

Trabajo Final de Grado

Ingeniería de Telecomunicación

Creación con datos de IMU (Inertial Measurement Unit) de un modelo para la técnica de piragüismo aplicando inteligencia artificial

Autor: Alejandro López de la Herranz

Tutor: Antonio Jesús Sierra Collado

Dpto. Ingeniería telemática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2022



Trabajo Final de Grado
Ingeniería de Telecomunicación

Creación con datos de IMU (Inertial Measurement Unit) de un modelo para la técnica de piragüismo aplicando inteligencia artificial

Autor:

Alejandro López de la Herranz

Tutor:

Antonio Jesús Sierra Collado

Dpto. de Ingeniería Telemática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2022

Proyecto Fin de Carrera: Creación con datos de IMU (Inertial Measurement Unit) de un modelo para la técnica de piragüismo aplicando inteligencia artificial

Autor: Alejandro López de la Herranz

Tutor: Antonio Jesús Sierra Collado

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2022

El Secretario del Tribunal

A mi familia

A mis maestros

Agradecimientos

Este trabajo he podido realizarlo gracias a la ayuda de mis padres, Pascual y Rafaela. Que para empezar me han dado la vida. Y me han dejado hacer la carrera que yo quisiera sin tener que preocuparme de nada. Y para finalizar me han tenido que soportar impotentes mi angustia que he sufrido durante toda la carrera. Ellos me han ayudado incluso a coser los bolsillos que he necesitado para el trabajo. Y le doy gracias a mi hermano Rubén por haberme apoyado y comprendido tanto.

Le tengo que agradecer a todos los compañeros de trabajo que he ido conociendo. He tenido la suerte de que mi clase estaba lleno de personas responsables, profesionales y amables. Que me han ayudado siempre que he pedido su ayuda. Un agradecimiento especial para mis compañeros de prácticas y a todos los que me han ayudado a estudiar

Para finalizar, darles las gracias a todos los amigos. Sobre todo, los que he ido haciendo durante la carrera, con los que qué he sentido una increíble conexión. Ellos se preguntaban y pensaban cosas bastante similares a las mías. Pero siempre dándole su toque personal, como yo supongo. Cuando yo estaba preocupado por los estudios ellos me tranquilizaban. Sobre todo, cuando nos juntábamos a comer en las piscinas y hablábamos de nuestras cosas. Y me he sentido muy querido y entendido por todos ellos. Y habéis hecho mi vida mejor en tantos aspectos que no os lo podéis ni siquiera imaginar. Todos ellos son personas que se merecen lo mejor en la vida. Espero que ellos me acompañen durante mucho tiempo en mi vida.

Además, debo agradecerle a mi club de piragüismo y a mis compañeros de este deporte. Ellos me han apoyado en todo momento este proyecto y me han ayudado a llevarlo a cabo. Especialmente le debo las gracias al director del club, a los entrenadores y a los participantes de las pruebas.

Alejandro López de la Herranz

Sevilla, 2022

Resumen

La técnica del piragüismo es un elemento fundamental en competiciones de este deporte. Ya que, permite alcanzar más velocidad con menos esfuerzo. Debido a que la fuerza del movimiento de paleo se realiza con todo el cuerpo. Por lo tanto, esto también hace mejorar la resistencia en dicha salida. Es por ello por lo que este trabajo saca un modelo de dichos movimientos a partir de una salida. Este permitirá ver al piragüista los errores que pueda estar cometiendo durante su palada. De forma que este pueda ir corrigiendo y adaptando la técnica que se considere oportuna. El trabajo se centrará en el tren superior, ya que, este deja ver el comportamiento más básico de la palada.

Para ello, el primer paso es sacar los datos de la palada del piragüista. Con ese objetivo se ha creado un sistema de sensores que se puede equipar en el cuerpo del deportista. Que tengan como función medir los ángulos que forman las articulaciones de los deportistas durante el paleo. Para ello, se colocan los sensores fijos a una parte del cuerpo que permita medir dichas articulaciones. La arquitectura del sistema consta de tres dispositivos principales. Estos son los **IMUs** (*Inertial Measurement Unit*), los microcontroladores y una Raspberry pi. A parte de estos dispositivos se ha hecho uso de un dispositivo móvil que funciona como interfaz para el usuario. Además, se ha hecho uso fundamentalmente de dos protocolos **I2C** (*Inter-Integrated Circuit*) y Bluetooth. Estos se usan para la comunicación entre los diferentes dispositivos. Además, se ha usado el protocolo wi-fi para la comunicación con el usuario.

El microcontrolador utiliza el protocolo **I2C** para controlar el funcionamiento del **IMU**. A su vez, la Raspberry pi iniciará una comunicación Bluetooth con los microcontroladores. Que será usado para gestionar y controlar a todos los microcontroladores a través de la Raspberry pi. Por otro lado, se conectará la Raspberry pi al **AP** (*Access point*) creado a través del dispositivo móvil. Se hará uso de la aplicación **SSH** (*Secure Shell*) como interfaz con el usuario de la aplicación.

Este sistema permitirá enviar los datos obtenidos a través de los diversas **IMUs** a la Raspberry pi. Los ángulos de la IMU son calculados a partir del giroscopio y el acelerómetro, que son sensores del **IMU**. Para que estos funcionen deben de ser calibrados anteriormente. Estos serán dados en forma de cuaternión. Los valores calculados serán muestreados y enviados a la Raspberry pi a través del sistema descrito anteriormente. Allí estos serán almacenados dentro de una **BBDD** (base de datos). Se asocia dichos datos a los datos de la salida y del **IMU** correspondiente.

Una vez obtenidos los datos, se saca la información de las articulaciones a partir de estos. Para ello se tratan y fusionan datos de los diferentes sensores. De manera que a partir de estos se pueda sacar el ángulo de giro de las articulaciones. Para ello hay que tener claro la posición inicial de los sensores a cada parte del cuerpo correspondiente. Estos también serán almacenados en la **BBDD**. Para sacar más información, se ha creado un modelo se comparan dichos datos. Debido a que la relación entre algunos de estos sigue un patrón. Esto se debe a que el deportista sigue un comportamiento parecido en sus paladas. Por lo tanto, existirá una dependencia entre estos movimientos. Las diferentes relaciones entre las diferentes gráficas serán las que definan las características de la técnica. El modelo muestra los movimientos del brazo, en el comportamiento cíclico de la palada y la eficiencia de la rotación de la cadera. Para simplificar dichos modelos se han hecho regresiones lineales a través de inteligencia artificial. De esta forma, se sacará de forma más sencilla datos sobre la palada. Y se ha hallado parámetros que indique lo fiel que es la regresión a la realidad.

Para validar el sistema diseñado, se han realizado pruebas de dos minutos a diferentes deportistas. Cada uno de ellos han ido a un ritmo distinto. Y a posteriori se han sacado gráficas con puntos que representa la dependencia de los movimientos. Y hay un plano o una recta que representan la regresión lineal calculada por la inteligencia artificial.

Abstract

The technique of the kayaking is a basic element in competitions of this sport. Because it allows reaching more velocity with less effort. Due to the force of the movement of stroke is made with all the muscles of the body. Therefore, that also improves the resistance in that test. That's why this work create a model of these movements from a test. This will allow seeing the errors of the sporter that can be done for the movement. To that the sporter can solve them and change the technique correctly. The work focus in the superior part of body, due to that allow seeing the basic behaviour of the movement.

For that reason, the first step is to get the data of the movements of sporter. With that goal, A system of sensors, that can be wearable, has been created. It has the goal of measuring the angles that the joints make. The architecture of the system has three main devices. Those are the IMUs (Inertial Measurement Unit), the microcontrollers and a Raspberry pi. And There is another device that has been used. That is a smartphone that works like user interface. Furthermore, there are two different protocols among the different devices. These are I2C (Inter-Integrated Circuit) and Bluetooth. These will be used for the communications among the different devices. Moreover, Wi-fi protocol is used for the communication of user.

The microcontroller uses the I2C protocol for controlling the IMU operation. In turn, the Raspberry pi will start a Bluetooth communication with the microcontrollers. That communication will be used for managing and controlling to all microcontrollers through the Raspberry pi. Additionally, the smartphone will be an AP (Access Point) that create a network where the Raspberry pi connect to. It has used the wi-fi protocol for the communication with the user.

That system will allow sending the got data to the Raspberry pi. The angles of the joints will be calculated from the data of the gyroscope and accelerometer. Those are sensors of the IMU, and they must be calibrated for working. The IMU gives the angles in way of quaternion. The calculated values will be sampled and sending to the Raspberry pi through of that system. There, they will be stocking up on a database. That data is linked to the data of the test and the IMU that has made.

Later, the joint information is created from the data of the IMUs. For that, the data of the different sensors is treated and merged. With the goal of getting the angles of joints. For that, there are that knowing the initial position of the parts of the body of the kayaker. These will also be stored up on database. For getting information, a model is created for comparing that data. Due to the relation between that information follow a pattern. Due to the sporter follow the pattern of the behaviour of the movement. Therefore, there will be a dependency among these movements. The different relations among the different graphics will define the feature of the technique. The model shows the movements of the arm, the cyclic behaviour of the arm and rotation efficiency. To simplify the model, lineal regressions are made to the graphics with artificial intelligence. And there are parameters that said if the regression is true with the reality.

For validating the designed system, tests of two minutes a have been made. The sporters have followed different rhythm of stroke in the different tests. And the points graphics that represent the relation between the movement have been made. And a plane or straight line will represent the lineal regressions of the artificial intelligence.

Índice

Agradecimientos	I
Resumen	III
Abstract	V
Índice	VII
Índice de Tablas	X
Índice de Figuras	XI
Notación	XVI
1 Objetivos	1
2 Introducción	3
3 Estado del arte	5
3.1 <i>Introducción</i>	5
3.2 <i>Uso de la tecnología en el deporte</i>	5
3.2.1 Tecnología en el arbitraje	5
3.2.2 Tecnología aplicada para lesiones	6
3.2.3 Tecnología para mejorar el deporte	8
3.3 <i>Piragüismo</i>	11
3.4 <i>Sensores</i>	12
3.4.1 Acelerómetro	12
3.4.2 Giroscopio	14
3.4.3 Procesamiento de los datos	16
3.4.3.1 Ángulos de Euler	16
3.4.3.2 Cuaterniones	17
3.4.3.3 Calibración	19
3.4.4 IMU MPU-6050	20
3.5 <i>Protocolo I2C</i>	21
3.5.1 Buses	21
3.5.2 Dispositivos	21
3.5.3 Protocolo	22
3.6 <i>Microcontroladores</i>	23
3.6.1 Componentes	23
3.6.2 Procedimiento	24
3.6.3 Tipos de procesadores	24
3.6.4 ESP 32	25
3.7 <i>Bluetooth</i>	26

