

E.T.S. de Ingeniería Industrial,  
Informática y de Telecomunicación

# Cálculo y visualización de rangos articulares de las extremidades superiores en tiempo real



Grado en Ingeniería Informática

Trabajo Fin de Grado

**Autora:** Elisa Oreja Murillo

**Tutores:** Pablo Lecumberri Villamediana y Marisol Gómez Fernández

Pamplona, 23 de enero 2019

# **CÁLCULO Y VISUALIZACIÓN DE RANGOS ARTICULARES DE LAS EXTREMIDADES SUPERIORES EN TIEMPO REAL**



Autora: Elisa Oreja Murillo  
Tutores: Pablo Lecumberri Villamediana y Marisol Gómez Fernández

Departamento de Estadística, Matemáticas e Informática  
Universidad Pública de Navarra

Enero de 2019



# Agradecimientos

Me gustaría dar mis más sinceros agradecimientos a todo el equipo de Movalsys, por haberme acogido y haberme ayudado en todo lo posible desde el primer momento.

A Itziar y Berta, por amenizarme todas las mañanas, haberme dado tanta confianza desde el principio y crear un ambiente tan agradable en el laboratorio.

A Pablo, por esa paciencia infinita con todos y cada uno de los problemas que me surgían en el día a día.

Y a todos en general, por haber depositado en mí esa confianza gracias a la cual he ido creciendo un poquito más como persona.



# Resumen

**Abstract** — The principal objective of this project is helping, thanks to technology that we have these days, to rehabilitate patients who have motor troubles in some of their limbs. In this case, we will focus on the upper ones, the arms, considering the shoulder and elbow joints.

In this project, the main goal is to obtain and visualize joint angles for the upper limbs, thanks to the received data from three different inertial sensors which have to be located on chest, upper arm and forearm.

In this way, the progress of a lesion can be tracked, so that the specialists can understand and interpret the obtained results, and decide the best way to rehabilitate the patient.

**Key words** — Joint angles, inertial sensors



# Resumen

**Resumen** — El objetivo principal de este proyecto es ayudar, gracias a la tecnología de la que hoy en día disponemos, a rehabilitar a pacientes con dificultades motoras en alguna de las extremidades de su cuerpo. En este caso, nos centraremos en los brazos, teniendo en cuenta para su estudio las articulaciones del hombro y del codo.

En este proyecto concreto, el objetivo principal es obtener y visualizar rangos articulares de las extremidades superiores. Para ello se usan los datos proporcionados por tres sensores inerciales colocados en el pecho, brazo y antebrazo.

Así, se puede observar cómo progresa la persona que padece una lesión o alteración motora. Además, los especialistas sanitarios pueden interpretar fácilmente los resultados obtenidos y decidir cuál es la rehabilitación más adecuada para cada paciente.

**Palabras clave** — Ángulos articulares, sensores inerciales





# Índice general

<b>1. Introducción</b>	<b>3</b>
<b>2. Estado del Arte</b>	<b>7</b>
2.1. Sistemas de medición de la movilidad articular . . . . .	7
2.1.1. Instrumentos de medición . . . . .	7
<b>3. Metodología</b>	<b>11</b>
3.1. Cómputo de los ángulos . . . . .	12
3.1.1. Sensores inerciales . . . . .	12
3.1.2. Procesado de señales . . . . .	20
3.2. Visualización . . . . .	28
3.2.1. Librería Ogre3D . . . . .	29
3.2.2. Elementos del interfaz . . . . .	37
<b>4. Resultados y análisis</b>	<b>39</b>
4.1. Resultados . . . . .	39
<b>5. Conclusiones y líneas futuras</b>	<b>43</b>
5.1. Conclusiones . . . . .	43
5.2. Líneas futuras . . . . .	44
<b>Bibliografía</b>	<b>45</b>
<b>6. Anexos</b>	<b>47</b>



# Índice de figuras

1.1. Quirófano de hoy en día [1] . . . . .	4
1.2. Representación del control médico de hoy en día de manera digital [2] . . . . .	5
1.3. Medición de ángulos articulares a partir de los ejes longitudinales de los huesos del dedo pulgar de la mano [7] . . . . .	5
2.1. Sistematización de la goniometría [8] . . . . .	8
2.2. Regla utilizada para la medición analógica de ángulos (Tómese como referencia la Figura 2.1) [9] . . . . .	8
2.3. Electrogoniómetro [11] . . . . .	9
2.4. Flexómetro de Leighton [12] . . . . .	10
2.5. Ejemplo de utilización del software <i>Kinovea</i> . . . . .	10
3.1. Comparación de posiciones entre los ejes en el sistema global y los ejes propios del sensor . . . . .	14
3.2. Ejes horizontal y frontal [8] . . . . .	15
3.3. Eje vertical [8] . . . . .	15
3.4. Comparación de posiciones entre los ejes en el sistema de los segmentos del cuerpo y los ejes propios del sensor . . . . .	16
3.5. Movimiento de flexión/extensión del codo [16]. . . . .	18
3.6. Movimiento de pronación/supinación del codo [17]. . . . .	18
3.7. Esquema de obtención de relación entre la orientación del sistema del brazo y del sistema global . . . . .	19

3.8. Diagrama de clases de la medición que proporcionan los sensores . . . . .	20
3.9. Realizamos primero traslación y después rotación [18]. . . . .	24
3.10. Realizamos primero rotación y después traslación [18]. . . . .	24
3.11. Rotación de vectores en el espacio tridimensional . . . . .	24
3.12. Rotación de un vector . . . . .	25
3.13. Rotación de un vector de un ángulo $\beta$ . . . . .	25
3.14. Representación de los ejes con los que se cumplen una serie de propiedades más abajo mostradas . . . . .	28
3.15. Logo Ogre3D [19] . . . . .	29
3.16. Ejemplos de <i>mesh</i> [21] . . . . .	30
3.17. Disposición de los ejes en cada caso . . . . .	31
3.18. Representación de los ejes sobre las partes de las articulaciones superiores en la posición inicial del esqueleto en el programa . . . . .	32
3.19. Representación de los ejes sobre las partes de las articulaciones superiores en un ejemplo de posición del paciente . . . . .	33
3.20. Cuaterniones que conocemos y cuaternión que necesitamos hallar . . . . .	34
3.21. Resultado de posición del vector $u$ al cambiarlo de sistema de coordenadas . . . . .	35
3.22. Rotación del brazo debido a la rotación del pecho . . . . .	37
3.23. Interfaz de la aplicación . . . . .	38
4.1. Funcionamiento del programa: Paciente con sensores colocados . . . . .	40
4.2. Funcionamiento del programa: Resultado de reproducción de movimiento obtenido en pantalla . . . . .	40
4.3. Señales de apertura de ángulo del brazo realizando el movimiento de flexión . . . . .	41
4.4. Señales de apertura de ángulo del brazo realizando el movimiento de pronación/supinación . . . . .	41



# 1

## Introducción

La memoria que se presenta está dividida en los siguientes capítulos:

- CAPÍTULO 1: Introducción.
- CAPÍTULO 2: Estado del arte.
- CAPÍTULO 3: Metodología.
  - Cómputo de los ángulos.
  - Visualización.

## CAPÍTULO 1. INTRODUCCIÓN

---

- CAPÍTULO 4: Resultados.
- CAPÍTULO 5: Conclusiones y líneas futuras.
- CAPÍTULO 6: Anexos.

Desde su origen, el ser humano hace ejercicio físico o cierto tipo de trabajos adoptando posturas que no resultan saludables, sobre todo cuando se realiza diariamente y de manera continuada. Éste puede ser el caso de trabajadores de cadenas de producción de diferentes sectores industriales o de los que realizan labores de mantenimientos, por ejemplo, en los modernos aerogeneradores donde aparece la dificultad añadida del espacio reducido donde se trabaja. Esto, a veces, puede ocasionar lesiones músculo esqueléticas.

Por otro lado, por determinadas razones o tal vez sin motivo aparente, pueden tener lugar lesiones cerebrales, como podrían ser un traumatismo craneoencefálico o un accidente cerebrovascular.

No hace falta que nos vayamos excesivamente atrás en el tiempo para poder apreciar el gran avance que ha supuesto la tecnología en la mejora nuestra salud. Solamente remontándonos 100 años atrás, apenas había coches, por lo que por aquel entonces era completamente impensable que pocos años después fuese a implantarse la tecnología en los hospitales o centros dedicados a la ayuda en la salud de las personas.



Figura 1.1: Quirófano de hoy en día [1]

Gracias a este rápido desarrollo de la tecnología, hoy en día podemos, por ejemplo:



- Prevenir muchas de las posibles enfermedades mediante un diagnóstico precoz.
- Realizar operaciones quirúrgicas, demasiado delicadas como para llevarlas a cabo con nuestras propias manos, mediante robots.
- Ayudar a una recuperación más temprana, eficaz y personalizada de sujetos que ya han sufrido un daño.

Dentro de este último apartado es donde se enmarca el desarrollo de este proyecto.



Figura 1.2: Representación del control médico de hoy en día de manera digital [2]

En particular, este proyecto está directamente relacionado con la ergonometría, que consiste en el estudio del uso adecuado y eficiente del cuerpo en el trabajo y el descanso, incluyendo el diseño de máquinas y del medio ambiente [3].

Por otra parte, la palabra goniometría deriva del griego: *gonion*, "ángulo- *metron*, "medición", lo cual nos da su significado: "Disciplina que se encarga de estudiar la medición de los ángulos"[4], que es de lo que se va a encargar nuestro programa.

La kinesiología, según la RAE, es el método terapéutico por medio de movimientos activos o pasivos de todo el cuerpo o de alguna de sus partes [5]. Es decir, esta disciplina busca la mejoría física a través del movimiento, intentando que los tiempos de recuperación sean menores y recuperar el máximo de capacidades funcionales.

Dentro de la kinesiterapia, la goniometría es la técnica de medición de los ángulos creados por la intersección de los ejes longitudinales de los huesos a nivel de las articulaciones [6].

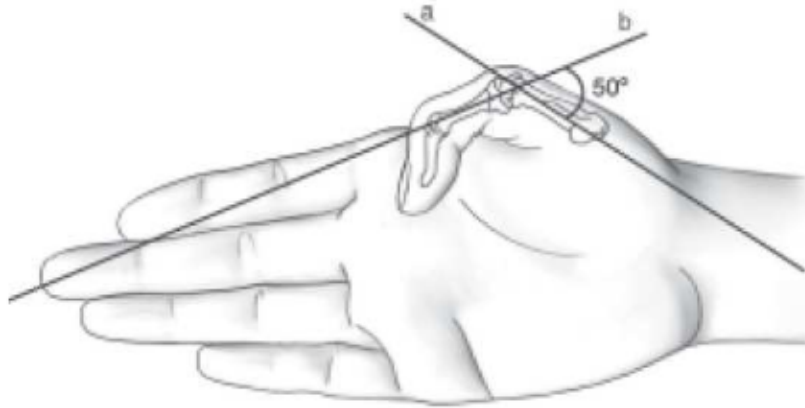


Figura 1.3: Medición de ángulos articulares a partir de los ejes longitudinales de los huesos del dedo pulgar de la mano [7]



# 2

## Estado del Arte

### **2.1. Sistemas de medición de la movilidad articular**

#### **2.1.1. Instrumentos de medición**

##### **Goniómetro**

El instrumento más utilizado para medir ángulos y determinar rangos articulares es el goniómetro. En el mercado también existen diferentes tipos de reglas y de niveles que pueden

## CAPÍTULO 2. ESTADO DEL ARTE

---

utilizarse para llevar a cabo ese mismo fin. También podemos nombrar los flexómetros que poseen un conductor de fibra óptica que, gracias al movimiento de la fibra, miden ángulos articulares.

Para poder llevar a cabo una medición precisa con este tipo de métodos, habría que tener en cuenta la posición del paciente y de los ejes longitudinales de sus huesos, y la posición del especialista encargado de realizar la medición. Es decir, resultados obtenidos utilizando este tipo de dispositivos presentan una gran subjetividad.

Por otro lado, resulta complicado realizar mediciones dinámicas con la instrumentación antes mencionada. Por ejemplo, el goniómetro se debe ajustar para cada movimiento, lo cual no lo hace nada práctico.

