_y(:,i)=datos_imus(:,10+(i-1)*10); %Señal de ángulo Yaw
end

t=tiempo(:,1)-tiempo(1,1); %Desplazo el tiempo tomando en cuenta los las celdas
IMUs en cero,
n=max(size(t)); %Halla el numero de filas
time_IMUs=t(end)*(0:n-1)'/n; %Vuelvo a linealizar el tiempo con pasos iguales

[imu_p]=Filtro_singularidades(imu_p,'Error');
[imu_r]=Filtro_singularidades(imu_r,'Error');
[imu_y]=Filtro_singularidades(imu_y,'Error');

imu_e(:,:,1)=imu_p; %imu Euler
imu_e(:,:,2)=imu_r;
imu_e(:,:,3)=imu_y;

Phi_imu=permute(imu_e,[1 3 2]);
%Phi_imu(:,[1 2],:)=Phi_imu(:,[1 2],:);

%El sistema IMUs genera una singularidad al arranque por lo que se corta
este registro en 1 segundo
%corte=find(time_IMUs<1);
%Phi_imu(corte,,:)=[];
%time_IMUs(corte)=[];
t_imu=time_IMUs(:,1)-time_IMUs(1,1); %Desplazo el tiempo tomando en cuenta los las
celdas IMUs en cero,

fdamat = suavizado_bspline(Phi_imu(:,:,1),t_imu, 200, 6);
fi_brazo(:,[2 1 3],1)=fdamat(:,:,1);

fdamat = suavizado_bspline(Phi_imu(:,:,2),t_imu, 200, 6);
fi_antebrazo(:,[2 1 3],1)=fdamat(:,:,1);

clear dat_imu_euler
dat_imu_euler(:,1)=t_imu;%t;
dat_imu_euler(:,[2 3 4])=fi_brazo(:,:);
dat_imu_euler(:,[5 6 7])=fi_antebrazo(:,:);

n=max(size(Phi_imu));
```

```

for i=1:n
    Q1(i,:) = convertirrotacion(pi*fi_brazo(i,:)/180,'xyz','cuaternion');
    Q2(i,:) = convertirrotacion(pi*fi_antebrazo(i,:)/180,'xyz','cuaternion');
end

[Q1]=Filtro_singularidades(Q1,'Error');
[Q2]=Filtro_singularidades(Q2,'Error');

fdamat = suavizado_bspline(Q1,t_imu, 200, 6);
q1(:, :,1)=fdamat(:, :,1);

fdamat = suavizado_bspline(Q2,t_imu, 200, 6);
q2(:, :,1)=fdamat(:, :,1);

dat_imu_cuat(:,1)=t_imu;
dat_imu_cuat(:, [2 3 4 5])=Q1;
dat_imu_cuat(:, [6 7 8 9])=Q2;

save resultados.mat dat_fot_euler dat_imu_euler dat_fot_cuat dat_imu_cuat

[ni nj]=size(dat_imu_cuat);

Variables_imu(1,:)={'DataRate=74.000000',' ',' '};
Variables_imu(2,:)={'DataType=Quaternion',' ',' '};
Variables_imu(3,:)={'version=3',' ',' '};
Variables_imu(4,:)={'OpenSimVersion=4.1',' ',' '};
Variables_imu(5,:)={'endheader',' ',' '};
Variables_imu(6,:)={'time','r_humerus_imu','r_ulna_radius_hand_imu'};

for i=1:ni
    for j=1
        Variables_imu(i+6,j)=dat_imu_cuat(i,j);
    end
end

for i=1:ni
    Variables_imu(i+6,2)=[num2str(dat_imu_cuat(i,2)), ',', num2str(dat_imu_cuat(i,
3)), ',', num2str(dat_imu_cuat(i,4)), ',', num2str(dat_imu_cuat(i,5))];
    Variables_imu(i+6,3)=[num2str(dat_imu_cuat(i,6)), ',', num2str(dat_imu_cuat(i,
7)), ',', num2str(dat_imu_cuat(i,8)), ',', num2str(dat_imu_cuat(i,9))];
end

Tabla_exel=['A' 'B' 'C' 'D' 'E' 'F' 'G' 'H' 'I' 'J' 'K' 'L' 'M' 'N' 'O' 'P' 'Q' 'R'
'S' 'T' 'U' 'V' 'W' 'X' 'Y' 'Z'];
nj=nj+6;
ncol=nj/numel(Tabla_exel);

filename = ['data.xlsx'];
xlswrite(filename,Variables_imu,'imu',['A1:',Tabla_exel(round((ncol-fix(ncol))
*numel(Tabla_exel))),num2str(ni)]);

```