3.7.1	Aplicaciones	27
3.7.2	Perfiles Bluetooth	28
3.7.3	Funcionamiento	29
3.7.3.1	Controladores	29
3.7.3.1.1	Capa Física	30
3.7.3.1.1.1	Canales físicos	30
3.7.3.1.1.2	Enlaces físicos	31
3.7.3.1.2	Capa lógica	31
3.7.3.1.2.1	Transporte lógico	31
3.7.3.1.2.2	Enlaces lógicos	32
3.7.3.2	HCI	32
3.7.3.3	Host	32
3.7.3.3.1	L2CAP	33
3.7.3.3.2	SDP (Service Discovery protocol)	33
3.7.3.3.3	GAP	34
3.7.3.3.3.1	Modos y Procedimientos en BR/EDR	34
3.7.3.3.3.1.1	Descubrimiento	34
3.7.3.3.3.1.2	Vinculación	35
3.7.3.3.3.1.3	Establecimiento de enlace	35
3.7.3.3.3.1.4	Establecimiento del canal y de la conexión	35
3.7.3.3.3.1.5	Broadcast	35
3.7.3.3.3.2	Modos y procedimientos en LE	35
3.7.3.3.3.2.1	Descubrimiento	35
3.7.3.3.3.2.2	Conexión	36
3.7.3.3.3.2.3	Vinculación	36
3.7.3.3.3.2.4	Aviso periódico	36
3.7.3.3.3.2.5	Broadcast isocrónico	37
3.7.3.3.4	Atributos	37
3.7.3.3.4.1	ATT (Attribute protocol)	37
3.7.3.3.4.2	GATT	37
3.8	<i>Raspberry Pi</i>	37
3.9	<i>BBDD (Base de datos)</i>	39
3.9.1	Tipos de Base de datos	39
3.9.2	Lenguaje de base de datos	40
3.9.3	DBMS	40
3.9.3.1	BBDD autónoma	41
3.9.4	Servidor de base de datos	41
3.10	<i>Inteligencia Artificial</i>	41
3.10.1	Aplicaciones	42
3.10.2	Tipos y algoritmos de machine learning	43
3.10.3	Scikit learn	47
3.11	<i>Conclusión</i>	48
4	Aplicación	49
4.1	<i>Introducción</i>	49
4.2	<i>Arquitectura</i>	49
4.2.1	IMU	51
4.2.2	Microcontroladores	51
4.2.3	Raspberry pi	51
4.3	<i>Base de datos</i>	52
4.3.1	Diseño	52

4.4	<i>Protocolo</i>	54
4.4.1	Funcionamiento	54
4.4.1.1	Procedimientos para medir	54
4.4.1.2	Configuración de la base de datos	57
4.4.1.3	Análisis de los Datos	61
4.5	<i>Conclusión</i>	62
5	Validación del trabajo	63
5.1	<i>Introducción</i>	63
5.2	<i>Medidas</i>	63
5.2.1	Preparación	63
5.2.2	Realización	65
5.3	<i>Resultados</i>	67
5.4	<i>Conclusión</i>	109
6	Conclusión	110
7	Anexo	113
7.1	<i>Interfaces</i>	113
	Referencias	121

ÍNDICE DE TABLAS

Tabla 3.3-1. Ritmos a los que se puede remar y distancia o tiempos para los que están diseñados (18).	11
Tabla 3.4-1. Rangos de los sensores del MPU-6050	20

ÍNDICE DE FIGURAS

Esquema 4.2–1. Se muestra como están las conexiones dentro del módulo sensor y con el módulo central.	50
Esquema 4.2–2. Esquema de la arquitectura seguida a la hora de realizar las medidas al deportista.	50
Esquema 4.3–1. Tablas de la base de datos y las relaciones que guardan unas con las otras.	53
Esquema 4.4–1. Diagrama de paso de mensaje entre un microcontrolador y la Raspberry pi teniendo un funcionamiento correcto.	56
Esquema 4.4–2. Esquema de tren superior en estado inicial con los sensores colocados en la parte correspondiente del cuerpo. Los ssrr (sistemas de referencias) azules corresponden a los ssrr iniciales de los sensores y los rojos a los ssrr que deben tener las articulaciones,	57
Esquema 4.4–3 . Máquina de estado general que se ha seguido a la hora de fusionar los datos.	58
Esquema 4.4–4. Máquina de estado para adquirir los ángulos de la cadera	59
Esquema 4.4–5. Máquina de estado seguido para adquirir los ángulos realizados por cada hombro.	60
Esquema 4.4–6. Máquina de estados seguido para adquirir los ángulos realizado por cada codo.	61
Ilustración 4.2-1. VMS en funcionamiento	6
Ilustración 4.2-2. Gafas con eye tracker .	7
Ilustración 4.2-3. Ergómetro de piragüismo.	10
Ilustración 4.4-1. Acelerómetro mecánico actuando en una única dirección	13
Ilustración 4.4-2. Funcionamiento de acelerómetro MEMS	14
Ilustración 4.4-3. Funcionamiento de un giroscopio vibrante.	15
Ilustración 4.4-4. Giroscopio óptico	16
Ilustración 4.4-5. Sensor MPU-6050	20
Ilustración 4.5-1. Colocación de los diferentes dispositivos dentro del bus I2C.	22
Ilustración 4.6-1. Arquitectura Harvard de un microcontrolador	25
Ilustración 4.6-2. Placa ESP32 WROOM 32	26
Ilustración 4.7-1. Torres de protocolos de Bluetooth. A la izquierda nos encontraríamos la BR/EDR (rojo) y a la derecha la torre de protocolo de Bluetooth LE (azul)	33
Ilustración 4.8-1. Raspberry pi 4 modelo B	39
Ilustración 4.10-1. Chat de texto de una conversación con Gyant	43
Ilustración 4.10-2. Ejemplo de un algoritmo en forma de árbol de decisión.	44
Ilustración 4.10-3. Algoritmo básico de inteligencia de refuerzo	46
Ilustración 4.10-4. Muestra de una red neuronal y sus partes	47
Ilustración 5.2-1. Imagen del módulo sensor. Se puede observar cómo está conectado el microcontrolador ESP 32 (microchip negro) a la IMU MPU 6050 (atornillador al recipiente) mediante I2C.	51
Ilustración 5.4-1. Interfaz creada para realizar una medida. Se puede ver como todos las IMUs han tenido que	

estar conectados para calibrar los sensores y empezar a medir.	55
Ilustración 6.2-1. Traje realizado para fijar los sensores a cada parte del cuerpo.	64
Ilustración 6.2-2. Deportista equipando los sensores. Y está tomando la postura inicial para calibrar los sensores	65
Ilustración 6.2-3. Piragüistas que han ejecutado las pruebas.	66
Ilustración 6.2-4. Barcos de los piragüistas que han realizado las pruebas.	66
Ilustración 6.2-5. Palas de piragüistas que han realizado las pruebas.	67
Ilustración 6.2-6. Pruebas que se han realizado.	67
Ilustración 8.1-1. Ejemplo para mostrar la interfaz para introducir los datos de un nuevo piragüista.	113
Ilustración 8.1-2. Ejemplo para mostrar la interfaz para actualizar los datos de un piragüista.	114
Ilustración 8.1-3. Ejemplo para mostrar la interfaz para introducir los datos de un nuevo barco.	115
Ilustración 8.1-4. Ejemplo para mostrar la interfaz para actualizar los datos de un barco.	116
Ilustración 8.1-5. Ejemplo para mostrar la interfaz para introducir los datos de una nueva pala.	117
Ilustración 8.1-6. Ejemplo para mostrar la interfaz para actualizar los datos de una pala.	118
Ilustración 8.1-7. Ejemplo para mostrar la interfaz para actualizar los datos de una salida.	120
Video 6.2-1. Realización de una salida del primer sujeto haciendo un seguimiento desde un barco.	66
Gráfica 4.4-1. Representación de un cuaternión-Formado por un vector e y un ángulo de giro.	17
Gráfica 6.3-1. Evolución de los ángulos de los movimientos de las articulaciones a lo largo del tiempo de la salida 164. Aclaración. Las columnas son de derecha a izquierda la cadera, el hombro derecho e izquierdo, relativamente, y el codo derecho e izquierdo, relativamente. Y en el título esta el nombre de cada articulación.	68
Gráfica 6.3-2. Evolución de los ángulos de los movimientos de las articulaciones a lo largo del tiempo de la salida 165. Aclaración. Las columnas son de derecha a izquierda la cadera, el hombro derecho e izquierdo, relativamente, y el codo derecho e izquierdo, relativamente. Y en el título esta el nombre de cada articulación.	69
Gráfica 6.3-3. Evolución de los ángulos de los movimientos de las articulaciones a lo largo del tiempo de la salida 167. Aclaración. Las columnas son de derecha a izquierda la cadera, el hombro derecho e izquierdo, relativamente, y el codo derecho e izquierdo, relativamente. Y en el título está el nombre de cada articulación.	70
Gráfica 6.3-4. Evolución de los ángulos de los movimientos de las articulaciones a lo largo del tiempo de la salida 168. Aclaración. Las columnas son de derecha a izquierda la cadera, el hombro derecho e izquierdo, relativamente, y el codo derecho e izquierdo, relativamente. Y en el título está el nombre de cada articulación.	71
Gráfica 6.3-5. Evolución de los ángulos de los movimientos de las articulaciones a lo largo del tiempo de la salida 169. Aclaración. Las columnas son de derecha a izquierda la cadera, el hombro derecho e izquierdo, relativamente, y el codo derecho e izquierdo, relativamente. Y en el título esta el nombre de cada articulación.	72
Gráfica 6.3-6. Evolución de los ángulos de los movimientos de las articulaciones a lo largo del tiempo de la salida 170. Aclaración. Las columnas son de derecha a izquierda la cadera, el hombro derecho e izquierdo, relativamente, y el codo derecho e izquierdo, relativamente. Y en el título está el nombre de cada articulación.	73
Gráfica 6.3-7. Evolución de los ángulos de los movimientos de las articulaciones a lo largo del tiempo de la salida 171. Aclaración. Las columnas son de derecha a izquierda la cadera, el hombro derecho e izquierdo,	

relativamente, y el codo derecho e izquierdo, relativamente. Y en el título está el nombre de cada articulación. 74

Gráfica 6.3-8. Evolución de los ángulos de los movimientos de las articulaciones a lo largo del tiempo de la salida 172. Aclaración. Las columnas son de derecha a izquierda la cadera, el hombro derecho e izquierdo, relativamente, y el codo derecho e izquierdo, relativamente. Y en el título está el nombre de cada articulación. 75

Gráfica 6.3-9. Comparación de las entradas de la rotación del hombro derecho con respecto a la flexión del hombro derecho y la rotación de la cadera de la salida 164. Y plano del modelo lineal de las entradas. 76

Gráfica 6.3-10. Comparación de las entradas de la rotación del hombro izquierdo con respecto a la flexión del hombro izquierdo y la rotación de la cadera de la salida 164. Y plano del modelo lineal de las entradas. 77

Gráfica 6.3-11. Comparación de las entradas de la rotación del hombro derecho con respecto a la flexión del hombro derecho y la rotación de la cadera de la salida 165. Y plano del modelo lineal de las entradas. 78

Gráfica 6.3-12. Comparación de las entradas de la rotación del hombro izquierdo con respecto a la flexión del hombro izquierdo y la rotación de la cadera de la salida 165. Y plano del modelo lineal de las entradas. 78

Gráfica 6.3-13. Comparación de las entradas de la rotación del hombro derecho con respecto a la flexión del hombro derecho y la rotación de la cadera de la salida 167. Y plano del modelo lineal de las entradas. 79