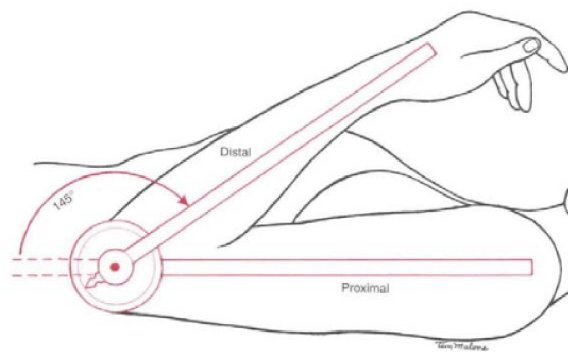


Figura 2.1: Sistematización de la goniometría [8]

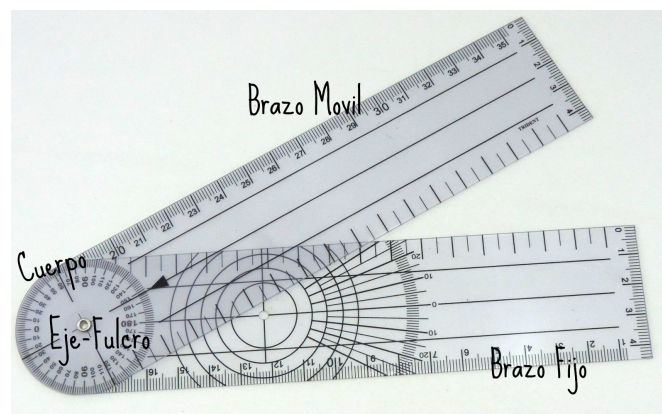


Figura 2.2: Regla utilizada para la medición analógica de ángulos (Tómese como referencia la Figura 2.1) [9]

### **Electrogoniómetro**

Un electrogoniómetro consiste en un goniómetro conectado a una batería eléctrica que permite un registro directo de los datos sobre un papel graduado. Las variaciones angulares son visualizadas en un osciloscopio y, simultáneamente, se plasman esos datos en un gráfico [10]

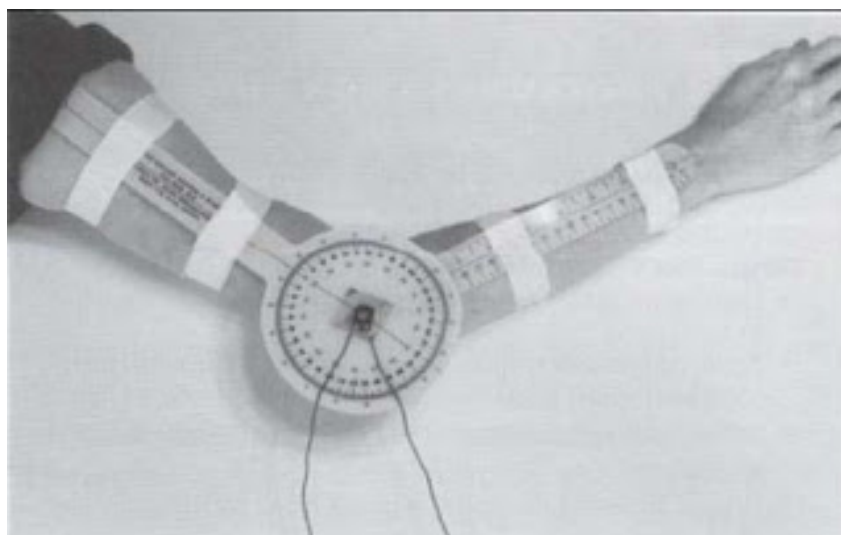


Figura 2.3: Electrogoniómetro [11]

### **Flexómetro de Leighton**

Creado en 1960, el flexómetro de Leighton mide la flexibilidad en función de grados. Consiste en un cilindro metálico dentro del cual hay dos partes móviles: un disco graduado de 0 a 360 grados y una aguja. Cuando el cilindro es desplazado de la vertical, merced a un mecanismo interno propio del aparato, tanto el disco como la aguja se mueven [10].



Figura 2.4: Flexómetro de Leighton [12]

### **Análisis biomecánico mediante vídeo**

Existen otras opciones un poco diferentes a las anteriores, como pueden ser los métodos de cálculo de ángulos articulares a partir de una imagen de vídeo. Esto es algo que realiza el reproductor de vídeo *Kinovea*.

Para ello, se seleccionan o marcan los puntos que corresponden a cada articulación y, el usuario manualmente con el ratón une dicha articulación con el suelo (proyección del punto en el suelo), de manera que el programa es capaz de calcular la distancia que hay entre ambos. Gracias a los datos marcados, mediante el algoritmo que lleva implementado y gracias a la visión artificial, el software es capaz de calcular un ángulo articular y dar su valor en grados.

Por ejemplo, en la imagen que se muestra a continuación, se pueden ver los grados que un deportista ha sido capaz de doblar la rodilla al levantar la pierna durante un saque de beisbol.

Este tipo de herramienta suele utilizarse de manera habitual en el mundo deportivo, aunque puede extenderse a cualquier otro campo. Su manejo es sencillo ya que únicamente necesitamos tener el video que nos interesa para poder realizar el cálculo. Como contrapartida presenta un grado de imprecisión bastante alto, ya que según la perspectiva de la imagen o de dónde esté

exactamente el punto de articulación de la persona puede que obtengamos datos imprecisos.

Además, debido a la Ley de Protección de Datos, no está permitido tener imágenes de usuarios almacenados de manera no anónima.



Figura 2.5: Ejemplo de utilización del software *Kinovea* [13]





# 3

## Metodología

En esta sección, se explica más en detalle cómo se ha llevado a cabo el cómputo de los ángulos, usando sensores inerciales y analizando las señales que nos proporcionan, gracias a la librería TSys. También se expone cómo se visualizan los mismos utilizando la librería gráfica Ogre3D con la que se ha trabajado en el lenguaje de programación C++.

El IDE (Entorno de Desarrollo Integrado) utilizado durante el desarrollo del proyecto ha sido QtCreator, de manera que el uso de las librerías de Qt (framework multiplataforma orientado a objetos) resulte más sencillo para llevar a cabo el desarrollo de interfaces gráficas de usuario. Este IDE trae consigo un editor de texto integrado con herramientas para el diseño,

codificación, testeo, implementación y mantenimiento del software a lo largo del ciclo de vida del producto [14].

Aparte de esto, se han utilizado tres librerías adicionales:

- La librería gráfica Ogre3D, ya mencionada anteriormente, para poder representar gráficamente un esqueleto replicando los movimientos del paciente.
- La librería de widgets QWT (Qt Widgets for Technical Applications), para poder utilizar clases C++ para la representación gráfica de señales.
- La librería TSys, utilizada para poder interpretar y tratar en el software las señales que nos llegan desde los sensores colocados en el paciente.

### 3.1. Cómputo de los ángulos

En este proyecto se pretende conocer cuántos grados es capaz nuestro paciente de flexionar el hombro y/o el codo.

A la hora de calcular dichos ángulos podemos distinguir claramente dos partes: la que tiene que ver con los sensores inerciales usados y la parte del procesado de las señales obtenidas de cada sujeto al realizar un movimiento.

#### 3.1.1. Sensores inerciales

Un sistema de referencia inercial se trata de un sistema de referencia regido por las leyes de movimiento de Newton. Un sensor capaz de medir magnitudes cinemáticas respecto a dicho sistema de referencia es lo que se conoce como un sensor inercial [15]. Estos sensores están compuestos por tres elementos principales, que son:

- Acelerómetros
- Magnetómetros
- Giroscopios

De cada uno de ellos tendremos tres, uno por cada eje tridimensional.

Los acelerómetros permiten, como su nombre indica, determinar cuál es la aceleración que recae sobre cada uno de los ejes del sensor.

Los magnetómetros detectan el campo magnético, de manera que los sensores, gracias a ellos, detectarán el Norte magnético, y éste será el eje  $x$  de las coordenadas del sistema global fijo que vamos a necesitar más adelante para realizar las mediciones. Los ejes  $y$  y  $z$  estarán colocados perpendicularmente a él tal y como vemos en la imagen de la figura 3.1

Los giroscopios permiten medir la velocidad angular. Así, es posible determinar cuánto rota el sensor alrededor de los tres ejes, y por tanto, saber si el paciente ha cambiado de postura.

Para poder llevar a cabo la medición de los ángulos de apertura de las articulaciones mediante los sensores inerciales, tendremos que seguir un protocolo de calibrado de los mismos. Esto es debido a que los ejes propios del sensor no son los mismos que los del sistema de coordenadas globales (Figura 3.1), ni tampoco que los del sistema de coordenadas del segmento en el que lo colocamos (Figura 3.4):

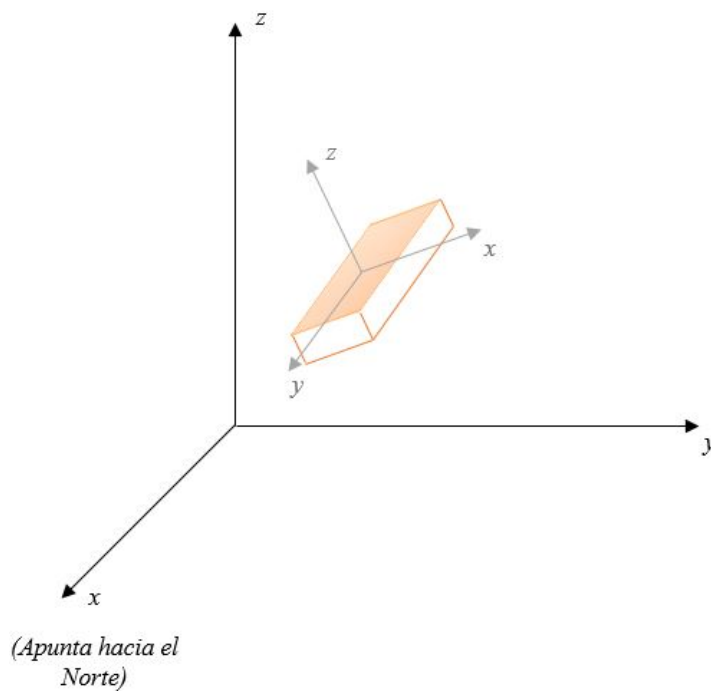


Figura 3.1: Comparación de posiciones entre los ejes en el sistema global y los ejes propios del sensor

El proceso de calibración consta de dos partes:

1. Relación con el sistema global
2. Relación con el sistema de coordenadas del segmento anatómico correspondiente a cada sensor

Los sensores proporcionan los cuaterniones y éstos sirven para relacionar las coordenadas del sistema de referencia del sensor con el sistema de referencia global. Cuando se coloca un sensor en un paciente, dicho sensor no estará calibrado. Es decir, en ese momento sólo conocemos la posición en la que están los ejes en el sistema global, y la posición de los ejes propios del sensor, tal y como se ha visualizado en la Figura 3.1.

Para llevar a cabo la primera parte de la calibración, lo primero que debemos hacer es colocar los tres sensores juntos, para que éstos tomen como referencia el mismo sistema global.

En teoría, todos deberían tomar el mismo, pero como puede haber pequeñas desviaciones, tomaremos uno de los tres como referencia y si alguno saliera distinto, lo convertiremos para tener un sistema global único e idéntico para todos ellos.

Para realizar la segunda parte de la calibración, debemos realizar una serie de movimientos concretos, teniendo en cuenta los ejes horizontal, frontal y vertical, mostrados en las siguientes imágenes:

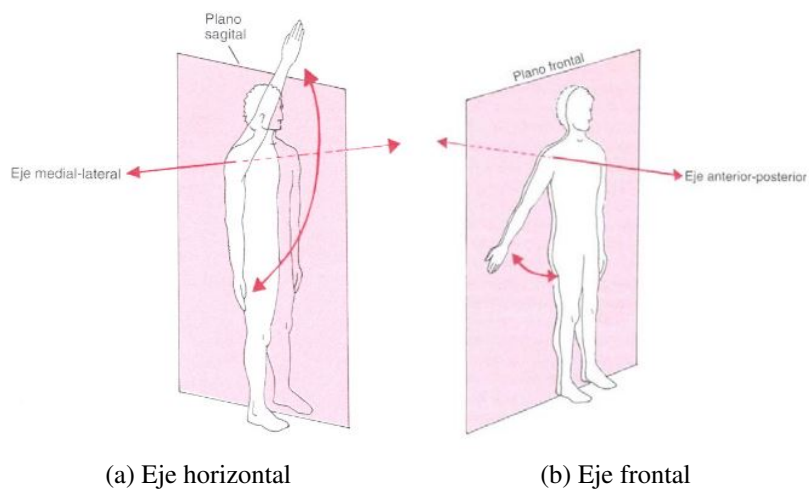


Figura 3.2: Ejes horizontal y frontal [8]

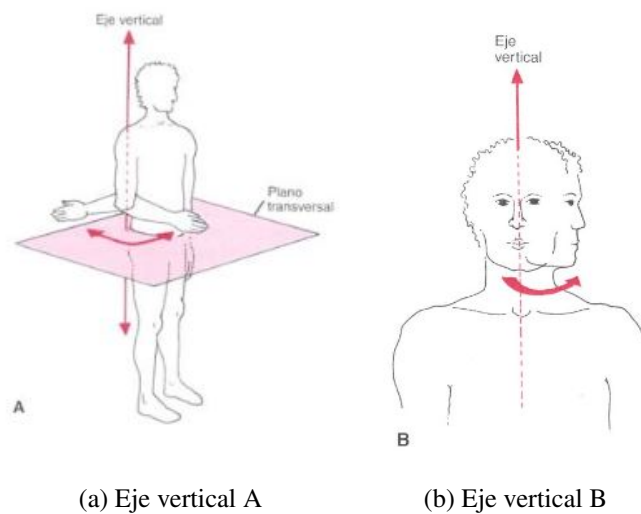


Figura 3.3: Eje vertical [8]

Para determinar cómo está orientado cada segmento del brazo con respecto al sistema global, es necesario conocer primero la relación entre el sistema del sensor y el sistema del segmento al que va unido. El cuaternión que realiza la transformación de uno a otro se calcula mediante el proceso de calibración de los sensores con respecto a los segmentos del cuerpo correspondientes.