```

delete Traker_imu_cuat.sto
diary Traker_imu_cuat.sto
fprintf([Variables_imu{1,1}, '\n']);
fprintf([Variables_imu{2,1}, '\n']);
fprintf([Variables_imu{3,1}, '\n']);
fprintf([Variables_imu{4,1}, '\n']);
fprintf([Variables_imu{5,1}, '\n']);
fprintf([Variables_imu{6,1}, '\t', Variables_imu{6,2}, '\t', Variables_imu{
{6,3}, '\t', '\n'}]);
for i=7:n+6
    fprintf([num2str(Variables_imu{i,1}), '\t', Variables_imu{i,2}, '\t',
Variables_imu{i,3}, '\t', '\n']);
end
diary off

```

En el Anexo II se detalla el programa desarrollado en Matlab para el tratamiento de datos de la calibración en los sensores IMUs y preparación de datos para cargar en el software OpenSim.

## ANEXO III

### Calibración fotogrametría

```
clear;clc;
load('resultados.mat')
% Cargar registros de calibración Fotogrametría
datos=dlmread('CALIBRACION_01.txt',' ',1,0); %Registro Fotogrametría

%::::::::::Fotogrametría::::::::::
%ANALISIS POR FOTOGAMETRIA
n=max(size(datos)); %Halla el numero de frames o filas
fps=200; %frames por segundo o frecuencia en Hz
t=(0:(n-1)')/fps; %Halla el vector de tiempo, linealizar el tiempo con pasos
iguales
corte=find(t<1);
datos(corte,:)=[];
t(corte)=[];
t_fot=t(:,1)-t(1,1); %Desplazo el tiempo tomando en cuenta los las celdas en cero,
n=max(size(datos)); %Halla el numero de frames o filas
%Matrices de posiciones
MC = calcM(datos,[1:11]); %[m] % cálculo cinemático
MC0=MC(1:300, :, :);
MC0=nanmean(MC0,1); %Postura estatica de referencia

%vectores unitarios sistema
i=[1 0 0];j=[0 1 0];k=[0 0 1]; %base
%Rotación del eje "X" 90 sentido antihorario y cambio de sentido en "Y" y "X"
i1=[1 0 0];j1=[0 0 1];k1=[0 1 0]; %movimiento

%Matriz de trasformacion de coordenadas de [Y Z -X] a [X Z Y]
MT=[dot(i,i1) dot(j,i1) dot(k,i1)
dot(i,j1) dot(j,j1) dot(k,j1)
dot(i,k1) dot(j,k1) dot(k,k1)];

%Se determina las posiciones relativas
MC1=MC; %Almaceno en una variable auxiliar con indice sub 1 para todas las
posiciones que van en el centro de los tres marcadores
for i=1:n
    for j=1:11;
        MC1(i, :, j)=(MC(i, :, j)-MC0(:, :, 1));
    end
end

for i=1:n
    for j=1:11;
        MC2(i, :, j)=(MC(i, :, j)-MC0(:, :, 1))*MT;
    end
end

%Posiciones estáticas
MC01=(MC0-MC0(:, :, 1));

[omega_brazo,svo_brazo,RGv_brazo,sGv_brazo,M_brazo,RG_brazo]= finitosolid(MC1(:, :, 5:7),MC01(:, :, [5:7]));
[omega_antebrazo,svo_antebrazo,RGv_antebrazo,sGv_antebrazo,M_antebrazo,
RG_antebrazo]= finitosolid(MC1(:, :, 2:4),MC01(:, :, 2:4));
```



```

for i=1:n
    XYZ_b(i,:) = convertirrotacion(omega_brazo(i,:), 'rodrigues', 'xyz');
    XYZ_ab(i,:) = convertirrotacion(omega_antebrazo(i,:), 'rodrigues', 'xyz');
    Q1(i,:) = convertirrotacion(omega_brazo(i,:), 'rodrigues', 'cuaternion');
    Q2(i,:) = convertirrotacion(omega_antebrazo(i,:), 'rodrigues', 'cuaternion');
end
%Definimos la variable principal del par que será fi
fi_b=180*XYZ_b(:,range(XYZ_b)==max(range(XYZ_b)))/pi; % es la componente más grade
del ángulo, expresada en grados
fi_ab=180*XYZ_ab(:,range(XYZ_ab)==max(range(XYZ_ab)))/pi;

fdamat = suavizado_bspline(Q1, t, 200, 6);
q1(:, :, 1)=fdamat(:, :, 1);

fdamat = suavizado_bspline(Q2, t, 200, 6);
q2(:, :, 1)=fdamat(:, :, 1);

dat_fot_cuat(:,1)=t_fot;
dat_fot_cuat(:, [2:5])=q1;
dat_fot_cuat(:, [6:9])=q2;

data_motion(:,1)=t_fot;
data_motion(:,2)=fi_b;
data_motion(:,3)=fi_ab;