Gráfica 6.3-14. Comparación de las entradas de la rotación del hombro izquierdo con respecto a la flexión del hombro izquierdo y la rotación de la cadera de la salida 167. Y plano del modelo lineal de las entradas. 80

Gráfica 6.3-15. Comparación de las entradas de la rotación del hombro derecho con respecto a la flexión del hombro derecho y la rotación de la cadera de la salida 168. Y plano del modelo lineal de las entradas. 81

Gráfica 6.3-16. Comparación de las entradas de la rotación del hombro izquierdo con respecto a la flexión del hombro izquierdo y la rotación de la cadera de la salida 168. Y plano del modelo lineal de las entradas. 82

Gráfica 6.3-17. Comparación de las entradas de la rotación del hombro derecho con respecto a la flexión del hombro derecho y la rotación de la cadera de la salida 169. Y plano del modelo lineal de las entradas. 82

Gráfica 6.3-18. Comparación de las entradas de la rotación del hombro izquierdo con respecto a la flexión del hombro izquierdo y la rotación de la cadera de la salida 169. Y plano representa el modelo lineal de las entradas. 83

Gráfica 6.3-19. Comparación de las entradas de la rotación del hombro derecho con respecto a la flexión del hombro derecho y la rotación de la cadera de la salida 170. Y plano representa el modelo lineal de las entradas. 84

Gráfica 6.3-20. Comparación de las entradas de la rotación del hombro izquierdo con respecto a la flexión del hombro izquierdo y la rotación de la cadera de la salida 170. Y plano representa el modelo lineal de las entradas. 85

Gráfica 6.3-21. Comparación de las entradas de la rotación del hombro derecho con respecto a la flexión del hombro derecho y la rotación de la cadera de la salida 171. Y plano representa el modelo lineal de las entradas. 86

Gráfica 6.3-22. Comparación de las entradas de la rotación del hombro izquierdo con respecto a la flexión del hombro izquierdo y la rotación de la cadera de la salida 169. Y plano representa el modelo lineal de las entradas. 86

Gráfica 6.3-23. Comparación de las entradas de la rotación del hombro derecho con respecto a la flexión del hombro derecho y la rotación de la cadera de la salida 172. Y plano representa el modelo lineal de las entradas. 87

Gráfica 6.3-24. Comparación de las entradas de la rotación del hombro izquierdo con respecto a la flexión del hombro izquierdo y la rotación de la cadera de la salida 172. Y plano representa el modelo lineal de las entradas. 88

Gráfica 6.3-25. Comparación de las entradas de la flexión del codo derecho con respecto a la flexión y abducción del hombro derecho de la salida 164. Y plano representa el modelo lineal de las entradas. 89

Gráfica 6.3-47. Comparación de las entradas de la flexión del hombro derecho (arriba) e izquierdo (abajo) con respecto la rotación de la cadera de la salida 171. Y la línea recta representa el modelo lineal de las entradas.

Gráfica 6.3-48. Comparación de las entradas de la flexión del hombro derecho (arriba) e izquierdo (abajo) con respecto la rotación de la cadera de la salida 172. Y la línea recta representa el modelo lineal de las entradas.

Notación

q^*	Conjugado
VAR	Video Assitant Referee
VMS	Video Monitor System
IMU	Inercial Motion Unit
I2C	Inter-Integrated Circuit
SDP	Service Discovery Service
DBMS	Database Managment System
V	Voltios
A	Amperios
DMP	Digital Motion Processor
IDE	Integrated Development Enviroment
BBDD	Base de datos
	Módulo
SoC	System on Chip
SVM	Support vector machine
EMC	Error cuadrático medio
SSH	Secure Shell
RFCOM	Comunicación de radio frecuencia
M	
LAN	Local Area Network
SQL	Structural Query Language
Ssrr	Sistema de referencia
=	Igual
PCA	Principal Component Analysis
SSL/TLS	Security socket layer y Transport layer security
S	
HDMI	High-definition multimedia Interface
CSI	Camera Serial Interface
DSI	Display serial Interface
ARM	Advanced RISC machine
CISC	Complex instrustion set computer
RISC	Reduced instruction set computer
USB	Universal serial bus
G	Unidad que indica la fuerza de la gravedad
$^{\circ}/s$	Cantidad de angulo en grado por segundo
Cov	Covarianza
GHz	Giga Hercios
GPS	Global Positioning System

MPU	Multiple Process Unit
PTS	Precision time System
MRI	Magnetic Resonance Imaging
US	Ultrasound
HCI	Host-Controller interface
PWM	Pulse width modulation
GAP	General Access profile
ATT	Attribute protocol
GATT	Generic Attribute Profile
SIM	Subscriber identify module
BIS	Broadcast Isochronous stream
BIG	Broadcast Isochronous group
UUID	Universally Unique Identifier
SDA	System data
SCL	System clock
RAM	Random Access memory
ROM	Read only memory
SRAM	RAM estática
SSID	Identificador de servicio Bluetooth
I2S	Inter integrated sound
XML	Extend Markup language
BLE	Bluetooth low energy
L2CAP	Logical Link Control and Adaptation Protocol
A2DP	Advanced Audio Distribution Profile
AVRCP	Audio/Video Remote Control Profile
FTP	File Transfer Protocol
DIP	Device ID Profile
HFP	Hands-free Profile
HIC	Human Interface Device Profile
PAN	Personal Area Networking Profile
SPP	Serial Port Profile
SAP	SIM Access Profile
IoT	Internet of Things
ISM	Industrial, Scientific and Medic Band
q_ω	Parte real de un cuaternión
q_x	Parte imaginaria del eje x
q_y	Parte imaginaria del eje y
q_z	Parte imaginaria del eje z
UART	Universal Asynchronous Receiver-Transmitter
SPI	Serial Port Interface
PSRAM	Pseudo-estático RAM
MCU	Microcontroller unit
I/O	Entrada/Salida
QSPI	Quad Serial Port Unit

ACL	Asynchronous Connection Link
SCO	Synchronous Connection Oriented
eSCO	Extended Synchronous Connection Oriented
AFH	Adaptative Frecuency Hop
SIG	Special Interest Group
LE	Low Energy
BR	Basic rate
EDR	Enchanced data rate
FDMA	Frecuency Division Multiple Access
TDMA	Time Division Multiple Access
TCP	Transport Control Protocol
IP	Internet Protocol
LMP	Link Management Protocol
LL	Link Layer
APB	Active Peripheral Broadcast
CPB	Conecttionless Peripheral Broadcast
ADVb	Advertisement Basic
PADVb	Periodic Advertisement Broadcast
SDIO	Secure Digital I/O
MIPI	Mobile Industry Proccessor Interface
GB	Gigabyte
EPROM	Erasable Programmable Read only Memory
EEPROM	Electrically Erasable Programmable Read only Memory
M	
ARW	Angular Random Walk
JSON	JavaScript Object Notation
BSON	Binary JSON
AP	Punto de acceso
3D	Tres dimensiones
MAC	Media Access Control
VBA	Visual Basic Application

1 OBJETIVOS

Con este trabajo se quiere caracterizar el movimiento de paleo que hacen los piragüistas. Ya que depende de cómo se realice este puede contribuir al menor o mayor esfuerzo del ejercicio. Esto afecta a cualidades como la velocidad, resistencia y estabilidad en la palada que pueda conseguir el deportista. Y estas cualidades pueden significar una victoria o derrota en una competición.

Para ver la postura que tiene el piragüista durante el movimiento de paleo, se va a construir un modelo de la palada. Este modelo tendrá como objetivo sacar las características de la palada. Este modelo se sacará a partir de medidas de la postura durante el movimiento de paleo. Y una vez que se obtenga los datos se debe sacar conclusiones a partir de ellos. Estas conclusiones podrán servir de ayuda a los deportistas, para saber cómo realizan la palada. Y a partir de estos datos aprender a mejorarla. Además, puede servir de utilidad para futuros proyectos.

Para realizar este trabajo adecuadamente se tiene que cumplir los siguientes objetivos:

1. Instalar en los diferentes palistas varios sensores. Estos se encargarán de hacer las medidas necesarias para definir la cinemática de los deportistas. Se intenta sacar las características que permitan ver las características que tiene la postura. Los sensores deben dar datos acerca de la situación en la que se encuentran las articulaciones. En concreto se va a centrar en el tren superior. Ya que se considera que es donde se hace el trabajo más básico en el piragüismo. Por lo tanto, los sensores tendrán que medir el movimiento de la cadera, hombros, codo y muñecas. O al menos, dar información útil acerca de esos movimientos. Se necesitarán varios sensores ya que son varias articulaciones las que se desean medir. Debe de tener velocidad de muestreo lo suficientemente alta. Ya que si no se alcanza está no se podrá determinar el movimiento de la palada solo con esos datos. Estos deben de ser cómodo para los deportistas, al menos para ejecutar cómodamente el movimiento.
2. Se necesitará un módulo central que se encargue de reunir toda la información y almacenarla. De manera, que se necesita que este dispositivo sea capaz de comunicarse con los sensores. Y necesita que vaya a la velocidad necesaria que necesita para Para de poder recopilar la información con la que va a tratar. Debe de ser capaz de tratar con datos de las dimensiones de las medidas de los sensores. Debe de ser capaz de comunicarse todos los sensores a la vez. Ya que interesa obtener el movimiento de todas las articulaciones al mismo tiempo. Por lo tanto, debe de ser capaz de realizar ejecuciones en paralelo de un programa.
3. Los sensores tienen que comunicarse inalámbricamente con el módulo central. Esto es debido a que los sensores irán incorporados en el cuerpo de los deportistas. Por lo tanto, si fuera cableado se dificultaría la realización del ejercicio al deportista. Este protocolo debe de soportar el volumen de datos que se producen con las medidas. Que la comunicación sea inalámbrica hace que los sensores deban ser alimentado independientemente. Se debe asegurar la conectividad entre los sensores y el módulo central durante la medida. Esto no siempre será posible debido a que el entorno de la medida será un río, un lago o el mar. Por lo tanto, puede ser complicado que estas zonas tengan una conectividad a internet.
4. El usuario de la aplicación debe de tener un control sobre los sensores. Esto es debido a que deben conocer cuando empezar a realizar las mediciones y cuando acabarla. Además, por consumo, interesa que estos envíen datos solo en las salidas. A parte, se tiene que comunicar para saber cuándo calibrar los sensores.

5. Se necesita que las mediciones sean en escenarios reales. Esto se debe a que queremos que las mediciones sean fieles a la realidad. Es decir, se necesita instalarle los sensores en un espacio donde puedan remar. Esto puede ser tanto en un río, una ría, un lago o en el propio mar. Por lo tanto, se necesitará que todo equipo electrónico sea protegido antes el agua. Además, al menos los sensores deberán ir atados al cuerpo. Ya que están midiendo la posición de las articulaciones de este. Por lo tanto, se necesitará una forma de mantener los sensores atados fijos a partes del cuerpo. Ya que el sensor debe de representar la dinámica de esta parte del cuerpo. Si el sensor varía de posición con respecto a esa parte del cuerpo, ya no será fiel a la realidad. Ya que la medida del sensor será la dinámica de la articulación más la del propio sensor.
6. Si fuera necesario se necesitan sacar datos útiles de las medidas de los sensores. Para ello se debe ser capaz de sacar las medidas de los sensores de donde se almacenan. Y realizar los procedimientos necesarios para sacar la información sobre la postura que necesitamos. El dispositivo que realice esto debe de ser capaz de acceder a los datos almacenados. Y de tener la suficiente capacidad de procesamiento de la información necesario.
7. Los datos de la postura necesitan ser tratados para poder dar información útil para los deportistas. En tal caso, se necesitaría que se desarrollarán para poder visualizar características de la palada. Y posteriormente sacar modelos sencillos para interpretar estas características. De esta forma, se conseguirá que los deportistas se den cuenta de su forma de remar. De tal manera, que puedan cambiar los aspectos con los que no están de acuerdo.