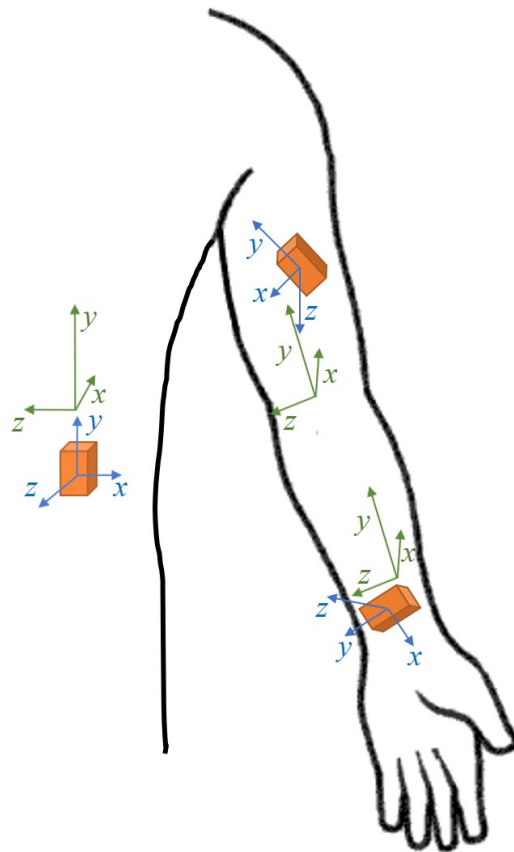


Figura 3.4: Comparación de posiciones entre los ejes en el sistema de los segmentos del cuerpo y los ejes propios del sensor

Para ello, nos colocamos los tres sensores, cada uno en su lugar correspondiente, y llevaremos a cabo tres pasos para realizar una correcta calibración:

1. Con el brazo recto hacia abajo, esperamos en esta postura durante 3 segundos. En este paso no importa si el codo está o no flexionado.

2. Con el brazo recto hacia abajo y el codo doblado 90 grados, con el antebrazo realizamos un movimiento de pronación/supinación durante 5 segundos (girar el antebrazo de manera que la palma de la mano vaya girando hacia arriba y hacia abajo).
3. Comenzando también con el codo doblado 90 grados, doblamos y estiramos el mismo hacia arriba y hacia abajo, con la palma de la mano mirando hacia nuestro cuerpo, durante otros 5 segundos.

Una vez realizados estos tres pasos, los sensores están calibrados correctamente, y el especialista encargado de llevar a cabo la rehabilitación del paciente podría dar comienzo a la medición de los ángulos. Lo que éste estudiaría sería, en el caso del ángulo del hombro:

- **Ángulo de flexión:** Ángulo con la vertical en el plano sagital. Correspondiente al movimiento mostrado en la figura 3.2a.
- **Ángulo de abducción:** Ángulo con la vertical en el plano frontal. Correspondiente al movimiento mostrado en la figura 3.2b.
- **Ángulo de rotación:** Ángulo con la horizontal en el plano transversal. Correspondiente al movimiento mostrado en la figura 3.3a

En el caso del codo, el estudio se realizaría de manera análoga, teniendo en cuenta los movimientos correspondientes a las siguientes imágenes:



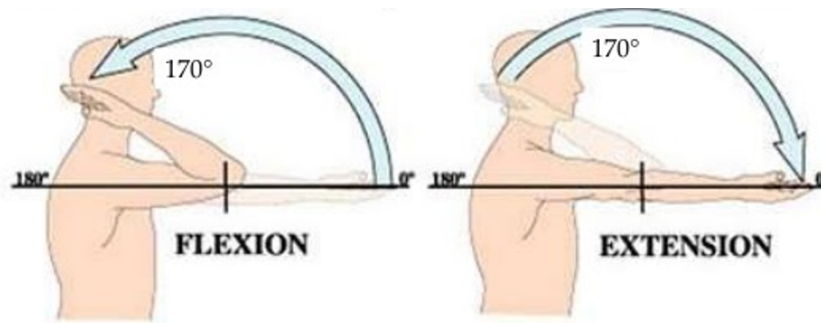


Figura 3.5: Movimiento de flexión/extensión del codo [16].

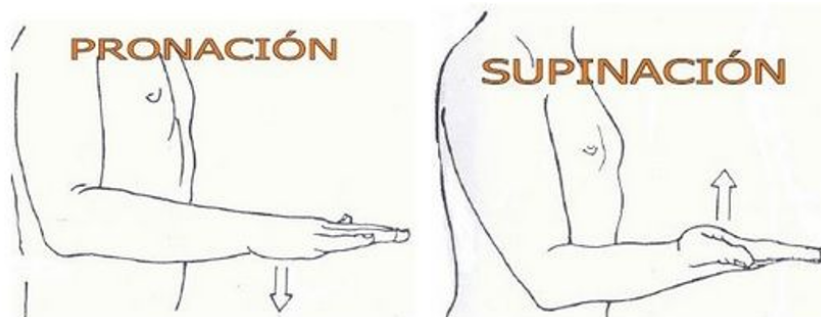


Figura 3.6: Movimiento de pronación/supinación del codo [17].

Ahora, lo que nos permite detectar la velocidad angular es, como nombrábamos antes, el giroscopio del sensor, de manera que integrándola, podemos obtener la nueva orientación adquirida. Esto no es preciso al 100 %, por lo que entre los giroscopios, los magnetómetros y los acelerómetros, mediante una serie de operaciones matemáticas, llevan a cabo una corrección del error cometido para que la nueva orientación del sensor proporcionada con respecto al sistema global sea fiable.

En la siguiente figura podemos ver cómo relacionamos los diferentes sistemas de coordenadas para conseguir saber cómo está situado el brazo con respecto a las coordenadas del sistema global:

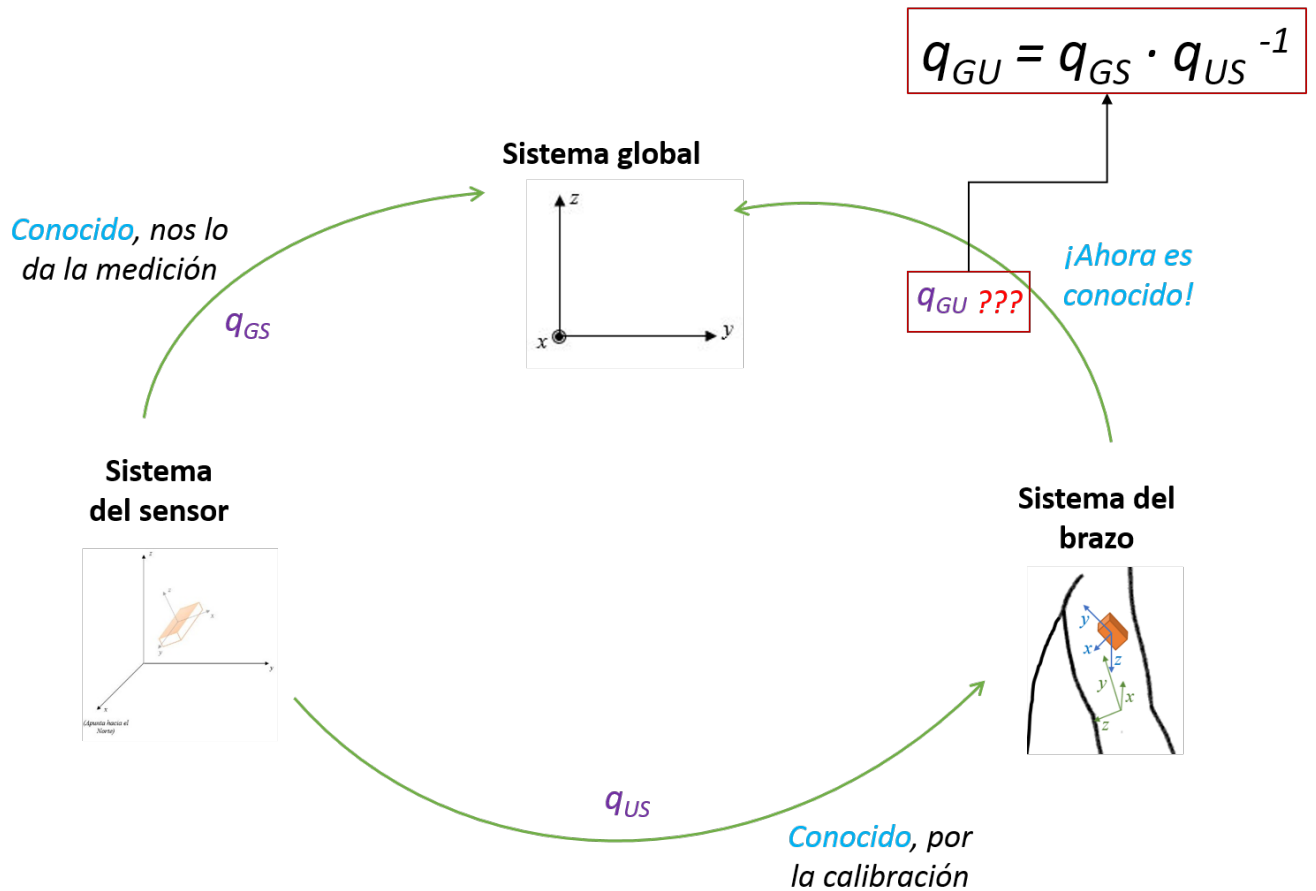


Figura 3.7: Esquema de obtención de relación entre la orientación del sistema del brazo y del sistema global

De igual manera, ahora que ya conocemos la relación entre el sistema de cada segmento anatómico y el sistema global, si quisiéramos empezar a medir desde un punto que nosotros queremos fijar, lo que haríamos no sería poner el punto inicial como  $0^\circ$ , sino que lo que haríamos sería calcular la diferencia entre los grados correspondientes al punto inicial y al final. Por ejemplo, nosotros de entrada sabemos que si una persona se encuentra con los brazos hacia abajo, teniendo la palma orientada hacia su propio cuerpo, su hombro y su codo se encuentran en posición de flexión = 0, por lo que comenzaríamos a medir desde ese punto de referencia.

### 3.1.2. Procesado de señales

En este apartado, se describirá cómo se tratan los datos obtenidos a partir de las señales que nos proporcionan los sensores.

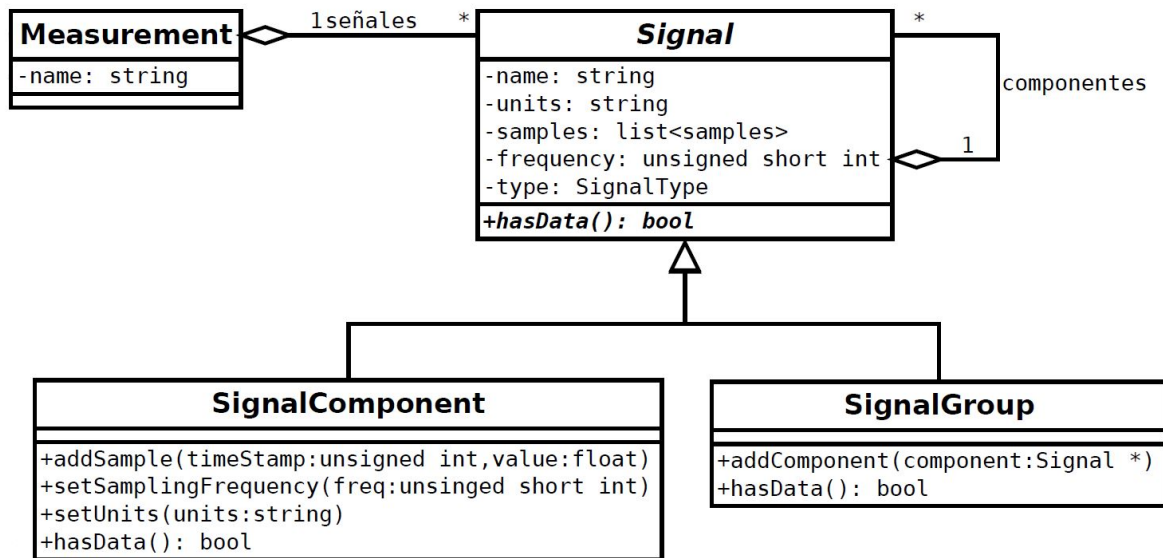


Figura 3.8: Diagrama de clases de la medición que proporcionan los sensores

Los sensores enviarán al programa una serie de mediciones (*Measurement*), las cuales están compuestas por una o varias señales (*Signal*), que son:

- Grupos de señales (*SignalGroup*)
- Componentes de señal (*SignalComponent*)

Por ejemplo, en el programa realizado, existe una clase llamada *UpperArm3DView*, en la que cogemos el *Measurement*, y extraemos los grupos de señal del mismo en relación a su tipo (el tipo en este caso sería hombro o codo), de manera que tendremos el *ShoulderGroup* y el *ElbowGroup*, dentro de los cuales estarán los *SignalComponent* que les corresponde.