clear dat_fot_euler
dat_fot_euler(:,1)=t_fot;
dat_fot_euler(:, [2 3 4])=180*XYZ_b(:, :)/pi;
dat_fot_euler(:, [5 6 7])=180*XYZ_ab(:, :)/pi;

save resultados.mat dat_fot_euler dat_imu_euler dat_fot_cuat dat_imu_cuat

fdamat = suavizado_bspline(data_motion(:, [2 3]), t, 200, 6);
fi_fot(:, :, 1)=fdamat(:, :, 1);
Dfi_fot(:, :, 1)=fdamat(:, :, 2);
D2fi_fot(:, :, 1)=fdamat(:, :, 3);

data_motion(:, [2 3])=fi_fot;
dlmwrite('data_motion.txt', data_motion, 'delimiter', ' ')

MC2=MC2-[0.0180352 -0.792995 -0.169971];

data(:,1)=1:n;
data(:,2)=t_fot;
data(:, [3:5])= MC2(:, :, 1)*1000;
data(:, [6:8])= MC2(:, :, 8)*1000;
data(:, [9:11])= MC2(:, :, 10)*1000;

[ni nj]=size(data);

```

```

Variables_traker(1,:)={'PathFileType','4','(X/Y/Z)','Traker_fotogrametria.
trc',' ',' ',' ',' ',' ',' ',' ',' '};
Variables_traker(2,:)=
{'DataRate','CameraRate','NumFrames','NumMarkers','Units','OrigDataRate','OrigDataS
tartFrame','OrigNumFrames',' ',' ',' '};
Variables_traker(3,:)=(num2str(fps),num2str(fps),num2str(n),'3','mm',num2str
(fps),'1',num2str(n),' ',' ',' ');
Variables_traker(4,:)=
{'Frame#','Time','r_acromion',' ',' ','r_humerus_epicondyle',' ',' ','r_radius_styloid'
, ',' , '};
Variables_traker(5,:)={' ',' ','X1','Y1','Z1','X2','Y2','Z2','X3','Y3','Z3'};

for i=1:ni
    for j=1:nj
        Variables_traker(i+5,j)=data(i,j);
    end
end

delete Traker_fotogrametria.trc
clc;
diary Traker_fotogrametria.trc
fprintf([Variables_traker(1,1),'\t',Variables_traker(1,2),'\t',Variables_traker
(1,3),'\t',Variables_traker(1,4),'\t','\n']);
fprintf([Variables_traker(2,1),'\t',Variables_traker(2,2),'\t',Variables_traker
(2,3),'\t',Variables_traker(2,4),'\t',Variables_traker(2,5),'\t',Variables_traker
(2,6),'\t',Variables_traker(2,7),'\t',Variables_traker(2,8),'\t','\n']);
fprintf([Variables_traker(3,1),'\t',Variables_traker(3,2),'\t',Variables_traker
(3,3),'\t',Variables_traker(3,4),'\t',Variables_traker(3,5),'\t',Variables_traker
(3,6),'\t',Variables_traker(3,7),'\t',Variables_traker(3,8),'\t','\n']);
fprintf([Variables_traker(4,1),'\t',Variables_traker(4,2),'\t',Variables_traker
(4,3),'\t',Variables_traker(4,4),'\t',Variables_traker(4,5),'\t',Variables_traker
(4,6),'\t',Variables_traker(4,7),'\t',Variables_traker(4,8),'\t',Variables_traker
(4,9),'\t','\n']);
fprintf([Variables_traker(5,1),'\t',Variables_traker(5,2),'\t',Variables_traker
(5,3),'\t',Variables_traker(5,4),'\t',Variables_traker(5,5),'\t',Variables_traker
(5,6),'\t',Variables_traker(5,7),'\t',Variables_traker(5,8),'\t',Variables_traker
(5,9),'\t',Variables_traker(5,10),'\t',Variables_traker(5,11),'\t','\n']);
for i=6:n+5
    fprintf([num2str(Variables_traker(i,1)),'\t',num2str(Variables_traker(i,
2)),'\t',num2str(Variables_traker(i,3)),'\t',num2str(Variables_traker(i,4)),'\t',
num2str(Variables_traker(i,5)),'\t',num2str(Variables_traker(i,6)),'\t',num2str
(Variables_traker(i,7)),'\t',num2str(Variables_traker(i,8)),'\t',num2str
(Variables_traker(i,9)),'\t',num2str(Variables_traker(i,10)),'\t',num2str
(Variables_traker(i,11)),'\t','\n']);
end
diary off

Tabla_exel=['A' 'B' 'C' 'D' 'E' 'F' 'G' 'H' 'I' 'J' 'K' 'L' 'M' 'N' 'O' 'P' 'Q' 'R'
'S' 'T' 'U' 'V' 'W' 'X' 'Y' 'Z'];
ncol=nj/numel(Tabla_exel);

filename = ['data.xlsx'];

```



```

xlswrite(filename,Variables_traker,'traker',[ 'A1:',Tabla_exel(round((ncol-fix
(ncol))*numel(Tabla_exel))),num2str(ni)]);