Como se puede comprobar el sistema permitiría medir el movimiento de los piragüistas en situaciones reales. De tal forma, que se pueda medir durante sesiones de entrenamiento. Para ello, hay que cumplir requisitos en busca de la seguridad de los equipos y la comodidad de los deportistas. Y a posteriori, se realiza un tratamiento de los datos que nos de información de la palada. Para ello se tiene que almacenar los datos y realizar el tratamiento adecuado de ellos. Por último, se representaría la información hallada a través de los datos.

2 INTRODUCCIÓN

En este apartado, se hablará sobre todo el proyecto en completo y cada una de las partes del trabajo. En el estado del arte, se comentará sobre todos los elementos que hemos implementado en el trabajo. Este incluye los protocolos que se han hecho uso en el trabajo. En nuestro caso, las conexiones más importantes han sido Bluetooth e **I2C** (*Inter-Integrated Circuit*). En el caso de Bluetooth, se ha explicado todas las aplicaciones en las que se puede usar, ya que esta comunicación es de más alto nivel que **I2C**. **I2C** es un protocolo de conexión cableada entre dispositivos electrónicos. En ambos casos, se explica que mensajes existen. Además, en el caso de **I2C**, el protocolo que sigue. En el caso de Bluetooth, el paso de mensajes dependerá del modo en que se usa. Además, estas comunicaciones son más complicada y no es solo envío y lectura de datos. Esto se debe a que es un lenguaje de más alto nivel. Se ha explicado los dispositivos con los que vamos a trabajar. Además, se habla sobre el funcionamiento de los sensores que necesitamos. Además, se ha expuesto los diferentes modelos de este que existen. Y se ha detallado del sensor que se ha utilizado. También se ha hablado sobre que son los microcontroladores y sus aplicaciones. Y sobre todo se ha detallado el caso del microcontrolador ESP 32 con Bluetooth y wi-fi. Y el último elemento sería la Raspberry pi. Se ha explicado tanto las cualidades de la versión utilizada como la de la última versión y sus aplicaciones.

En realidad, habría un componente más en la comunicación, otro protocolo y una aplicación. No se ha incluido en el estado del arte, ya que, se ha trabajado indirectamente con estas tecnologías. Si no que ha sido una solución a una dificultad de comunicación con la Raspberry pi mientras se realizan las medidas. Y es que para mandar órdenes a la Raspberry pi, se necesita un Shell de este. Sin embargo, en este entorno no es posible conectar un monitor cableado y es complicado llevar uno. Por ello, se ha ejecutado una Shell remota a través de SSH (Secure Shell), a la que podía acceder con mi smartphone. Para crear la conexión entre la Raspberry pi y el dispositivo se ha hecho uso de wi-fi. Esta tiene un mayor alcance que Bluetooth. Además, se puede hacer un AP (punto de acceso) desde la mayoría de los dispositivos móviles.

Para terminar el estado del arte, se explica los conceptos de **BBDD** (*base de datos*) y la inteligencia artificial. De la **BBDD**, se explica que tipos de estas existen y cuales con las más actuales. Además, se ha hablado sobre el **DBMS** (*DataBase Managment System*). Que sirve para acceder y modificar esta. Y sobre el servidor de la **BBDD**. Este permite acceder a los clientes mediante un usuario, ya sean remotos o locales. En la inteligencia artificial, se ha definidos que son y sus muchas aplicaciones. Y luego, se ha centrado el apartado en los algoritmos de *machine learning*. Después, se ha hablado sobre la librería de Python centrada en *machine learning*. Por último, mencionar los campos de la tecnología en el deporte, ya que es lo que se está haciendo. Se ha hablado como se aplica está en el campo del arbitraje, en la parte médica y para mejorar la ejecución de este. Y un poco acerca de la técnica que existe en el piragüismo.

En el apartado de aplicaciones, se ha comentado los procedimientos que he seguido a la hora de realizar el trabajo. Primero se ha hablado de como se ha realizado la conexión entre los diferentes dispositivos. Es decir, el **IMU** (*Inertial Measurement Unit*) **MPU** (*Multiple Process Unit*) 6050 está conectado a un microcontrolador ESP 32 a través de **I2C**. Para que este controle los datos del sensor. Luego, está la comunicación de la Raspberry pi con todos los controladores a través de Bluetooth. En que este se encarga de dirigir a los microcontroladores de forma conjunta. Por último, estaría el acceso a la Raspberry pi conectado por wi-fi al dispositivo móvil. Después, se habla del almacenamiento de los datos recogidos por los **IMUs**. Sobre el servidor local de **BBDD** creado y el diseño de esta **BBDD**. Posteriormente, se habla del procedimiento en el que se realiza las medidas. Se habla sobre que fichero se debe ejecutar para cada función y lo que hacen estos internamente. Sobre todo,

interesa la gestión de la comunicación con los microcontroladores. Luego, se explica de cómo los usuarios pueden modificar esta base de datos. Aunque deben hacer uso del **DBMS** para leer los datos de esta. En este apartado, también se comenta el tratamiento de los datos de medida para alcanzar los datos deseados. Para ello, se hace usos de funciones matemática se ha utilizado para conseguir dicha información. Por tanto, tanto la obtención de estos datos como los análisis son a posteriori. Por último, hablar sobre la representación de las medidas una vez que ya han sido calculadas. Y sobre el desarrollo del modelo para alcanzar las características de la técnica. Se mencionan sobre qué datos y la razón por los que se han escogido para realizar el modelo. Además, se describen las técnicas de *machine learning* usadas.

En la validación del trabajo, se habla sobre como he realizado las medidas sobre un escenario real. Ha descrito el procedimiento anterior a realizar las medidas. En donde se equipa al deportista con los sensores y estos se calibran. Y posteriormente, se ha mostrado los resultados obtenidos. Y se ha concretado para los resultados que mediante las gráficas he comprobado su fidelidad con la realidad. Además, sé comenta qué se puede ver de los resultados obtenidos. También se ha comentado errores que he tenido en las medidas que se reflejan en los resultados.

3 ESTADO DEL ARTE

3.1 Introducción

Se va a hacer una descripción del estado del arte de las tecnologías usadas en nuestro trabajo. Esto va a incluir una descripción del contexto de la tecnología en el deporte. Aquí se va a explicar tanto los diferentes tipos de aplicaciones que se pueden dar en el panorama actual. También procederemos a describir el estado en el que se encuentra todas las tecnologías de las que se ha hecho uso. Así como indicar las versiones que se ha utilizado de las mismas. Se expondrán tanto de los protocolos como de los dispositivos que he estado usando. Además, se enseñará la tecnología de almacenamiento utilizada. También se han mostrado nociones básicas acerca del piragüismo. Además, se explicará los conocimientos matemáticos que se sigue a la hora de hacer el procesamiento de los datos. Y para finalizar se verá los conocimientos de las inteligencias artificiales. Además, se detallará en la inteligencia artificial de la que se ha hecho uso. Que ha sido tan importante para una mejor comprensión del análisis de los datos.

3.2 Uso de la tecnología en el deporte

Cada vez más se va viendo como en los deportes se va utilizando recursos tecnológicos para mejorar su ejecución. Y se puede ver en diversos ámbitos. Como pueden ser como ayudar a los árbitros que se cumpla el reglamento del deporte. También se utiliza la tecnología para evitar las lesiones en los deportistas. Por último, mencionar toda aquella tecnología que permite el entrenamiento de los deportistas. En este último tema asunto es el que trata mi trabajo. Y más en concreto, hablaremos sobre como es este campo en el piragüismo.

3.2.1 Tecnología en el arbitraje

Vamos a empezar a hablar sobre la tecnología que facilita la actuación del árbitro. Vamos a empezar hablando de los sistemas de monitorización de video que se utilizan en los diferentes deportes (1). Se puede poner de ejemplo los conocidos **HawkEye** en tenis, el **VAR** (*Video Assistant Referee*) en el futbol y del **VMS** (*Video Monitoring System*) en el waterpolo (1). Esta tecnología no necesita un aprendizaje muy complejo por parte del usuario. Lo que es ideal a la hora de poder entrenar a los árbitros en su uso. Estos métodos utilizan sistemas de video que analizan las jugadas durante las competiciones. Para este y otros propósitos explicados más adelante se llena los estadios de redes de cámaras. De esta forma, se puede hacer un análisis de las grabaciones de esta. Este consiste en detectar el objeto de varios objetos. Ya estos puedan ser un objeto como son las pelotas de los deportes anteriormente mencionados. Y también estos sistemas identifica la colocación de los jugadores. En el caso del tenis tenemos un sistema **3D** (*three dimensions*) que realiza el seguimiento de la bola. Existen sistema, como este, puede detectar eventos que ocurren durante la competición. En este caso el evento que suele detectar es que la bola bote en el suelo.



Ilustración 3.2-1. VMS en funcionamiento (2).

A parte del uso de sistemas de video, los árbitros tienen otras herramientas tecnológicas para ayudarle a hacer su trabajo. Como es el caso del **PTS** (*Precision Time System*) en el baloncesto (3). Este sistema conecta el silbato del árbitro a los relojes que hay en el estadio de baloncesto. Cuando pita, el sensor que hay en el silbato lo detecta. Este envía una señal a todos los relojes para que se paren. Y cuando vuelve a usar el silbato, este envía una señal para reanudar los relojes. Aun así, este sistema tiene sus fallos. Esto es debido a que hay un tiempo entre que ocurre un evento que el árbitro tiene que pitar hasta que lo pita. Por ello es necesario que haya operarios pendientes del tiempo. Y en el caso de que haya un mal funcionamiento en reloj, se tendría que poder añadir tiempo extra.

3.2.2 Tecnología aplicada para lesiones

Desde un punto de vista más médico tenemos el uso de la tecnología para evitar y analizar lesiones. En este conjunto de usos también vamos a englobar toda aquella tecnología usada para la rehabilitación. Al fin al cabo todos estos usos están relacionados con la medicina. Para empezar, hablemos sobre imágenes médicas en el deporte. Nos podemos encontrar diferentes métodos, algunos de ellos pueden ser la **MRI** (*Magnetic Resonance imaging*) y las imágenes **US** (*UltraSound*). También se suele mezclar ambas técnicas para examinar mejor las lesiones en músculos y tendones (4). Esto es debido a que la resonancia ayuda a compensar el limitado campo de visión de US.