Un *SignalGroup* está compuesto por varias *SignalComponent*. Si por ejemplo, nos fijamos en la velocidad angular, tendríamos como *SignalGroup* a la *velocidadAngular*, y dentro de

ella, los tres SignalComponent correspondientes a *velocidadAngular\_x*, *velocidadAngular\_y* y *velocidadAngular\_z*. . Lo mismo ocurriría con la aceleración ya que, dentro de su grupo de señales, también tiene de tres componentes, una por cada uno de los ejes.

Para llevar a cabo el cómputo de los ángulos a partir de los datos que nos proporcionan los sensores en forma de señal, utilizamos los cuaterniones.

Un cuaternión es un elemento que está definido por lo siguiente:

- Un eje (será el eje de rotación).
- Un ángulo (ángulo que rotaremos alrededor del eje)

Podemos representarlo de la siguiente manera:

$$Q = (\text{escalar}, \text{vector}.x, \text{vector}.y, \text{vector}.z)$$

Es recomendable utilizar cuaterniones por las siguientes razones:

- No dependen de un eje primario.
- La multiplicación y la división entre cuaterniones es más precisa.
- Se pueden interpolar.
- Se pueden convertir en una matriz de rotación.

[18]

### Producto de cuaterniones

$q_1 \cdot q_2 = r = (r_s, r_x, r_y, r_z)$ , siendo  $r_s$  la parte escalar del cuaternión, y  $r_x, r_y, r_z$ , la parte vectorial del mismo.

Cada una de estas dos partes que componen al cuaternión, cuando realizamos un producto, se calculan de la siguiente manera:

$$r_s = q_{1s} \cdot q_{2s} - \vec{q}_{1v} \cdot \vec{q}_{2v}$$

$$\vec{r}_v = q_{1s} \cdot \vec{q}_{2v} + q_{2s} \cdot \vec{q}_{1v} + \vec{q}_{1v} \times \vec{q}_{2v}$$

Cuando multiplicamos un cuaternión por su inverso, obtenemos al cuaternión identidad:

$$q \cdot q^{-1} = e, \text{ siendo } e = (1, 0, 0, 0).$$

El cuaternión identidad es el elemento neutro para el producto de cuaterniones, por lo que:

$$q \cdot e = q$$

### Obtención de un cuaternión a partir de un ángulo y un vector

Dado un ángulo  $\alpha$  de giro y un vector  $\vec{v}$  como eje de rotación, a partir de ellos podemos crear un cuaternión unitario de la siguiente manera:

$$q = \left( \cos \frac{\alpha}{2}, \sin \frac{\alpha}{2} \cdot \vec{v} \right) = \left( \cos \frac{\alpha}{2}, \sin \frac{\alpha}{2} \cdot v_x, \sin \frac{\alpha}{2} \cdot v_y, \sin \frac{\alpha}{2} \cdot v_z \right)$$

Dentro de Qt, por ejemplo, existe un método llamado *fromAxisAndAngle(axis, angle)* que nos crea automáticamente el cuaternión a partir de esos dos elementos.

### Representación de una rotación mediante un cuaternión

Para poder decir que un cuaternión representa una rotación, éste debe cumplir que su norma sea igual a la unidad:

$$q_s^2 + q_x^2 + q_y^2 + q_z^2 = 1$$

El cuaternión inverso de uno que representa una rotación (es decir, que es unitario por ser su norma = 1) coincide con el conjugado:  $q^{-1} = q^* = (q_s, -q_x, -q_y, -q_z)$

El producto entre un cuaternión y su inverso es uno de los pocos casos en los que se cumple la propiedad conmutativa entre cuaterniones.

Demostración de que el cuaternión conjugado es el inverso de un cuaternión de norma 1:

$$r_s = q_s \cdot q_s - (q_x \cdot (-q_x) + q_y \cdot (-q_y) + q_z \cdot (-q_z)) = q_s^2 + q_x^2 + q_y^2 + q_z^2 = 1$$

$$\begin{aligned} r_v &= q_s \cdot (-q_x, -q_y, -q_z) + q_s \cdot (q_x, q_y, q_z) + q_v \times q_v^{-1} = q_s \cdot (-q_x, -q_y, -q_z) + q_s \cdot (q_x, q_y, q_z) + \\ &\begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ q_x & q_y & q_z \\ -q_x & -q_y & -q_z \end{vmatrix} = q_s \cdot (-q_x, -q_y, -q_z) + q_s \cdot (q_x, q_y, q_z) + ((\vec{i} \cdot q_y \cdot (-q_z)) + (\vec{j} \cdot q_z \cdot (-q_x)) + \\ &(\vec{k} \cdot q_x \cdot (-q_y))) - ((\vec{k} \cdot q_y \cdot (-q_x)) + (\vec{i} \cdot q_z \cdot (-q_y)) + (\vec{j} \cdot q_x \cdot (-q_z))) = \cancel{q_s \cdot q_x} - \cancel{q_s \cdot q_y} - \\ &\cancel{q_s \cdot q_z} + \cancel{q_s \cdot q_x} + \cancel{q_s \cdot q_y} + \cancel{q_s \cdot q_z} + (-q_y \cdot q_z \cdot \vec{i} - q_z \cdot q_x \cdot \vec{j} - q_x \cdot q_y \cdot \vec{k}) - (-q_y \cdot q_x \cdot \vec{k} - q_z \cdot q_y \cdot \\ &\vec{i} - q_x \cdot q_z \cdot \vec{j}) = -q_y \cdot q_z \cdot \vec{i} - q_z \cdot q_x \cdot \vec{j} - q_x \cdot q_y \cdot \vec{k} + q_z \cdot q_y \cdot \vec{i} + q_x \cdot q_z \cdot \vec{j} + q_y \cdot q_x \cdot \vec{k} = \\ &(0 \cdot \vec{i}, 0 \cdot \vec{j}, 0 \cdot \vec{k}) \end{aligned}$$

Nota: En este caso hemos podido simplificar los productos de cuaterniones, ya que como se ha explicado anteriormente, se cumple la propiedad conmutativa. Notar que este es un caso particular ya que, en general, el producto de cuaterniones no es conmutativo.

Al fin y al cabo un cuaternión puede representarse en forma de matriz de rotación, y es sabido que un producto de matrices no cumple la propiedad conmutativa.

Un ejemplo bastante visual en el que es fácil comprobar gráficamente que un producto de matrices no es conmutativo, es operar con la rotación y la traslación. No se trata exactamente de un producto de cuaterniones, ya que la traslación sería representada a partir de una matriz de traslación, no de rotación, pero sirve para comprobar la no conmutatividad entre producto de matrices.

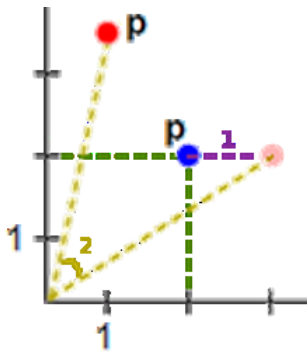


Figura 3.9: Realizamos primero traslación y después rotación [18].

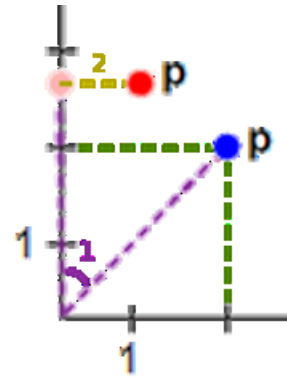
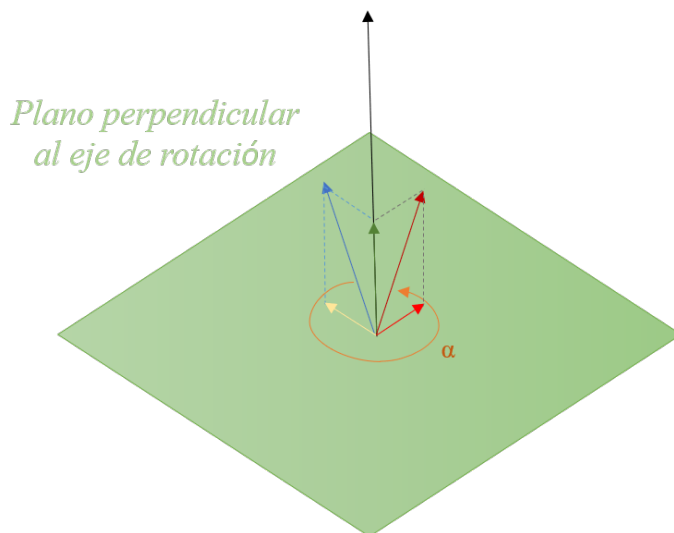


Figura 3.10: Realizamos primero rotación y después traslación [18].

Es sencillo comprobar visualmente cómo el punto rojo, que es el que representa la posición final, toma una posición distinta en cada una de las dos situaciones.

### Rotaciones en el espacio tridimensional



$$v = v_{\perp} + v_{\parallel}$$

La componente  $v_{\parallel}$  no cambia con la rotación.

$$v' = v'_{\perp} + v_{\parallel}$$

$\bar{u} \equiv$  Eje de rotación

$v \equiv$  Vector que se desea rotar

$v_{\perp} \equiv$  Proyección de  $v$  sobre el plano perpendicular al eje de rotación

$v_{\parallel} \equiv$  Proyección de  $v$  sobre el eje de rotación

$v' \equiv$  Resultado de rotar  $v$  un ángulo  $\alpha$  alrededor del eje de rotación  $\bar{u}$

$v'_{\perp} \equiv$  Resultado de rotar  $v_{\perp}$  un ángulo  $\alpha$  alrededor del eje de rotación  $\bar{u}$

Figura 3.11: Rotación de vectores en el espacio tridimensional

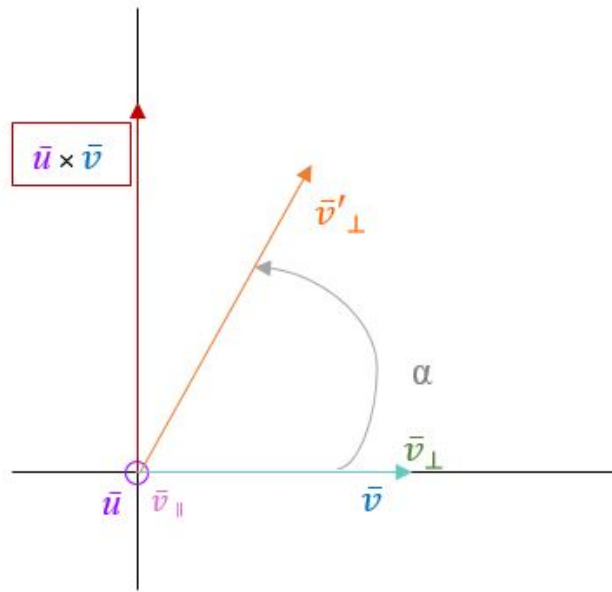


Figura 3.12: Rotación de un vector

Referencias a la ilustración superior:

$$\vec{v}'_{\perp} = |\vec{v}_{\perp}| \cdot \cos \alpha \cdot \frac{\vec{v}_{\perp}}{|\vec{v}_{\perp}|} + |\vec{v}_{\perp}| \cdot \sin \alpha \cdot \frac{\vec{u} \times \vec{v}}{|\vec{u} \times \vec{v}|} \implies \vec{v}'_{\perp} = \vec{v}_{\perp} \cdot \cos \alpha + (\vec{u} \times \vec{v}) \cdot \sin \alpha$$

Por tanto, dado que sabemos que  $v = v_{\perp} + v_{\parallel}$ , podemos deducir que:

$$\vec{v}' = \vec{v}_{\perp} \cdot \cos \alpha + (\vec{u} \times \vec{v}) \cdot \sin \alpha + \vec{v}_{\parallel}, \text{ ya que } \vec{v}_{\parallel} = \vec{v}'_{\parallel}$$

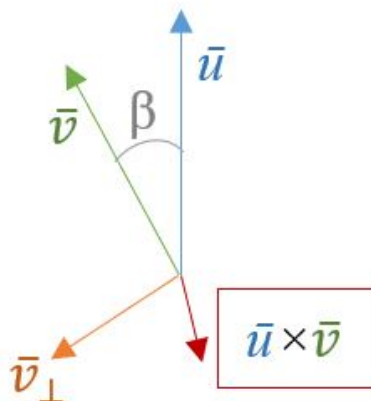


Figura 3.13: Rotación de un vector de un ángulo  $\beta$



Referencias a la ilustración superior:

$$|\vec{u} \times \vec{v}| = |\vec{u}| \cdot |\vec{v}| \cdot \sin \beta = |\vec{v}_{\perp}|$$

$$|\vec{u}| = 1$$

$$|\vec{v}_{\perp}| = |\vec{v}| \cdot \sin \beta$$

### Rotaciones mediante el producto de cuaterniones

Cuaternión que representa una rotación:

$$q = (\cos \frac{\alpha}{2}, \vec{u} \cdot \sin \frac{\alpha}{2})$$

$$\vec{u} \equiv \text{Eje de rotación}$$

$$\alpha \equiv \text{Ángulo de rotación}$$

$$|q| = 1$$

$$q^{-1} = \frac{q^*}{|q|} = q^* = (\cos \frac{\alpha}{2}, -\vec{u} \cdot \sin \frac{\alpha}{2})$$

Operación de rotación con cuaterniones:

Como primer elemento de la operación vamos a tener el cuaternión que deseamos rotar; el segundo elemento será el vector que queremos rotar (por lo que la parte escalar será 0), y como tercer elemento tendremos al cuaternión inverso del primero (es decir, la parte escalar será la misma, pero la parte vectorial estará multiplicada por el valor  $-1$ ).