[ni nj]=size(data_motion);

Variables_motion(1,:)={'Motion_fot_matlab',' ',' '};
Variables_motion(2,:)={'version=1',' ',' '};
Variables_motion(3,:)={'nRows=',num2str(n),' ',' '};
Variables_motion(4,:)={'nColumns=3',' ',' '};
Variables_motion(5,:)={'inDegrees=yes',' ',' '};
Variables_motion(6,:)={' ',' ',' '};
Variables_motion(7,:)={'Units are S.I. units (second, meters, Newtons,
...)' ,' ',' '};
Variables_motion(8,:)={'Angles are in degrees.' ,' ',' '};
Variables_motion(9,:)={' ',' ',' '};
Variables_motion(10,:)={'endheader' ,' ',' '};
Variables_motion(11,:)={'time','r_shoulder_elev','r_elbow_flex'};

for i=1:ni
    for j=1:nj
        Variables_motion(i+11,j)=data_motion(i,j);
    end
end

delete Motion_fot_Matlab.mot
clc;
diary Motion_fot_Matlab.mot
fprintf([Variables_motion(1,1),' \t', '\n']);
fprintf([Variables_motion(2,1),' \t', '\n']);
fprintf([Variables_motion(3,1),' \t', '\n']);
fprintf([Variables_motion(4,1),' \t', '\n']);
fprintf([Variables_motion(5,1),' \t', '\n']);
fprintf([Variables_motion(6,1),' \t', '\n']);
fprintf([Variables_motion(7,1),' \t', '\n']);
fprintf([Variables_motion(8,1),' \t', '\n']);
fprintf([Variables_motion(9,1),' \t', '\n']);
fprintf([Variables_motion(10,1),' \t', '\n']);
fprintf([Variables_motion(11,1),' \t', Variables_motion(11,2),' \t', Variables_motion
(11,3),' \t', '\n']);
for i=12:n+11
    fprintf([num2str(Variables_motion(i,1)), '\t', num2str(Variables_motion(i,
2)), '\t', num2str(Variables_motion(i,3)), '\t', '\n']);
end
diary off

ncol=nj/numel(Tabla_exel);

filename = ['data.xlsx'];
xlswrite(filename,Variables_motion,'motion',[ 'A1:',Tabla_exel(round((ncol-fix
(ncol))*numel(Tabla_exel))),num2str(ni)]);

```

En el Anexo III se detalla el programa desarrollado en Matlab para el tratamiento de datos de la calibración del sistema de fotogrametría.

## ANEXO IV

### Función desfase\_H

```
function pos_desfase=desfase_H(data,t)

Pos=find(nanstd(data)==max(nanstd(data)));
nt=find(t<=4);

%fdamat = suavizado_bspline(data(:,:,1), t, 200, 6);
fdamat = suavizado_bspline(data(:,:,1), t, 200, 4);
Fi(:,:,1)=fdamat(:,:,1);
DFi(:,:,1)=fdamat(:,:,2);
D2Fi(:,:,1)=fdamat(:,:,3);

qs=Fi(:,Pos);
qs=qs-nanmean(qs);
v=DFi(:,Pos);

n=max(size(Fi)); %Halla el numero de frames o filas

% ahora hacemos un vector de indices
i=1:n;
%Sacamos los puntos en los que la velocidad es positiva. No pongo cero para
%cortar exponenciales
j=i(ne(0,diff(v>prctile(abs(v),5)))); % esto sa los indices donde la velocidad
tiene maximo o minimo
% como aqui empezamos en cero, es posible que al principio o final haya
% puntos de velocidad cero con poco desplazamiento. Eliminamos aquellos
% cuyo fi sea inferior a un umbral, por ejemplo un tercio del p95
kk=abs(qs(j))>(prctile(abs(qs(j)),95)/2);
j=j(kk);
j=j(find(j>nt(end)));
m=max(size(j));
k=1:m; % para tene un vector con los indices de j y asi poder sacar uego cosas
% umbral de salto entre dos máximos o mínimos
u=diff(prctile(qs,[25 75]))/2;
%iel=logical([(diff(qs(j)')>u) 0]);
ii=sign(qs(j)')<0; %logical([(diff(qs(j)')>u) 0]);
ii=[ii(1) diff(ii)];
iel=j(logical(ii==1)); %indices donde terminan los ciclos de subida
%iel=j(logical(iel==1)); %indices donde terminan los ciclos de subida
pos_desfase=iel(1,2);
end
```

En el anexo IV se detalla la función desfase\_H utilizada en la validación de las técnicas de captura de movimiento de fotogrametría e IMUs.