En este campo también podemos encontrar técnicas más avanzadas como son el **eye tracking**. Esta técnica permite analizar los movimientos oculares. Esto se realiza mediante una herramienta llamada **eye tracker**. Para ello esta lanza rayos infrarrojos a los ojos que rebotan en estos y son recogidos por el propio dispositivo. Para poder ver alguna clase de movimiento ocular de los sujetos a estos se les pones una serie de imágenes. Esta clase de técnica se puede utilizar para identificar las diferentes conmociones cerebrales que sea podido producir en el deporte (5). Para ello de la monitorización el ojo se saca varias variables que permiten clasificar las conmociones.



Ilustración 3.2-2. Gafas con **eye tracker** (6).

Además, podemos incluir en este grupo los sensores que se pueden equipar en nuestro cuerpo. Y es que gracias a los datos proporcionados por esta clase de sensores. Se podría determinar si se realizan movimiento peligroso durante la práctica del deporte (7). Un ejemplo de esto puede ser el estudio del movimiento del hombro tanto en béisbol como en voleibol. Esto es debido a que estos deportistas suelen sufrir un sobreesfuerzo en esa parte del cuerpo. Por esa razón utilizan los datos medidos por este tipo de sensores para clasificar los movimientos que se realizan. Incluso algunos de estos sensores permiten dar información a partir de las medidas en tiempo real. Permitiendo de esta manera que se pueda realimentar al atleta con la información del sensor en tiempo real (7). Por ejemplo, en un experimento se utilizó un pitido para indicar como estaba haciendo los movimientos. También este tipo de sensores pueden permitir hacer el seguimiento sobre algunas lesiones que tenga el deportista. Por ejemplo, en el futbol americano se utiliza un sensor para el seguimiento de conmociones cerebrales (7). Para ello se utilizan un tipo de sensores que miden la aceleración angular y lineal de la cabeza. Estos están ubicados en el casco y en el protector bucal, de esta forma se consigue que sea poco intrusivo. Estos permiten detectar las conmociones que puede ser crucial a la hora de curar al deportista. A parte permite obtener información vital para evitar que este tipo de lesiones se sigan produciendo. Estos sensores tienen una tasa de efectividad del 95%.

Como habrá podido observar todos los sensores que se utilizan en este apartado son del mismo tipo. Esto es debido a que estos permiten ser usados en ambientes de deportes y son asequibles. Por ello son los únicos que son útiles a la hora de poder prever lesiones. Ya que estos permiten identificar un movimiento incorrecto durante los ejercicios. Los demás sensores te dan otra clase de información acerca del desarrollo del deporte. Esta forma de medición no está considerada la forma estándar de captura de movimiento. Aunque como se ha visto que estos sensores son increíblemente útiles en ámbitos deportivos. Por lo tanto, la forma que tenemos para validar este tipo de sensores es a través de sistemas ópticos (7). Ya que estos están considerados como el método estándar del reconocimiento de movimiento. Debido a su pequeño error en las medidas. Para comprobarlo se utilizan ambos métodos para capturas las trayectorias de las diferentes partes del cuerpo. Y posteriormente se comparan ambos métodos utilizados para ver las diferencias de las diferentes trayectorias. De esta forma podemos observar la fiabilidad de estos tipos de sensores frente al método estándar. Ya que esta la consideramos como una aproximación bastante buena de la realidad.

También existen, como se ha mencionado anteriormente, herramientas que facilitan la rehabilitación de los deportistas lesionados. Esta el caso de los sensores flexible que se aplican a deportista de élite para la rehabilitación de lesiones (8). Se encargan de lesiones que afectan a las funciones neuromusculares y muscular-esqueléticas. Estos tipos de sensores permiten trazar los movimientos que realizan las articulaciones. El cambio de la resistencia cuando se aplica una fuerza en el sensor es lo que permite detectar este movimiento. Estas lesiones que se miden con estos sensores pueden ser provocada durante la práctica de un deporte o por infartos. El uso de este tipo de sensores todavía está limitado a condiciones de laboratorio. Esto es debido a que los sensores que usan no permiten detectar todas las libertades y amplitudes de movimiento. Además, este tipo de sensores no son tan portable como los demás. Aunque, por supuesto, el aumento de la tecnología hace que estos sensores sean cada vez de menor tamaño. Esto se debe al avance en Smart-clothing, la comunicación inalámbrica y a los instrumentos y medidas de los investigadores.

3.2.3 Tecnología para mejorar el deporte

Por último, la tecnología también puede ayudar a mejorar la ejecución de los ejercicios. A parte de poder crear material que ayude tanto a entrenadores como a los propios deportistas. Para ello existen diferentes tecnologías como pueden ser el análisis de videos o los trajes de sensores. Los dispositivos de grabación de video se han vuelto cada vez más barato, hasta estar al alcance de todos. De hecho, la mayoría de persona hoy en día disponemos de un dispositivo de grabación en nuestro bolsillo. Es por ello por lo que el uso de estos en los entrenamientos es cada vez más frecuente (1). Estos normalmente se utilizan para un posterior análisis de la realización del deporte practicado. Por supuesto, este análisis se realiza con el objetivo de que en el futuro puedas mejorar la forma de practicarlo.

Otro de los usos de estas grabaciones es para entrenar a los entrenadores. Ya que el análisis de videos es un conocimiento fundamental para elaborar tus tácticas propias en el futuro. Es por lo que se trabaja en métodos de clasificar las secuencias de video en la estructura táctica que existe en él. Por ejemplo, en el tenis se analiza el movimiento de los jugadores y de la pelota para descubrir el comportamiento del jugador. De esta forma tú puedes averiguar la táctica que sigue dependiendo de la posición que este la bola o los otros jugadores (1). Esto se puede conseguir con un dispositivo móvil combinado con un ordenador para un análisis posterior. Los entrenadores también lo que suelen hacer es poner diferentes cámaras alrededor del atleta. De esta forma consigue captar los movimientos que este hace desde diferentes ángulos. Lo que le permite al entrenador un mejor análisis y más profundo. Los entrenadores suelen usar estos métodos principalmente con dos objetivos fundamentales. Estos son analizar a los competidores de su deportista o analizar sus propios videos para encontrar fallos en el desarrollo del deporte.

También en este ámbito nosotros podemos encontrar los sensores que podemos equipar a nuestro cuerpo. Estos tienen la ventaja de que permiten un análisis en tiempo real. No como el análisis de video que mencionamos anteriormente. Ya que ese tipo de análisis no se puede realizar en tiempo real, sino que necesita que sea posterior a haberlo grabado (7). Esto es porque necesita toda la información de los atletas para poder clasificar su táctica. Además, está diseñado para que los atletas puedan utilizarlo sin depender de estar en un espacio cerrado. Sino que estos sensores permiten ver la realización del deporte al aire libre. Esto es fundamental para deportes que se practican en este tipo de ambiente. Por ejemplo, en deportes como en el atletismo o en el piragüismo, aunque se puedan realizar videos. Las cámaras se necesitan que sean móviles y a lo mejor no captas el plano que quieres ver. A parte, necesitas de mucho más equipamiento como buscar una forma de seguir al deportista. Y debes de ir grabándole en ese ángulo mientras que lo vas siguiendo. Por ejemplo, en atletismo o en piragüismo esto se puede hacer mediante un vehículo teniendo un copiloto que vaya grabando. También estos son fundamentales en deportes condiciones extremas como natación o esquiar. Donde puede ser incluso más complicado realizar grabaciones. Donde hay que tener en cuenta que los sensores resistan las temperaturas o el agua (7). Estos permiten describir los movimientos o la característica en el ambiente en el que se entrenan estos deportes. Es por ello por lo que se ha elegido este método para poder analizar el movimiento de paleo de los piragüistas.

Estos por supuesto también tienen limitaciones, que son difíciles de arreglar. Como si existen los elementos ferromagnéticos, los sensores con magnetómetro pueden fallar (7). Ya que, estos pueden distorsionar a las medidas que se hacen en los sensores que toman medidas inerciales. Introduciendo un error en estas que afecta mucho en la precisión de los datos medidos. Ya que con los valores de este se calcula el norte magnético. Que es el que se utiliza para saber cuál es el vector de la gravedad. Este vector es el que se toma como referencia para calcular las medidas. Por lo tanto, si el vector tiene errores estos se propagarán a las demás medidas. Este error se produce en la extrapolación de los datos procedente de las medidas de la aceleración. Además, los sensores inalámbricos suelen tener una pérdida de la señal que emiten. Lo que se traduce en que existen medidas incorrectas debido a que la transmisión de los datos ha fallado (7). Los sensores necesitan ser inalámbricos en la mayoría de los casos para permitir la movilidad de los deportistas. Aunque, por supuesto, hay que tener en cuenta estos fallos en la comunicación. Ya que al ser inalámbricos hace que haya más fallos en la comunicación.

Vamos a empezar hablando de los sensores inerciales, que son los más utilizados en este ámbito. Esto es debido a que estos permiten la descripción completa de los movimientos de los atletas. Estos tipos de sensores contienen a su vez diferentes dispositivos dependiendo del tipo de sensor que sea. Por ejemplo, los IMU (Inertial Motion Unit) solo tienen un acelerómetro y un giroscopio. También existen más variedades de sensores inerciales. Que además de tener los sensores anteriormente mencionados, incluyen un magnetómetro. Los datos que se producen en este se combinan con los del acelerómetro y del giroscopio para dar medidas más precisas. Ya se ha observado como estos pueden ser utilizados para prevenir lesiones. A continuación, vamos a ver como estos dan información acerca de los entrenamientos de diferentes deportes. En voleibol se utilizan este tipo de sensores para ver la altura de salto que pueden alcanzar los deportistas de forma precisa (7). Se hace de tal forma que se colocan estos sensores a varios deportistas a la vez mediante el sistema IMU VERT. Se hace de esta forma para evitar usar el método óptico. Debido a que los marcadores que este necesita para tomar las medidas podrían ser obstruidos de la visión de la cámara. Incluso estos podrían ser tomados como la referencia actual en salto que existe en el voleibol. Esto es debido a que estos permiten poner un umbral que permite considerar que se considera un salto vertical y que no lo es (9). Esto no te lo permite los sistemas ópticos que tienen una sensibilidad más baja. Esto puede provocar que se consideren saltos verticales a aquellos que no lo son. Además, estos son comerciales. Por lo tanto, están disponible para que los diferentes equipos de voleibol lo puedan obtener.

Hay más formas de usar esta clase de sensores de la que se dijo anteriormente. Pero en este caso se centran en un único movimiento del cuerpo. Como él que se ha mencionado anteriormente del salto vertical con el sistema IMU VERT. Pero también podemos utilizar varios de estos sensores para representar movimientos más complicados. De esta forma conseguimos formar un traje de sensores que describa los movimientos de articulaciones. Para esquiar se han hecho trajes de sensores que permiten ver la cinemática que siguen la parte inferior del cuerpo (7). Este te permite observar los ángulos de las articulaciones y sus velocidades. También te permite ver el ángulo, la fuerza y la velocidad del esquí y el centro de masa del conjunto. También nos encontramos con el caso en el caso que implementa para el estudio del cuerpo completo. Es el caso de Canoe-sense, que un sistema de monitorización de la cinemática del cuerpo para canoa (10). Para permitir ver características de las fases del movimiento de la palada. En concreto se realiza un estudio para una canoa de competición de equipo de dos personas. En este caso los sensores son colocados en el torso y en ambos brazos. Además, se incorpora un sensor al antebrazo y a la pierna predominante de cada deportista. Además, se incorpora esta clase de sensores a su material, tanto al barco como a sus palas. De tal forma, que se mide la cinemática tanto de los participantes como de los materiales. Todos los sensores que son colocados tienen un transceptor de radiofrecuencia. Que se conecta a un nodo receptor que está controlado por un PC (personal computer).