Cuando utilizamos esto en una librería de programación gráfica, es lo que internamente realiza el método *rotate()*.

$$q \cdot \vec{v} \cdot q^{-1} = (\cos \frac{\alpha}{2}, \vec{u} \cdot \sin \frac{\alpha}{2}) \cdot (0, \vec{v}) \cdot (\cos \frac{\alpha}{2}, -\vec{u} \cdot \sin \frac{\alpha}{2})$$

$$q \cdot \vec{v} = (\cos \frac{\alpha}{2}, \vec{u} \cdot \sin \frac{\alpha}{2}) \cdot (0, \vec{v}) = (-\sin \frac{\alpha}{2} \cdot (\vec{u} \cdot \vec{v}), \vec{v} \cdot \cos \frac{\alpha}{2} + \sin \frac{\alpha}{2} \cdot (\vec{u} \times \vec{v}))$$

Multiplicamos el resultado que acabamos de obtener al realizar la operación de  $q \cdot \vec{v}$  por  $q^{-1}$ , que es lo que nos quedaba para completar la operación inicial:

$$\begin{aligned} q \cdot \vec{v} \cdot q^{-1} &= (-\sin \frac{\alpha}{2} \cdot (\vec{u} \cdot \vec{v}), \vec{v} \cdot \cos \frac{\alpha}{2} + \sin \frac{\alpha}{2} \cdot (\vec{u} \times \vec{v})) \cdot (\cos \frac{\alpha}{2}, -\vec{u} \cdot \sin \frac{\alpha}{2}) \implies \\ \implies q \cdot \vec{v} \cdot q^{-1} &= -(\sin \frac{\alpha}{2} \cdot \cos \frac{\alpha}{2} \cdot (\vec{u} \cdot \vec{v}) + \sin \frac{\alpha}{2} \cdot \cos \frac{\alpha}{2} \cdot (\vec{u} \cdot \vec{v}) + \sin \frac{\alpha}{2} \cdot \sin \frac{\alpha}{2} \cdot (\vec{u} \times \vec{v}) \cdot \\ \vec{u}, \sin \frac{\alpha}{2} \cdot \sin \frac{\alpha}{2} \cdot (\vec{u} \cdot \vec{v}) \cdot \vec{u} + \cos \frac{\alpha}{2} \cdot \cos \frac{\alpha}{2} \cdot \vec{v} + \sin \frac{\alpha}{2} \cdot \cos \frac{\alpha}{2} \cdot (\vec{u} \times \vec{v} - \sin \frac{\alpha}{2} \cdot \cos \frac{\alpha}{2} \cdot (\vec{v} \times \\ \vec{u} - \sin \frac{\alpha}{2} \cdot \sin \frac{\alpha}{2} \cdot ((\vec{u} \times \vec{v}) \times \vec{u}))) \implies \\ \implies q \cdot \vec{v} \cdot q^{-1} &= (0, \sin^2 \frac{\alpha}{2} \cdot \vec{v}_{\parallel} + \cos^2 \frac{\alpha}{2} \cdot \vec{v}_{\parallel} + \cos^2 \frac{\alpha}{2} \cdot \vec{v}_{\perp} + 2 \cdot \sin \frac{\alpha}{2} \cdot \cos \frac{\alpha}{2} \cdot (\vec{u} \times \vec{v}) - \\ \sin^2 \frac{\alpha}{2} \cdot \vec{v}_{\perp}) \implies \\ \implies q \cdot \vec{v} \cdot q^{-1} &= (0, \vec{v}_{\parallel} + \cos \alpha \cdot \vec{v}_{\perp} + \sin \alpha \cdot (\vec{u} \times \vec{v})) \end{aligned}$$

Para obtener la expresión final, hemos aplicado las propiedades del producto vectorial y escalar de vectores de  $R^3$  que aparecen a continuación de la Figura 3.14.

Como podemos observar, obtenemos el mismo resultado trabajando con el producto de cuaterniones que sin ellos, como hacíamos en el punto anterior:

- Rotación hallada con el método del primer apartado:

$$\vec{v}' = \vec{v}_{\perp} \cdot \cos \alpha + (\vec{u} \times \vec{v}) \cdot \sin \alpha + \vec{v}_{\parallel}$$

- Rotación hallada con el método de producto de cuaterniones:

$$\vec{v}' = \vec{v}_{\parallel} + \cos \alpha \cdot \vec{v}_{\perp} + \sin \alpha \cdot (\vec{u} \times \vec{v})$$

Observamos que, aunque nos han salido las componentes en otro orden, los resultados obtenidos son equivalentes.

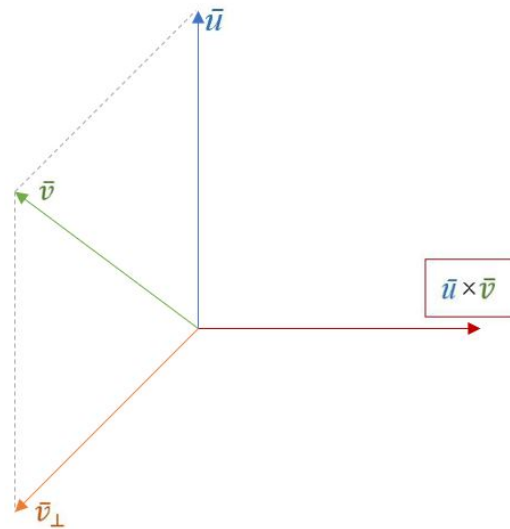


Figura 3.14: Representación de los ejes con los que se cumplen una serie de propiedades más abajo mostradas

Referencias a la ilustración superior:

$$(\vec{u} \times \vec{v}) \cdot \vec{u} = 0$$

$$(\vec{u} \cdot \vec{v}) = |\vec{u}| \cdot |\vec{v}| \cdot \cos \beta = |\vec{v}_{\parallel}|$$

$$(\vec{u} \cdot \vec{v}) \cdot \vec{u} = |\vec{v}_{\parallel}| \cdot \vec{u} = \vec{v}_{\parallel}$$

$$\vec{v} = \vec{v}_{\parallel} + \vec{v}_{\perp}$$

$$(\vec{u} \times \vec{v}) = (\vec{u} \times \vec{v}_{\parallel}) + (\vec{u} \times \vec{v}_{\perp}) = \vec{u} \times \vec{v}_{\perp}$$

$$(\vec{u} \times \vec{v}) \times \vec{u} = (\vec{u} \times \vec{v}_{\perp}) \times \vec{u} = \vec{v}_{\perp}$$

## 3.2. Visualización

Para poder mostrar en pantalla los ángulos de apertura calculados, mediante la interfaz gráfica de usuario de la aplicación, hemos utilizado la librería gráfica Ogre3D.

En los siguientes apartados se explica con detalle el proceso y los requerimientos del sistema para visualizar ángulos.

### 3.2.1. Librería Ogre3D



Figura 3.15: Logo Ogre3D [19]

Ogre3D es una librería gráfica 3D de código abierto (open-source), ampliamente usada en proyectos de áreas tan diversas como juegos, simuladores, software didáctico, visualización científica, etc [20].

Su uso, ha permitido mostrar de una forma más simple y sencilla las tres dimensiones de una figura que representa todas las articulaciones del cuerpo humano. Como ya se ha comentado, en este proyecto, aquéllas que presentan mayor interés son los hombros y los codos.

Hablando en términos un poco más técnicos, podríamos decir que el cuadro en el que vemos representado al esqueleto es un *overlay* donde hemos situado dicho objeto, al cual está enfocando la cámara. La cámara es otro objeto que, igual que en la vida real, enfoca al sujeto que deseamos ver en pantalla. Si se quisiera visualizar otro objeto que estuviese colocado en otra parte o posición distinta en el espacio, tendríamos que moverla para conseguirlo.

Por otra parte, el esqueleto viene dado a partir de un fichero de extensión *.skeleton*. Éste consiste en un conjunto de objetos de tipo *Bone* para poder animar el objeto 3D, el cual está generado gracias a otro fichero de extensión *.mesh*. El *Mesh* conforma la geometría de un objeto visible en una aplicación 3D, como podría ser la que se presenta en esta memoria.

El fichero que corresponde al *Skeleton* también puede estar almacenado con una extensión XML, en el que vemos representados todos los *Bones* con sus respectivos nombres y posiciones, las cuales, ya que están en un formato XML, podemos editar a nuestro antojo, pero el fichero que utilizarán las aplicaciones de Ogre a la hora de ejecutarlo serán aquellas que tengan extensión *.skeleton*.

El *Mesh* conforma la geometría de un objeto 3D visible en la aplicación. Un *mesh* consiste en una serie de triángulos, que también puede contener un conjunto de líneas y puntos.

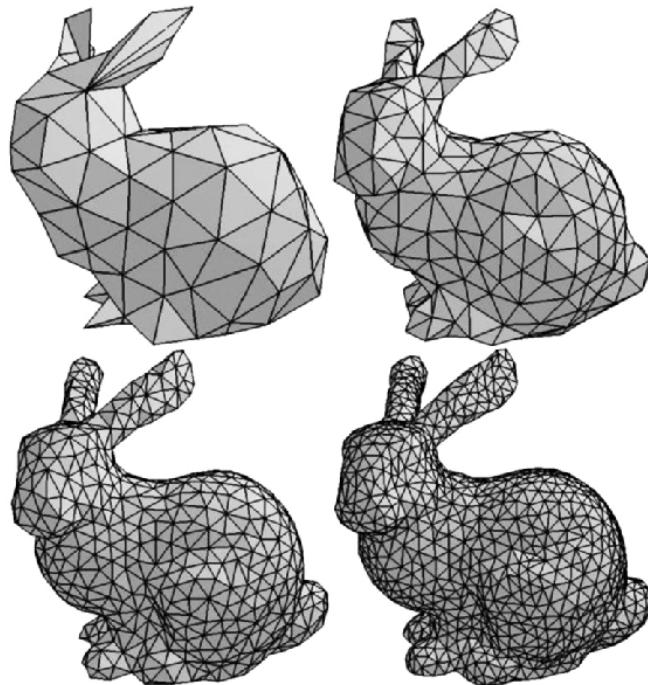


Figura 3.16: Ejemplos de *mesh* [21]

Este tipo de objeto puede estar, al igual que el *skeleton*, almacenado en dos extensiones diferentes. En este caso serán las extensiones *.xml*, y *.mesh*. En el primer caso, se trata de un fichero XML que podemos editar sin problema, pero las aplicaciones de Ogre no serán capaces de cargarlo. En el segundo caso, estarán en un formato binario, y las aplicaciones de Ogre podrán abrirlo y cargarlo sin ningún problema.

Para cargar un *mesh* en una escena, es necesario crear primero, a partir del *mesh*, una entidad o *Entity*, y gracias a ello, podemos asignar diferentes materiales a cada una de las entidades. Una entidad será una instancia de un *MovableObject* dentro de la escena que estemos visualizando en nuestra pantalla, es decir, que podría ser, por ejemplo, una persona, un perro o un coche y no tiene por qué tener una posición fija en la escena.