Se utilizan diferentes tipos de sensores para describir las actividades realizadas. Por ejemplo, se utilizan una combinación monitores de tasa cardiaca y GPS (Global Positioning System) para medir a los futbolistas australianos (7). Esto se hace para medir la carga de trabajo de los jugadores. La tasa cardiaca no permite observar la carga de trabajo por sí solo. Pero combinando esta con el GPS que permite observar la trayectoria de los ejercicios realizados. Gracias a estos datos se pudieron sacar conclusiones entre el éxito de un equipo y la carga de trabajo que estos tienen. En este tipo de clasificación de sensores también podemos encontrar otros como pueden ser conta-pasos o monitores de energía. Toda esta clase de sensores solo dan una característica muy general del movimiento que realizan los atletas. Por lo que están muy limitado a la hora de dar información de cómo se debe hacer los movimientos. Otros tipos de sensores son los sensores que permiten medir las fuerzas externas que se puedan aplicar sobre un objeto. Estas fuerzas se estiman resolviendo un problema dinámico inverso. Para ello necesitas la información que te proporcionan los sensores de presión (11). Esta información te proporciona una aproximación de la tasa de carga que se ha producido en el objeto. Algunos ejemplos donde se puede utilizar esta clase de sensores son los siguientes. Se puede aplicar para observar la fuerza que se ejerce sobre una jabalina durante el tiro de esta. También se puede observar la fuerza que se hace sobre un cuerpo humano cuando hace un salto de esquí. También esto se ha utilizado en el piragüismo para observar la fuerza que se realiza tanto en la pala como en la pedalina. La pedalina es la parte del barco donde colocas los pies. Y

en barcos de competición es donde iría el timón. Para ver la fuerza con la que va el remo y que haces con las piernas relativamente. Este tipo de sensores te permiten ver la fuerza ejercida con el movimiento realizado. Aunque así, este tipo de sensores tampoco te permiten ver con claridad todos los movimientos que realiza el deportista. También podríamos utilizar al sensor flexible para realizar este tipo de funciones (7). Sin embargo, como se ha podido observar estos están limitados a condiciones de laboratorio. Es por lo que esta clase de sensores tampoco servirían para el escenario reales de algunos deportes.

También podemos encontrar que este es usado en combinación con los sensores inerciales por ejemplo se ha aplicado en el golf (12). En este caso se ha utilizado en la cadera para averiguar los movimientos que hace esta articulación. Aunque este solo se puede doblar en una sola dirección en este caso. Por lo tanto, solo detecta el movimiento en una sola dirección. Como también es el caso del sistema *e-kayak* (13). Que haya tantos los parámetros dinámicos de los materiales de la palada como los cinemáticos. Puede calcular tanto la fuerza que se realiza con la pala y en la pedalina. Además, calcula datos como la frecuencia de la palada, la velocidad y aceleración del barco. Estos sensores van conectados por wi-fi a un ordenador y sus datos pueden dar información en tiempo real. Además de esto también utiliza GPS para calcular la posición y el recorrido utilizado.

En algunos deportes como el piragüismo no siempre se dan las condiciones necesarias para practicarlo. Ya sea porque se ha congelado el río o las condiciones meteorológicas no acompañen. En ese caso, se puede simular el medio, en el caso del piragüismo tenemos los ergómetros (14). Estos son capaces de simular las condiciones del agua mediante un motor y una cuerda. La cuerda está atada tanto al motor como a cada extremo de la cuerda. Y a través de una señal eléctrica que aplicas al motor puedes simular la fuerza de arrastre del agua.



Ilustración 3.2-3. Ergómetro de piragüismo (15).

También nos encontramos con otros tipos de sensores aparte de los IMU. Como son los marcadores catadióptricos que permite ver la cinemática del cuerpo humano. Estos sin embargo no se pueden realizar en escenarios reales. Este se ha usado para medir y compara movimientos de diferentes piragüistas. Que realizaban una simulación de movimientos que se realizarían en piragüismo en un laboratorio (16).

3.3 Piragüismo

El piragüismo como deporte surgió a mediados del siglo XIX. Fue John MacGregor quien creó un kayak con un fin recreativo (17). Este deporte se realiza sobre una embarcación a través del uso de una pala. Esta pala tiene una doble hoja en cada extremo de la pala. Estas hojas tienen una inclinación entre sí. De tal manera, que se va empujando el agua con las hojas, que se van alternando con cada palada. Al final, esto resulta ser un movimiento cíclico que se va repitiendo en un periodo de tiempo. Este deporte se puede realizar de varias maneras. Y estas serían piragüismo de aguas bravas, aguas tranquilas, Kayak polo. Pero a nosotros nos interesa el piragüismo de aguas tranquilas, en concreto, en competición. A tipo de embarcaciones que se utilizan en este deporte se llama piraguas. Y las piraguas consta de un asiento que se encuentra en medio de la piragua. Este se encuentra de tal forma que el deportista se siente mirando hacia la proa del barco. En las piraguas de competición, están tiene una pedalina para apoyar los pies, donde se encuentra el timón. Este timón es una barra, normalmente hecha de algún tipo de metal. Este timón está conectado directamente a una especie de aleta que sale en la parte de debajo de la popa. Y esta se mueve junto al timón, ya que ambos están conectados mediante dos poleas y una cuerda. El timón se giró con los pies, esto es así para que no se tenga que variar la palada en ningún momento. Ya que la otra forma de cambiar de dirección en piragüismo es remando solo con una hoja. O al menos, haciendo una buena palada con mucha fuerza en una hoja. Este sistema evita que la palada cambie, aunque se tenga que realizar un cambio de dirección.

Ritmos	Paladas/minuto	Distancia / Tiempo aproximado
R1	55-65	Más de una hora
R2	70-80	8000 – 10000 m
R3	85-100	1500-4000 m
R4	Max	1'30'' – 4'
R5	Max	30'' – 1'
R6	Max	5'' – 20''
R7	Competición	Cambio de ritmos

Tabla 3.3-1. Ritmos a los que se puede remar y distancia o tiempos para los que están diseñados (18).

En piragüismo existen diferentes formas de remar, dependiendo de cómo ejecutes la palada. Y se ha desarrollado una técnica en cuanto a la forma de remar, para realizar menos esfuerzo. Esta técnica no tiene por qué aplicarla todo el mundo de la misma manera. También depende de las condiciones en las que estes remando y del ritmo. Y existe la posibilidad de que existan diferentes tipos de técnicas dependiendo de la persona. Para empezar, a hablar sobre la técnica primero vamos a realizar un análisis sobre la palada. El movimiento de paleo consta a su vez de dos fases, estas son la fase acuática y aérea (19). Dentro de la fase acuática nos encontraríamos otras tres fases.

- Estas serían la fase de ataque, que es cuando clavamos la pala en el agua.
- La fase de tracción, que es cuando nos empujamos al agua para impulsarnos.
- Y, por último, la fase de la salida, en la que sacamos la hoja de la pala del agua.

Por otro lado, en la fase aérea se dividiría a su vez en otras dos fases.

- Estas serían la fase de subida, consistiría en subir la pala procedente de la fase de salida hasta cierta altura. Mientras que la otra hoja de la pala va bajando hacia el agua.
- Y posteriormente, tendríamos la fase de ataque aéreo que es justo antes que la de ataque acuático.

En el paleo, el tronco en la fase de tracción es el que debería realizar todo el esfuerzo. Mientras que los brazos

solo deben de funcionar como punto de apoyo del movimiento. De esta forma, el deportista se cansará menos al realizar el ejercicio. Esto es debido a que tiene más músculos en el torso que en el resto del cuerpo. Para ello las piernas se deben de mover de forma que permita el movimiento del torso en el asiento. La salida debe de costar lo menos posible, debido a que aquí es donde realizamos más fuerza. Ya que en caso de que el movimiento estamos intentando que la pala que empujaba agua sacarla del agua. Por lo tanto, es como si estuviera levantando esta cantidad de agua, por lo tanto, es donde se realiza mayor esfuerzo. Si este movimiento sale mal levantar esa cantidad de agua con cada palada con poca velocidad inicial. Sin embargo, si el movimiento fuera en forma de ovalo continuo, costaría menos esfuerzo. Ya que también usarías la fuerza que has aplicado durante la tracción. La fase aérea tiene que ser lo más rápida posible. Esto se debe a que cuando el barco avanza es en la fase acuática. Por lo tanto, ahí es donde tenemos que realizar todo el esfuerzo. Durante el ataque aérea, el brazo que sostiene la hoja que va a entrar en el agua tiene que ir lo más hacia delante posible. Ya que de esta manera conseguimos un recorrido en la tracción mayor. Debido a que hay que rotar el remo, hay una mano que sujeta la pala y otra que la deja liberada. De esta forma permite que con una mano la hagas girar y la pala gire. Esto es debido a la inclinación que existe entre las hojas. Como referencia de cómo tiene que ser la palada, los piragüistas forman un rectángulo imaginario con la mano (19).

3.4 Sensores

Como mencionamos en el anterior subapartado, se ha utilizado IMU para realizar las mediciones. Por lo tanto, vamos a describir las características de estos y cada una de sus partes. En la actualidad existen circuitos integrados que te permiten cumplir esa función siguiendo la misma lógica. Este tipo de sensores que funcionan con este tipo de circuito lo llamamos **MEMS** (*MicroElectroMechanical Systems*). Ellos tienen un gran número de uso, además de los que ya mencionamos para el deporte. Como ejemplo, hay que decir que se utilizan en ámbitos como la automoción, la navegación o la robótica (20) (21) (22) (23). En el caso de *CanoeSense* utilizan ambos sensores tanto el acelerómetro como el giroscopio (10).

3.4.1 Acelerómetro

Como ya dijimos, los sensores que nos interesa están formado por un acelerómetro y un giroscopio. El acelerómetro mide la aceleración en las tres componentes del espacio. Esto se suele hacer analógicamente a través de un peso que cae sobre un muelle en los tres ejes de movimiento. Este funciona de tal manera que cuando se hace una fuerza sobre la masa, esta provoca un movimiento en el muelle. De esta forma el movimiento que se ha producido en el muelle se puede traducir a una fuerza. Esta fuerza aplicando la segunda ley de Newton se puede transformar en una aceleración. Esa era la forma antigua de hacer los acelerómetros, pero actualmente se utilizan circuitos para realizarlo. Existen varias formas de hacer el efecto del acelerómetro con diferentes componentes eléctricos (24).