Si como se menciona en el párrafo anterior, quisiéramos dar distintas texturas y colores a nuestro objeto, podríamos generar nosotros mismos un fichero de extensión *.material*, en el que, a cada elemento del *.mesh*, podemos asignarle la apariencia que más nos guste. Notar que, para hacer dotar a un objeto de dicha apariencia, tenemos que tener esos colores y texturas almacenados en la carpeta correspondiente del proyecto de QT en el que nos encontramos, ya que éstos también serán ficheros y para poder utilizarlos, el programa tendrá que ser capaz de encontrarlos.

A la hora de utilizar esta librería, ha habido que tener especial cuidado en la comprensión de la posición de los tres ejes de coordenadas para la representación tridimensional. Esto es debido a que los ejes coordenados no son los mismos para Ogre, para el sistema global y para las posiciones iniciales relativas de cada una de las partes del esqueleto a representar en pantalla.

En la siguiente figura podemos observar la diferencia que existe en la disposición de los ejes de los dos sistemas de referencia.

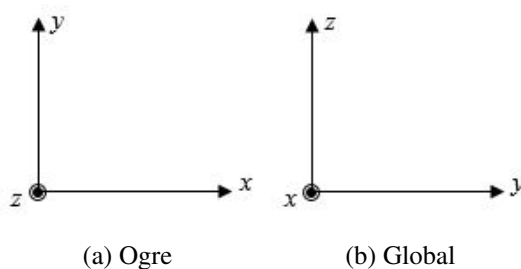


Figura 3.17: Disposición de los ejes en cada caso

La posición inicial que adopta nuestro esqueleto, es en forma de T (con los brazos extendidos), de manera que los ejes se disponen como en la figura 3.18, teniendo en cuenta que el eje  $y$  siempre va a estar colocado sobre el eje longitudinal del hueso:

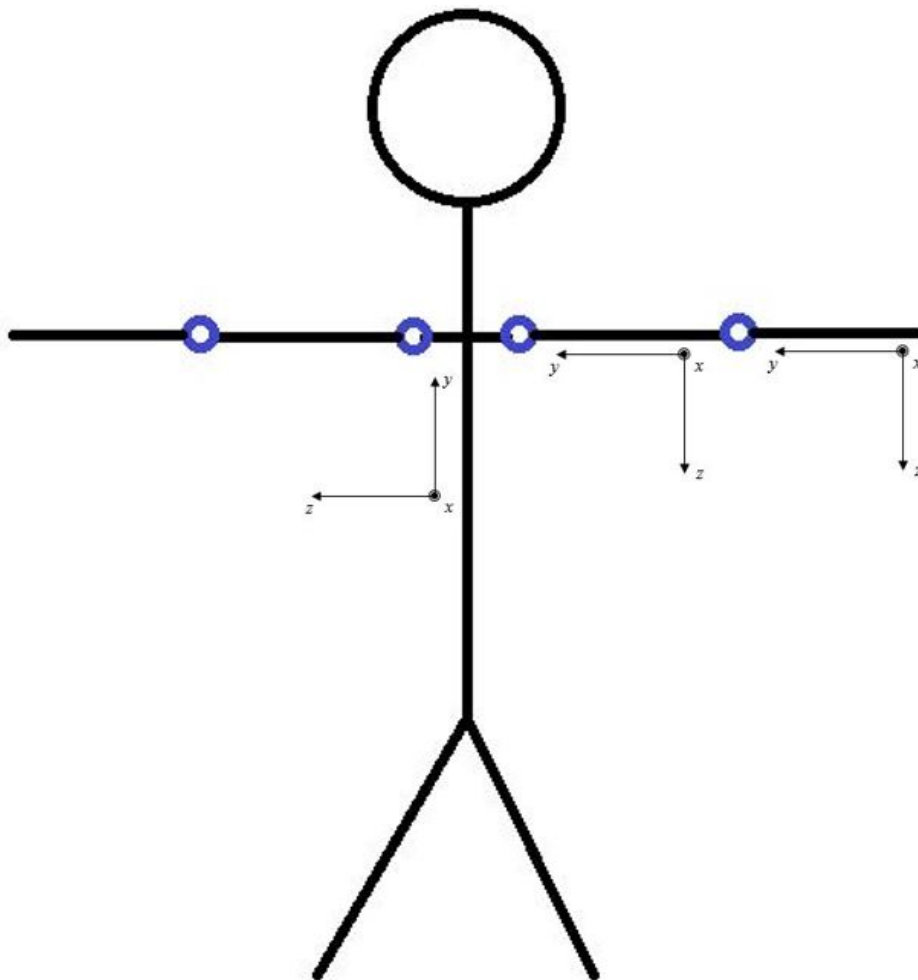


Figura 3.18: Representación de los ejes sobre las partes de las articulaciones superiores en la posición inicial del esqueleto en el programa

Si consideramos como posición final la del sujeto con los brazos hacia abajo, los ejes se desplazan junto con los huesos. Al realizar este movimiento el pecho mantiene la misma posición y así nuestro eje  $y$  seguirá estando sobre el eje longitudinal del hueso correspondiente. Es decir, la posición de los ejes es la misma que en la Figura 3.18:

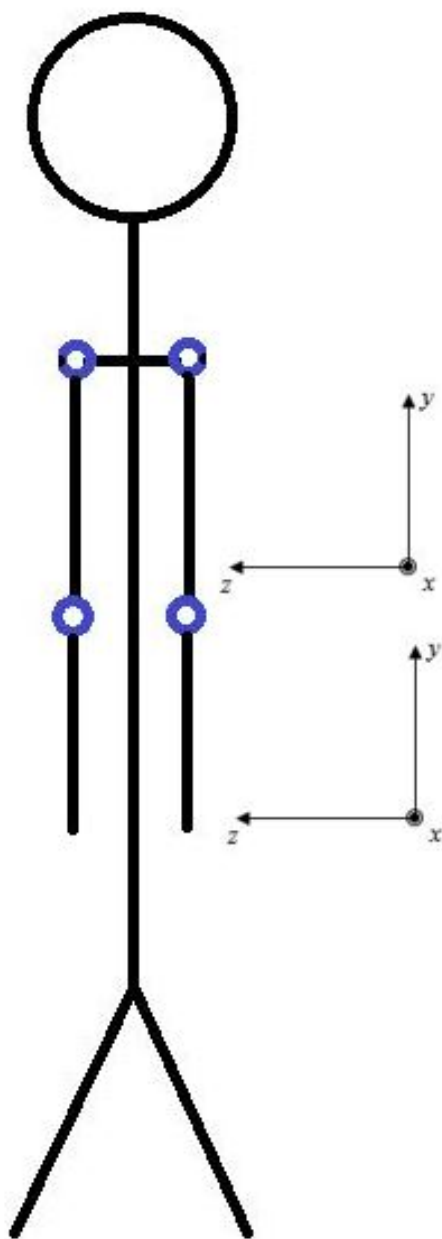


Figura 3.19: Representación de los ejes sobre las partes de las articulaciones superiores en un ejemplo de posición del paciente

Lo que se necesita saber es la diferencia de posición que existe del brazo y antebrazo, de la posición real con respecto a la inicial que se nos muestra al abrir la aplicación, la cual podremos hallar gracias a que previamente son conocidas las posiciones inicial (es la posición que toma el esqueleto en el programa, con los brazos en T, como se ha citado anteriormente), y



real (la que toma el paciente en realidad, que en este caso, hemos cogido como ejemplo que sea con los brazos hacia abajo). Podemos representarlo esquemáticamente de la siguiente forma:

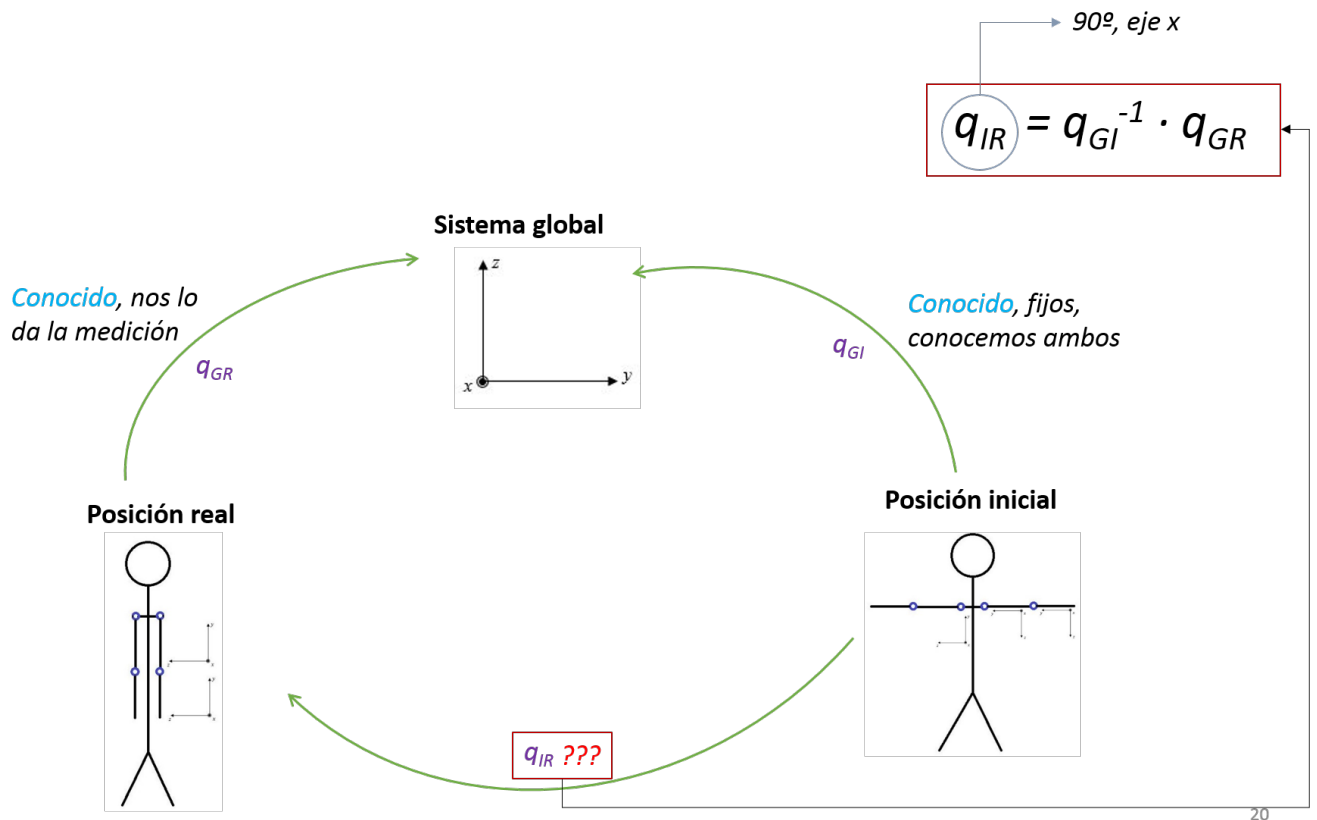


Figura 3.20: Cuaterniones que conocemos y cuaternión que necesitamos hallar

Podríamos decir que tanto  $q_{GR}$  como  $q_{IR}$  dependen del tiempo, ya que en ellos interviene la posición real, y ésta irá cambiando con el transcurso del tiempo con los movimientos que realiza el paciente.

Tenemos que llevar a cabo un cambio de coordenadas: de las del propio sensor, a las del sistema global ( $q_{GS}(t)$ ). Para ello, debemos rotar el sistema  $S$  para que coincida con el sistema  $G$ , alrededor de un eje de rotación  $u$  y un giro de  $\alpha$  grados; o lo que vendría a ser equivalente, rotar  $G$  para que coincida con  $S$ , un ángulo de  $-\alpha$ . Pero si queremos llevar a cabo un cambio de coordenadas de un vector de un sistema a otro, sería suficiente rotar el vector en el sistema original hasta posicionarlo en el lugar donde quedaría si girásemos el sistema completo.

Lo podemos ver más claro en una imagen:

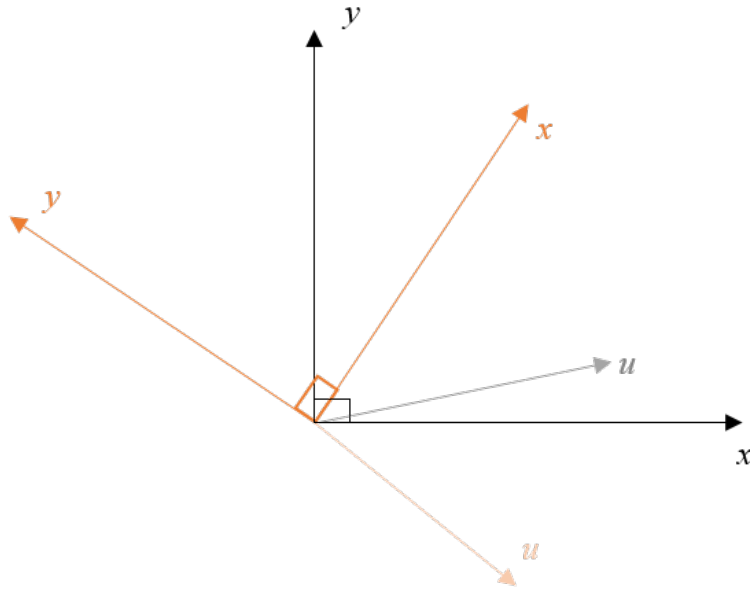


Figura 3.21: Resultado de posición del vector  $u$  al cambiarlo de sistema de coordenadas

Para que el sistema  $G$  coincida con  $I$  (el correspondiente a la posición inicial del esqueleto), debemos realizar una rotación de  $180^\circ$  alrededor del eje  $x$ , de manera que nuestro eje de rotación  $u$  sería el vector  $(1, 0, 0)$ , y  $\alpha = \pi$  rad.

Para poder expresar el eje de rotación en el sistema de coordenadas de  $G$ , tenemos que

$$q_{IR}(t) = q_{GI}^{-1} \cdot q_{GR}(t)$$

Debido a que el pecho, que es jerárquicamente el padre del brazo, gira, éste también lo hará con él, de manera que la rotación del pecho hará que a la posición inicial la denotemos, en lugar de como  $I$ , como  $I'$ , ya que ésta habrá cambiado, aunque no lo haya hecho en sí misma.

Para que los ejes de coordenadas del padre (pecho) coincidan con los ejes de coordenadas de la posición inicial del brazo, debemos realizar una rotación de  $90^\circ$  alrededor del eje  $x$ , es decir, que nuestro eje de rotación será  $u = (1, 0, 0)$ , y el ángulo será  $\alpha = \frac{\pi}{2}$ .

$$q_{PQ}(t) \equiv \text{Rotación del padre}$$

$$q_{II'}(t) = q_{PI}^{-1} \cdot q_{PQ}(t) \cdot q_{PI} \equiv \text{Rotación de la posición inicial}$$

$$q_{I'R}(t) = q_{II'}^{-1}(t) \cdot q_{IR}(t) \equiv \text{Rotación a aplicar al hijo debido a la rotación del padre}$$

Siendo:

$P$  = Posición inicial del padre

$Q$  = Posición real del padre

$I'$  = Posición inicial del segmento tras la rotación del padre

Para que dicha rotación se lleve a cabo correctamente, en el caso del brazo, debemos realizar lo siguiente:

- Obtener la diferencia de posición inicial a posición real del pecho, en el sistema global:

$$q_{I_c R_c} = q_{G I_c}^{-1} \cdot q_{G R_c}(t)$$

- Calcular la diferencia de posición inicial a posición real del brazo, también en el sistema

$$\text{global: } q_{I_u R_u} = q_{G I_u}^{-1} \cdot q_{G R_u}(t)$$

- Hallar la diferencia de posición inicial a posición inicial' provocada por la rotación del

$$\text{pecho: } q_{I_u I_u'} = q_{I_c I_u}^{-1} \cdot q_{I_c R_c}(t) \cdot q_{I_c I_u} = q_{I_u I_c} \cdot q_{I_c R_c}(t) \cdot q_{I_c I_u} = q_{I_u R_c} \cdot q_{I_c I_u}$$

Esto se debe a que se puede dar la siguiente situación, en la que vemos cómo si se mueve el pecho, el brazo, debido a la jerarquía seguida, se mueve con él:

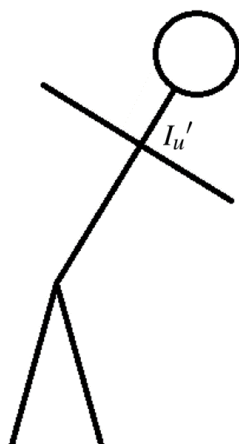


Figura 3.22: Rotación del brazo debido a la rotación del pecho

- Aplicar el cuaternión para que el brazo se mueva como debe. Este cuaternión es:

$$q_{I_u'R_u}(t) = q_{I_u I_u'}^{-1} \cdot q_{I_u R_u}(t)$$

Para que se represente tal y como es en la realidad, debemos tener cuidado para que todo este representado en un mismo sistema de coordenadas. En este caso, será en el sistema global, entonces también hay que expresar el eje de rotación en dicho sistema. Finalmente, debemos trasladar todo realizando un cambio de ejes para que coincidan con los de Ogre, de manera que se proyecte en pantalla lo que queremos.

### 3.2.2. Elementos del interfaz

El interfaz de la aplicación consta de una ventana, la cual consta de diferentes partes. Por un lado, la que nos permite elegir si queremos visualizar un ángulo de apertura perteneciente al hombro o al codo, y dentro de la que hayamos escogido, qué tipo de movimiento queremos estudiar en ese momento.

A su derecha, la parte que más llama la atención del interfaz de la aplicación. Ésta es

## CAPÍTULO 3. METODOLOGÍA

la imagen del esqueleto en 3D y que representa al sujeto que tenga colocados los sensores inerciales en los lugares correspondientes, para estudiar sus movimientos. Una vez elegido el movimiento que queremos estudiar, podemos darle al icono que muestra una flecha blanca con un fondo circular verde, que nos indica que podemos iniciar la medición. Dicho esqueleto mostrará en tiempo real los movimientos que realiza el sujeto.

Por otro lado, en la parte izquierda de la ventana, tenemos un icono que muestra una pequeña esfera del mundo, el cual utilizamos para determinar el sistema global fijo que van a utilizar los sensores inerciales para llevar a cabo las mediciones. Otro icono importante es el que muestra dos herramientas, que utilizaremos para llevar a cabo la calibración de los sensores. Si estamos ya midiendo y lo que queremos es finalizar la medición, habrá que pulsar un icono representado como un cuadro azul que se mostrará en este lado de la pantalla.

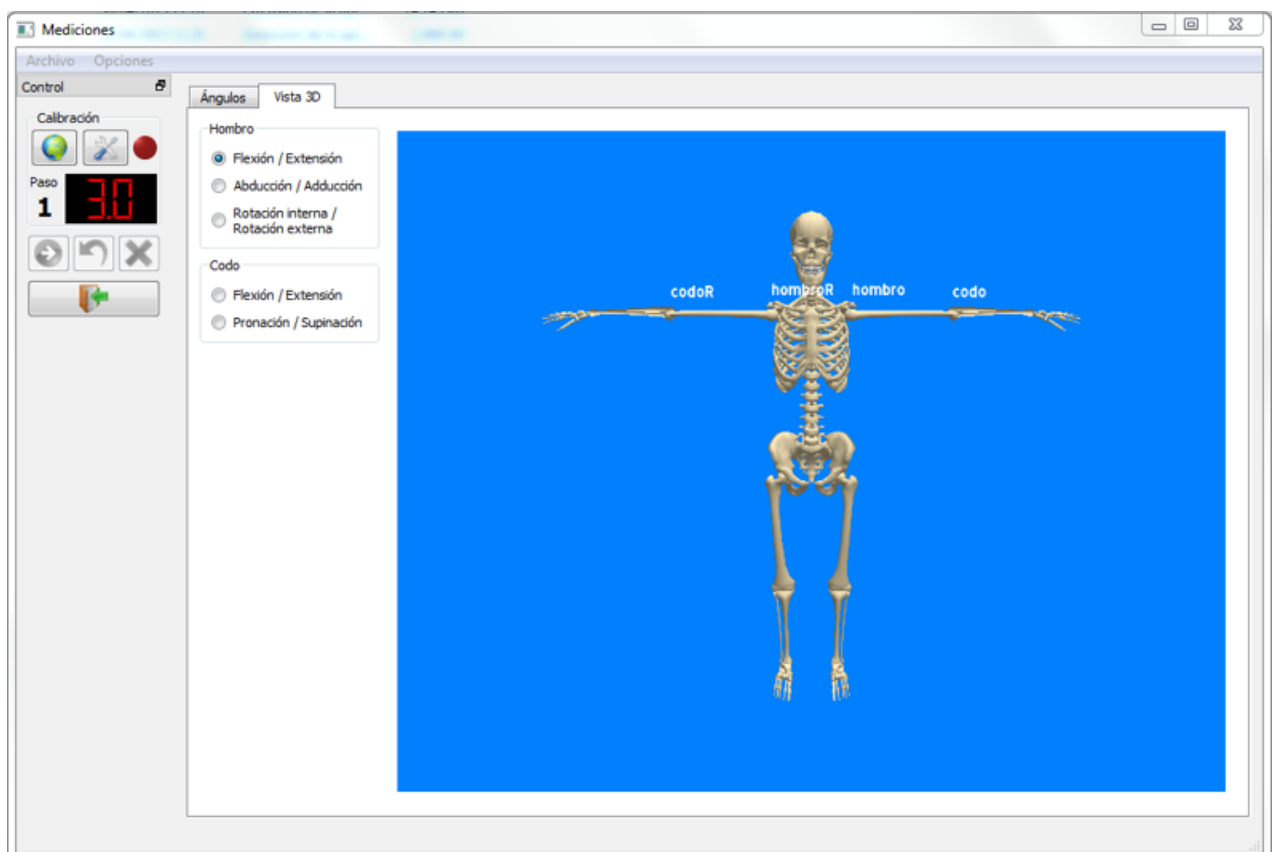


Figura 3.23: Interfaz de la aplicación

# 4

## Resultados y análisis

### 4.1. Resultados

Como resultado, podemos decir que se ha desarrollado una herramienta que permite visualizar, de forma precisa y objetiva, cuántos grados ha sido capaz un sujeto de abrir o cerrar el hombro o el codo. Desde un punto de vista clínico, esta herramienta, aplicada por ejemplo en el campo de la rehabilitación, permite tener datos de las diferentes sesiones de rehabilitación de manera personalizada. Así, el terapeuta puede usar datos objetivos para ver la evolución de sus pacientes.

## CAPÍTULO 4. RESULTADOS Y ANÁLISIS

---

Además, es una herramienta mucho más cómoda y sencilla, tanto para el paciente como para el especialista encargado de realizar las mediciones, ya que únicamente hay que colocar los sensores inerciales y comenzar a medir pulsando el botón *start* de nuestra interfaz de usuario.



Figura 4.1: Funcionamiento del programa: Paciente con sensores colocados

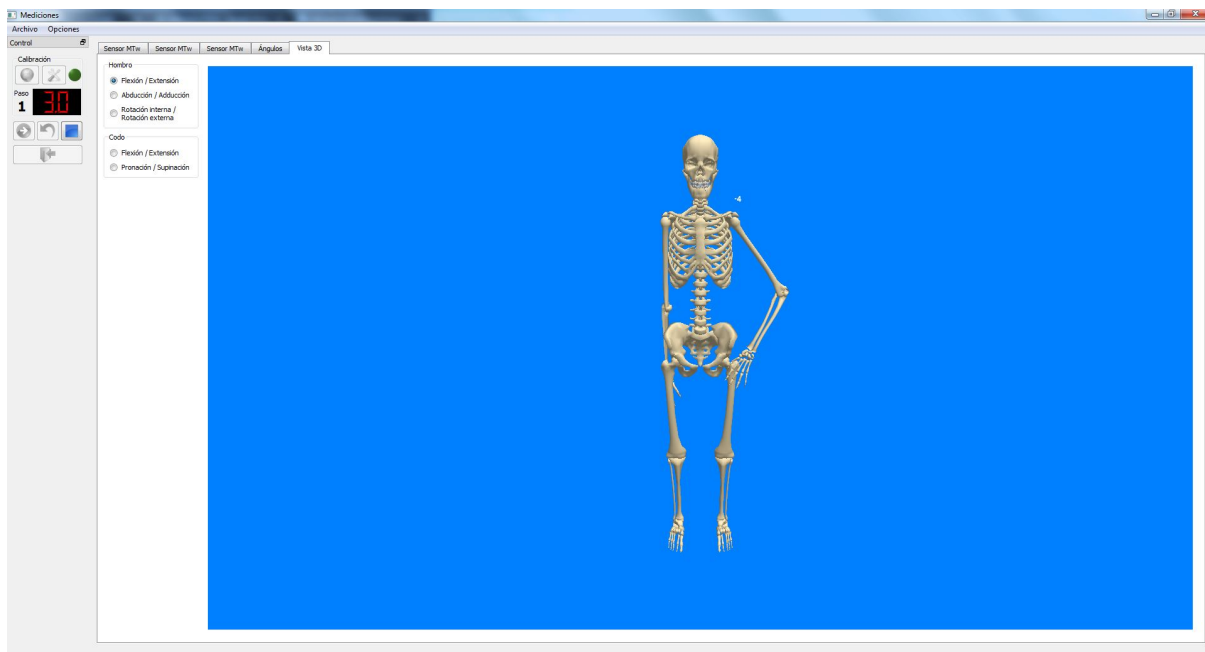


Figura 4.2: Funcionamiento del programa: Resultado de reproducción de movimiento obtenido en pantalla