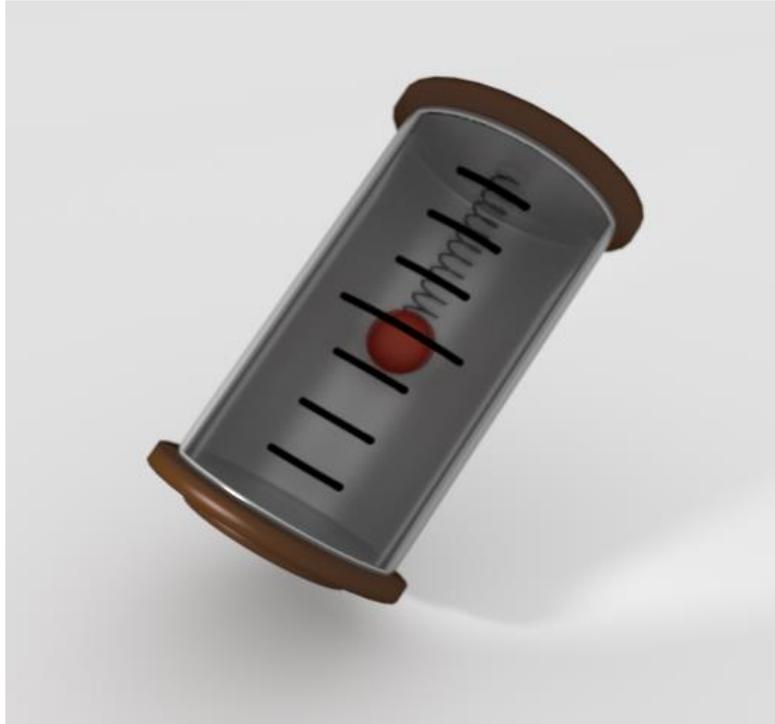


Ilustración 3.4-1. Acelerómetro mecánico actuando en una única dirección (25).

Una forma de realizar los acelerómetros es mediante componentes piezoeléctrico. Estos elementos son unos cristales que al aplicarle una fuerza hace que se genere una tensión en su estructura. Con lo cual obtenemos un mecanismo de a partir de la fuerza que aplicamos nos da una información u otra. Y dentro de hacerlo por componentes piezoeléctrico, existen maneras de realizarlo (26). La primera es mediante una alta impedancia a la salida. Y el cristal piezoeléctrico produce una carga eléctrica que está directamente conectado a la salida. Esto hace que se requiera que la salida y los instrumentos de salidas estén acomodados a este. Esta es la razón por la que se utiliza principalmente en laboratorios. El segundo tipo tiene una baja impedancia a la salida y también tiene una carga a la salida. Pero a este se le puede añadir un microcircuito y un transistor efecto de campo que convierte la carga en baja impedancia. Este es lo que sirve de interfaz a la instrumentación estándar. Por lo tanto, este tipo es lo que se suelen usar habitualmente.

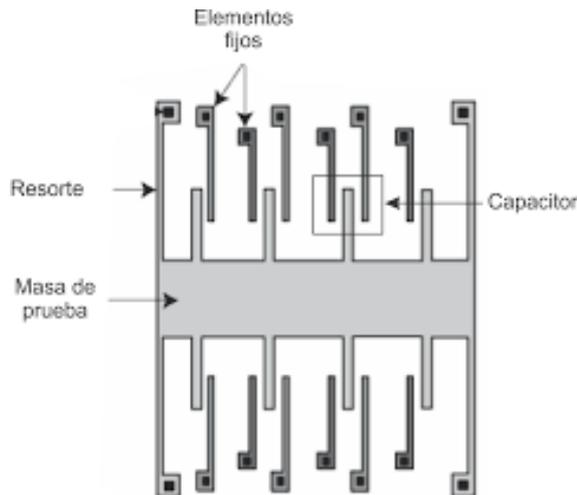


Ilustración 3.4-2. Funcionamiento de acelerómetro MEMS (27).

Otra forma de conseguir el acelerómetro es entre con dos microestructuras una al lado de otra (26). Estas están formando una pequeña capacitancia entre sí. Debido a que al ser tan pequeña la distancia entre las dos, las cargas siguen fluyendo sin aplicar fuerza. Por lo tanto, cuando se aplica una fuerza sobre esta la distancia aumentará o disminuirá. Con lo cual la impedancia que se genera debido a esa distancia también variará. Por lo que hay tenemos otro mecanismo con el cual podemos crear acelerómetros. Dependiendo de la información que nos proporcionen estos acelerómetros estos pueden ser analógicos o digitales. Serán analógicos si nos dan directamente una tensión que se traduce a un dato de la aceleración. Y será digital si la información se modula mediante **PWM** (*Pulse width modulation*) en la salida. Esto significa que a la salida de tensión habrá una señal cuadrada cada cierto tiempo. Dependiendo del tiempo que se mantenga alta la tensión esto se traduce a lo alto que ha sido la duración de esta.

3.4.2 Giroscopio

El giroscopio te da la velocidad angular que se ha producido sobre los tres ejes del espacio. Estos dispositivos utilizan la gravedad de la Tierra para calcularlo (28). Existen tres tipos de giroscopio el rotatorio, el vibratorio y el óptico. Aunque se puede decir que el primer tipo de giroscopio va incluido en el segundo ya que utilizan los mismos principios. Pero normalmente estos tipos de giroscopios se tratan por separados, aunque se metan en el mismo grupo. El rotatorio es el que es más clásico y consiste en un disco que rota libremente, llamado rotor. Este está montado sobre un anillo candal que va girando. Este sería básicamente el giroscopio rotatorio en una sola dirección. Está basado en el momento angular del rotor se mantiene, aunque el anillo se incline (29). Y se comprobó que el rotor mantenía su inclinación inicial a pesar de la rotación de la Tierra. Este sin embargo no se suele usar en la actualidad por el gran coste que supone este y su gran tamaño. Este supone una gran limitación en algunas aplicaciones que se ha visto (9) (12) (20) (22).

Ahora vamos a hablar sobre los giroscopios con estructura vibrante (30). Estos utilizan la naturaleza de una masa sobre un medio elástico para funcionar. Estos tienen dos modos de oscilación. El primer modo es aquel en el que se prepara a la masa a vibrar en todo momento. Este modo lo llamaremos portador, ya que es el que va a estar mientras que el giroscopio funcione. Por lo tanto, el segundo modo es el que nos va a dar información sobre la velocidad angular y que producirá cambios sobre la portadora. Debido a la rotación del giroscopio, este va a provocar el movimiento de la masa en una dirección diferente. Esto es debido a la Fuerza de Coriolis y es lo que llamamos el modo secundario. Esta fuerza permite explicar la relación existente entre el momento angular y la energía cinemática en un movimiento de rotación (31). Esta fuerza inercial, es decir que no es real, permite

explicar los movimientos rotatorios. Esto se puede conseguir de diferentes maneras a partir de diferentes tipos de elementos vibrantes. Según el diseño de los elementos vibratorios que forme el giroscopio. De esta forma podemos encontrar elementos de vibración continua o discreta. Existen los HRG y los que utiliza anillos como ejemplo de giroscopio de oscilación continua. Pero la gran mayoría de esta clase de giroscopio pertenecen a la clasificación de oscilación discreta. Estos elementos vibratorios son una barra, una rueda, un cardan, un diapasón o simplemente una masa con dos grados de libertad. Todos excepto el diapasón incorpora una sola masa en el medio elástico. Estos son muchos más efectivos que los giroscopios convencionales que los explicados anteriormente. Ya que, estos tienen la ventaja frente a los demás tipos que permiten ser miniaturizados. Esto es debido al desarrollo de la microelectrónica ha permitido que la mayoría de estos puedan ser construido. Además, supone un bajo coste, debido a que esta electrónico es barata cuando se construye en masa.

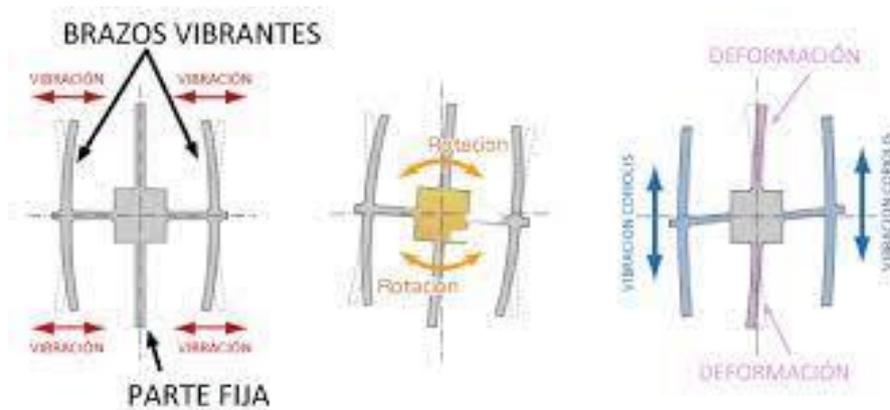


Ilustración 3.4-3. Funcionamiento de un giroscopio vibrante (32).

Para finalizar este tema, hablaremos sobre los giroscopios ópticos (29). La mayoría de ellos están basados en el efecto Sagnac. Este efecto se basa en un rayo de luz va cambiando de dirección para formar un bucle cerrado. Estos rayos de luz se introducen en el sentido horario y antihorario alrededor de una plataforma que permite rotar. Y el rayo que este en el sentido contrario al de la rotación llega más corto al origen. Con lo cual produce patrones de interferencias en los patrones de luz. Esto permite hacer una relación entre estos patrones y la velocidad angular. Por lo tanto, al observarlos podemos determinar de una forma muy precisa la velocidad angular. Es por eso por lo que este tipo de giroscopio se utiliza para la navegación de cohetes y de satélites. Se suelen hacer con rayos láser y espejos para cambiar la dirección, todo en un medio relleno de un gas inerte. Este gas se introduce para que no disminuya la velocidad de la luz afectando a la precisión de este. Otra solución es que introduzcamos en una fibra óptica.



Ilustración 3.4-4. Giroscopio óptico (33).

3.4.3 Procesamiento de los datos

De los anteriores dispositivos se ha sacado tanto la velocidad angular como la aceleración de este. Pero la información que obtenemos de cada dispositivo por lo general no nos interesa. Así que normalmente se combina para de esta forma hallar los datos que nos pueda interesar. Si queremos realizar un seguimiento del movimiento los parámetros que normalmente nos interesa son los ángulos y la posición (34). Esto es debido a que por el teorema de Chasles, que indica que cualquier movimiento se puede descomponer en dos. Para ello que nos interesa tenemos que integrar los datos que nos dan ambos dispositivos (35). En el caso del acelerómetro tenemos que hacer una integración doble para que nos de la posición. Y el caso del giroscopio tenemos que integrar una vez para hallar el ángulo en el que se encuentra. Esto hace que nos de errores al integrar que se propagan con el tiempo. El error que nos importaría sería un movimiento de rotación de este tras realizar el movimiento de translación. Ya que nos vamos a centrar en la rotación del propio sistema de referencia. Para obtener la información acerca de este movimiento se puede hacer uso de un gran número de herramientas matemáticas. Para el caso de sensores inerciales se suele trabajar con los ángulos de Euler o a través de los cuaterniones.