## CAPÍTULO 4. RESULTADOS Y ANÁLISIS

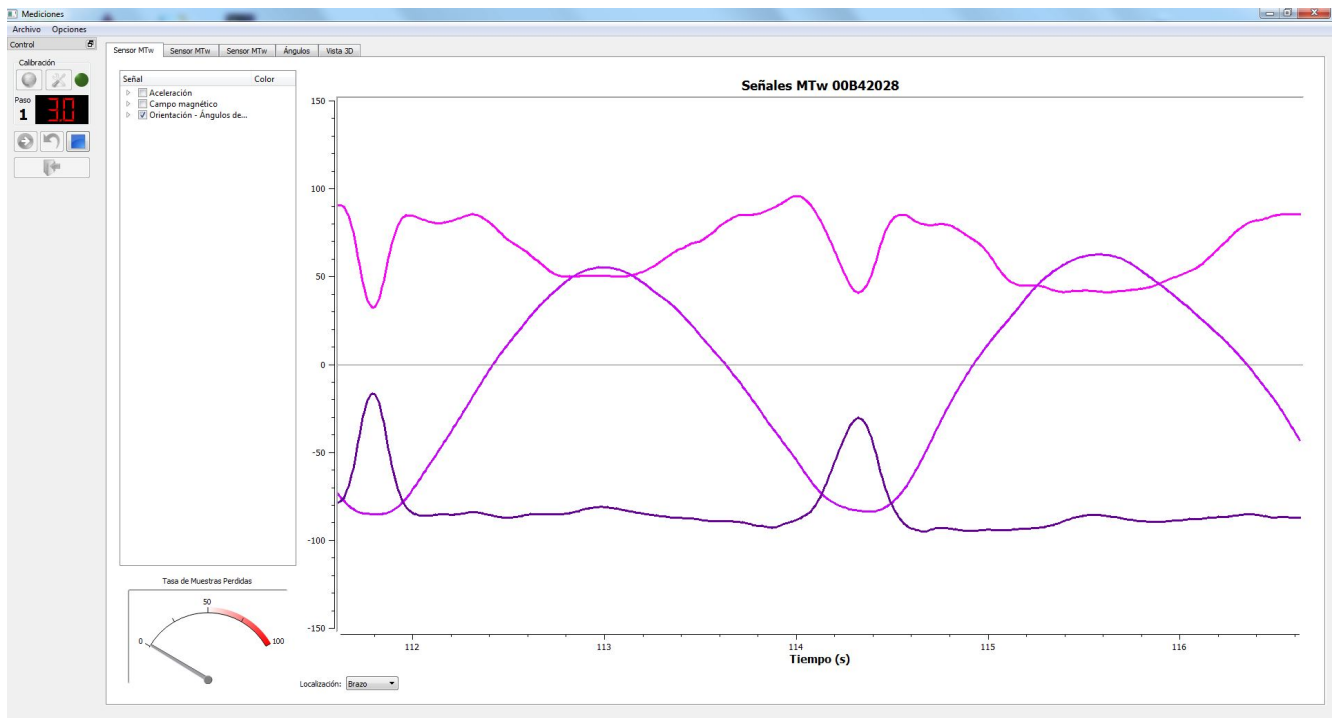


Figura 4.3: Señales de apertura de ángulo del brazo realizando el movimiento de flexión

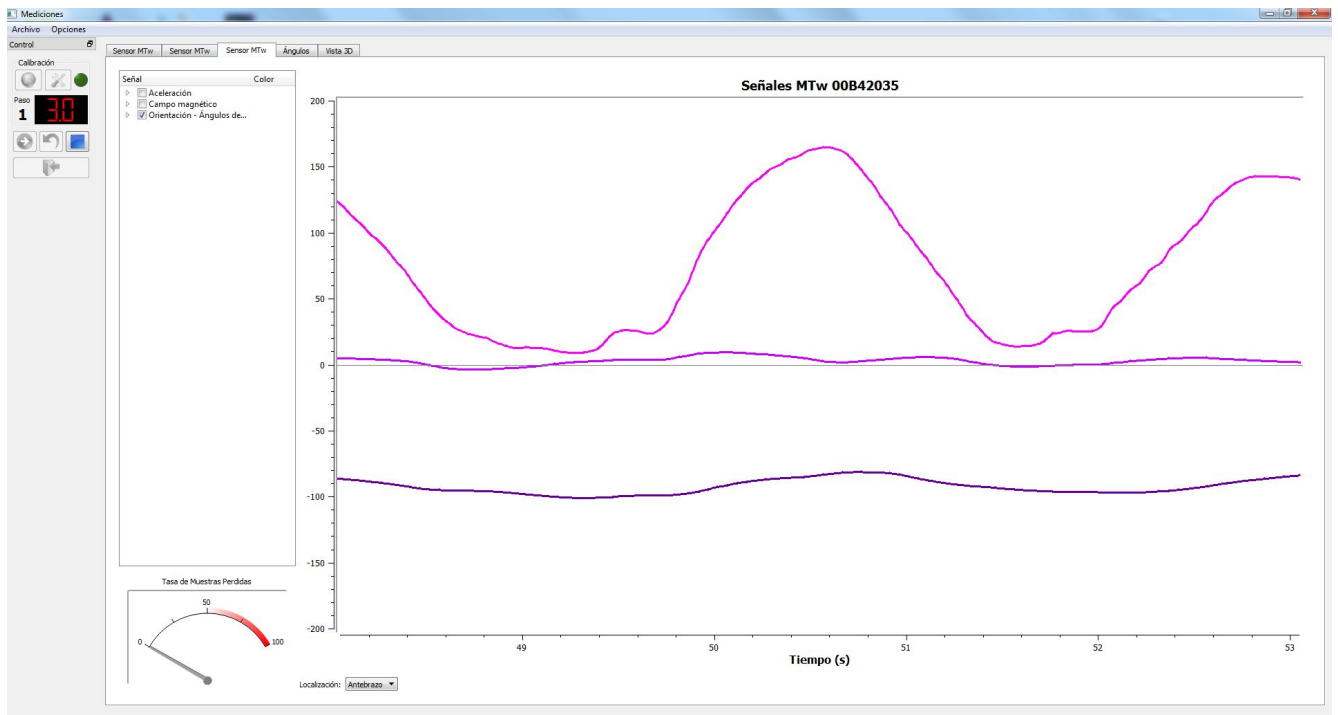


Figura 4.4: Señales de apertura de ángulo del brazo realizando el movimiento de pronación/supinación





# 5

## Conclusiones y líneas futuras

### 5.1. Conclusiones

Una vez finalizado el desarrollo del proyecto, podemos concluir que:

- Es posible medir rangos articulares en determinados segmentos de las extremidades superiores, usando sensores inerciales.
- La interface desarrollada es una herramienta de fácil manejo para visualizar en tiempo real los movimientos de los sujetos.

- El uso de la herramienta desarrollada permite obtener datos objetivos de la evolución de la rehabilitación o de la recuperación de la función de extremidades superiores.
- El interface y el software de medida desarrollado puede ser usado en el ámbito clínico y ambulatorio para realizar valoraciones funcionales en el tiempo y de forma precisa.

se trata de una herramienta muy útil y de sencilla utilización, que puede ayudar al mejor desarrollo de muchas personas que necesitan rehabilitación, ya que gracias a su cómoda y sencilla utilización, podemos determinar qué ejercicios son los que más y más rápido hacen que el paciente vaya mejorando en su día a día.

### **5.2. Líneas futuras**

Es una herramienta con un gran potencial, y para seguir desarrollando la aplicación, sería de utilidad asignar el campo de visualización de apertura de las articulaciones de otras partes del cuerpo como muñecas, caderas o rodillas, ya que también son zonas igualmente propensas a padecer algún tipo de lesión que necesite de rehabilitación para recuperar su función.

Este proyecto, de momento, está desarrollado únicamente para su utilización en un ordenador Windows, pero en un futuro podría valorarse su desarrollo para otros sistemas operativos como Android o iOS, para una utilización más cómoda de la herramienta, de manera que fuera más portable gracias a las tablets.

También se ha valorado cambiar la figura del esqueleto por otra figura más atractiva, ya que puede no resultar agradable para algunos pacientes.



# Bibliografía

- [1] *Quirófano de hoy en día*. URL: <https://goo.gl/images/tc6G53>.
- [2] *Medicina en formato digital*. URL: <https://goo.gl/images/jyShb1>.
- [3] *Definición de ergonometría*. URL: <http://www.iqb.es/diccio/e/er.htm>.
- [4] *Definición de goniometría*. URL: <https://es.scribd.com/doc/54012356/Definicion-de-goniometria>.
- [5] *Definición de kinesiterapia*. URL: <http://dle.rae.es/?id=MhV6JHw>.
- [6] *Goniometría dentro de la kinesiterapia*. URL: <http://www.medisoftware.com.ve/Download/Normas/libro-goniometria.pdf>.
- [7] *Medición de ángulos articulares a partir de ejes longitudinales*. URL: <https://goo.gl/images/Yx1pcX>.
- [8] *Ilustraciones de ejes horizontal, frontal y vertical*. URL: <https://www.ugr.es/~marroyo/docs/temas/pdf/4%20MOVILIDAD%2030%20SEPT.pdf>.
- [9] *Partes para la utilización de un goniómetro*. URL: <https://goo.gl/images/huNEmy>.
- [10] *Definición de flexómetro de Leighton y de electrogoniómetro*. URL: [http://ww2.educarchile.cl/portal.herramientas/sitios\\_educativos/educ\\_fisica/pags/aptitud\\_fisica/movielas\\_depor/cont2\\_evalu.html](http://ww2.educarchile.cl/portal.herramientas/sitios_educativos/educ_fisica/pags/aptitud_fisica/movielas_depor/cont2_evalu.html).

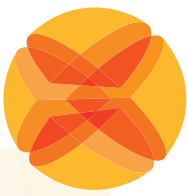
- [11] *Electrogoniómetro*. URL: <http://miofeedbackenkinesio.blogspot.com/2009/05/miofeedback.html>.
- [12] *Imagen de un flexómetro*. URL: <https://goo.gl/images/8MjJxL>.
- [13] *Kinovea*. URL: <https://www.youtube.com/watch?v=GzDl6behj7Q>.
- [14] *Definición QtCreator*. QT. URL: <https://www.qt.io/qt-features-libraries-apis-tools-and-ide/#ide>.
- [15] Itziar Uzqueda. *Definición de sensor inercial*.
- [16] *Movimiento de flexión y extensión del codo*. URL: <https://goo.gl/images/2KYdGi>.
- [17] *Movimiento de pronación y supinación del codo*. URL: <https://goo.gl/images/32cv9J>.
- [18] Benoît Bossavit. *Uso de cuaterniones*.
- [19] *Ogre3D Logo*. URL: <https://goo.gl/images/aUpWQ5>.
- [20] *Definición de Ogre*. Ogre3D. URL: <https://www.ogre3d.org/>.
- [21] *Ejemplo mesh*. URL: <https://goo.gl/images/AhSTqJ>.



# 6

## Anexos





**xsens**

# MTw Awinda

WIRELESS MOTION TRACKER



Highly accurate, small and lightweight 3D human wireless motion tracker. Used in a body area network, these portable and unobtrusive wireless trackers are ideal for developing customized wearable solutions.

## Specifications MTw™ Awinda™

Tracker placement	Easy fastening with Velcro straps		
Internal sampling rate	1000 Hz		
Latency	30 ms		
Buffer time (retransmissions)	10 s		
Battery life (continuous use)	6 Hours		
Dimensions Tracker	47 x 30 x 13 mm (1.85 x 1.18 x 0.51 in)		
Weight	16 g (0.56 oz.)		
Operating temperature range	0°C - 50°C		
<b>Communication</b>			
Range	Awinda station	Awinda dongle	
Open space	Up to 50 m (165 ft.)	Up to 20 m (65 ft.)	
Office space	Up to 20 m (65 ft.)	Up to 10 m (33 ft.)	
Wireless protocol	Xsens patented Awinda protocol <sup>1</sup>		
Receiver	Awinda Station / Awinda Dongle		
<b>Orientation</b>			
Static Accuracy (Roll/Pitch)	0.5 deg RMS		
Static Accuracy (Heading)	1 deg RMS		
Dynamic Accuracy (Roll/Pitch)	0.75 deg RMS		
Dynamic Accuracy (Heading)	1.5 deg RMS		
<b>Tracker components</b>			
	Angular velocity	Acceleration	Magnetic field
Dimensions	3 axes	3 axes	3 axes
Full scale	± 2000 deg/s	± 160 m/s <sup>2</sup>	± 1.9 Gauss

<sup>1</sup> The patented Awinda protocol ensures highly accurate time synchronized data sampling (within ≤10 μs) in all connected MTw's, which is essential for accurate joint angles.