3.4.3.1 Ángulos de Euler

Los ángulos de Euler, o de Tait-Bryan, permiten expresar la rotación de un sistema de referencia con respecto a otro sistema (34). Está demostrado que cualquier rotación de un cuerpo sólido puede expresarse por tres rotaciones elementales. Estas rotaciones se realizan en cada eje, permaneciendo este intacto mientras que los otros dos rotan según el ángulo. Estas rotaciones pueden clasificarse según el sistema de referencia sobre el que se aplica la rotación. Serán los ángulos extrínsecos si los ángulos de cada eje se expresan sobre el sistema de referencia inicial. Y serán intrínsecos si estos se expresan sobre cada nuevo sistema de referencia creado tras cada rotación elemental. La diferencia entre los ángulos de Euler y los Tait-Bryan es que unos repiten un eje del movimiento. Mientras que en el otro todos los ejes sobre los que se le aplica son diferentes. Aunque se suelen llamar a todo este tipo de notación como ángulos de Euler, aunque tenga ese matiz. También se puede clasificar este tipo de notación según la secuencia de ejes que vayas rotando (36) (37). Ya que las rotaciones elementales aplicadas no son conmutativas, se necesita saber que secuencia van a seguir estas. En total existen una docena de combinaciones posibles de secuencias de ejes que se rotan. Esto es debido a que no tiene sentido rotar dos veces seguidas sobre el mismo eje. El movimiento de rotación que se realiza geoméricamente se representa mediante una matriz. Esta matriz de rotación se puede formar de diferentes maneras dependiendo de la notación elegida. En el caso de los ángulos de Euler existen diferentes matrices para cada rotación elemental. Además, multiplicando estas matrices de rotación obtenemos la matriz que representa la rotación final del conjunto. Para

obtener esta matriz, las matrices elementales se deben de haber multiplicado siguiendo la secuencia de rotación elegida.

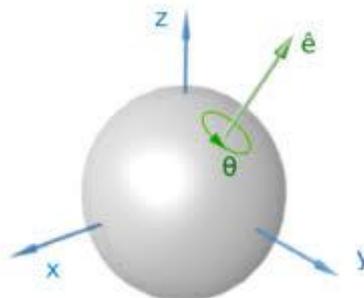
Los ángulos de Euler tienen un grave problema que es llamado como bloqueo de cardán (38). Este se produce cuando el ángulo aplicado sobre segundo eje rotado es $\pi/2$ radianes. Esto se debe a que cuando ocurre esto el primer eje rotado y el tercer eje se alinean. Las consecuencias de esto es que se pierde un grado de libertad en el movimiento en esta representación. Ya que el tercer el tercer eje no haría un movimiento diferente simplemente variaría la primera rotación. Por lo tanto, esa rotación elemental será igual a la suma de los dos ángulos realizados sobre el primer y tercer eje. Sin embargo, esta notación es una forma muy intuitiva de representar las rotaciones. Ya que esta forma de ver los ángulos es más intuitiva es la que nosotros se ha elegido para analizar los resultados.

3.4.3.2 Cuaterniones

Los cuaterniones son una clase de número complejo que están formados por un número real y tres números imaginarios (36). Esta clase de número sigue una serie de axiomas que es lo que lo convierte en cuaterniones. Si se fijan esta clase de número se podría tomar como una generalización de los números complejos. Esta clase de número se suelen utilizar para la descripción de una rotación. Esta es la función que nos interesa a nosotros y la que vamos a utilizar. El cuaternión puede representar cualquier transformación en el espacio de tres dimensiones. Sin embargo, si se quiere representar una rotación del sistema de referencia, entonces el cuaternión es unitario. La representación sería de tal forma que este representa un vector en el espacio y sobre este se gira un cierto ángulo. A partir de esta representación de un cuaternión se puede representar cualquier giro en el espacio. Aunque pueda haber varios cuaterniones que representen el mismo giro. En la Gráfica 3.4-1, se puede ver la representación visual de lo que se ha mencionado anteriormente. El cuaternión se forma de tal manera que la parte real sea el coseno del ángulo rotado. Y la parte imaginaria sea un vector unitario formado con cada variable imaginaria multiplicado por el seno del ángulo. La ecuación (3.4-1) sería la ecuación matemática descrita antes. El módulo resultante sería uno debido a que el módulo del vector es uno. Luego a partir de la ecuación (3.4-2) vemos que el módulo efectivamente el módulo es unitario.

$$q = \cos \theta + \sin \theta (e_x i + e_y j + e_z k) \quad (3.4-1)$$

$$\cos^2 \theta + \sin^2 \theta = 1 \quad (3.4-2)$$



Gráfica 3.4-1. Representación de un cuaternión-Formado por un vector e y un ángulo de giro (39).

La suma de dos cuaterniones se realizaría término por término de los elementos del número complejo. Para la multiplicación, se ha considerado unos axiomas que están representado en las ecuaciones (3.4-3), (3.4-4), (3.4-5) y (3.4-6) (40). Estos axiomas se tienen que cumplir para que formen números imaginarios. Y para que formen un espacio en tres dimensiones y se toma la convención de la mano derecha para los giros. Debido a los axiomas (3.4-4), (3.4-5) y (3.4-6) se puede observar que la multiplicación no es conmutativa.

$$i^2 = j^2 = k^2 = -1 \quad (3.4-3)$$

$$ij = k ; ji = -k \quad (3.4-4)$$

$$jk = i ; kj = -i \quad (3.4-5)$$

$$ki = j ; ik = -j \quad (3.4-6)$$

El giro de un vector se define a través de la ecuación (3.4-7). En el caso de la inversa de un cuaternión se tendría que buscar un elemento que tuviera la forma de (3.4-8). Y podemos llegar a la conclusión que llegamos en (3.4-9). A partir de este cuaternión nosotros podemos hallar la matriz de rotación de la que hablábamos antes. Además, a partir de la matriz podemos hallar el quaternion que representa la rotación. Esto permite que los cuaterniones sean más simple de trabajar con ello computacionalmente. Aunque también estos son poco intuitivos para nosotros cuando trabajamos con ellos. Este tipo de representación no tiene el tipo de problema de bloqueo de cardán que tenían los ángulos de Euler. Debido a esto es por lo que nosotros lo vamos a utilizar a la hora de tomar las mediciones.

$$v^* = qvq^* \quad (3.4-7)$$

$$qq^{-1} = q^{-1}q = 1 \quad (3.4-8)$$

$$q_1q_1^* = |q|^2 \rightarrow q_1 \frac{q_1^*}{|q|^2} = 1 \rightarrow q_1^{-1} = \frac{q_1^*}{|q|^2} \quad (3.4-9)$$

Esta forma de representar ángulos se puede transformar en otra forma de representarlo y viceversa. En nuestro caso, podemos transformar estos cuaterniones en ángulos de Euler. De esta forma, podemos utilizar las ventajas que nos da ambas representaciones. Por un lado, los cuaterniones son más sencillos a la hora de realizar calculo. Sin embargo, son más complicados de representar y más difícil de ver. Para verlo más fácilmente podemos transformarlo a ángulos de Euler. Para ello están las ecuaciones (3.4-10), (3.4-11) y (**3.4-12**). Estas están hechas teniendo en cuenta que el quaternion modifica al vector y no al sistema de referencia.

$$\tan \alpha = \frac{2(q_x q_\omega - q_y q_z)}{1 - 2(q_x^2 + q_z^2)} \quad (3.4-10)$$

$$\sin \beta = 2(q_x q_y + q_z q_\omega) \quad (3.4-11)$$

$$\tan \gamma = \frac{2(q_y q_\omega - q_x q_z)}{1 - 2(q_y^2 + q_z^2)} \quad (3.4-12)$$

3.4.3.3 Calibración

Para que evitar cometer errores, necesita que los dispositivos tengan alineados y escalados sus ejes para que estos sean ortogonales entre sí (35). En todo caso la calibración se realiza en estático para todos los dispositivos. Esto es debido a que para ambos dispositivos utilizamos el vector de la gravedad como referencia. Por lo tanto, Si estamos movimiento, la calibración no tomará los valores correctos. Ya que tomará el vector de la suma de fuerzas del movimiento y la gravedad como si fuera el vector de la gravedad. Por lo tanto, a partir de ese momento todas las medidas que se realicen con el dispositivo serán erróneas. Todos los métodos que existen para calibrar los diferentes dispositivos son muy variados. Aunque todos ellos deben tener un propósito en común para cada dispositivo. En el caso del acelerómetro, este consiste en que la magnitud del vector de la gravedad debe ser igual a la gravedad terrestre. Este número es muy buena referencia ya que es una variable bastante estable. Esto ocurre debido a que la mayoría de las variables no varían independientemente de la situación posibles. Excepto si calibras el dispositivo en algún punto que este lo suficiente alejado del planeta. Aunque este tipo de situaciones no se suelen dar o si se dan son en aplicaciones muy específicas. El objetivo que sigue la calibración del giroscopio se usa como referencia el vector de gravedad dado por el acelerómetro. Este se compara con el vector de gravedad hallado a través de la integración de las velocidades angulares medidas.

Utilizar únicamente estas dos formas de calibración no permitirían aumentar por si solos la precisión. Esto es debido al ruido producido por el sensor y por el problema de que el giroscopio se deja llevar (23). Este último problema se produce debido a la integración de ruidos e imperfecciones en las medidas (41). Estas son principalmente los cambios lentos y los ruidos de muy alta frecuencia. A estos últimos se les llama **ARW** (*Angular Random Walk*). El eje perpendicular a la gravedad es el más sensible a sufrir esta clase de problema. Esto es debido a que estas perfecciones en los ejes, donde está la gravedad, pueden ser contenido. Ya que, en esos ejes podemos utilizar el acelerómetro para obtener la posición relativa de la gravedad. Otra forma de cancelar este error es utilizar un filtro paso bajo. Además, los acelerómetros son sensible a las vibraciones y a cualquier fuerza que no sea gravitatoria. Aunque el otro al irse sumando produzca con el tiempo los errores más grandes.

Una forma bastante utilizada a la hora de medir el error es la de utilizar un magnetómetro (41). Este dispositivo te permite saber dónde está el norte magnético de la Tierra midiendo el campo magnético de este. Este se utiliza sobre todo para evitar el error del eje perpendicular a la gravedad. Por lo tanto, este fusiona los datos juntos a otros dispositivos para de esta forma poder reducir el error del giroscopio. Este también permite el mal alineamiento entre los ejes de los sensores inerciales y los del magnetómetro. Este tipo de sensores no pueden operar cuando en el entorno existen campos magnéticos tan intenso como el del norte. Estos son muy sensibles a elementos ferromagnéticos que puedan existir en la zona (23). Además, estos necesitan ser recalibrados cada vez que el entorno magnético cambia (42).

Existe otra forma de evitar este tipo de error de una forma más eficiente y sin usar otros dispositivos (41). Para ello se actualiza el valor de la velocidad angular a nulo cuando sepamos que este no está rotando. De esta forma conseguimos que el error no se vaya propagando y así evitar graves errores. Aunque es cierto que este método es muy eficaz sí que es cierto que esto solo se puede hacer en ciertas aplicaciones. Esto es debido a que en aplicaciones como la nuestra el objetivo no se mantiene inmóvil en ningún momento. Este método viene muy bien en aplicaciones como en automovilística que se puede saber cuándo se detiene el vehículo.

3.4.4 IMU MPU-6050

Este **IMU** fue inventado por *Invensense* e incorpora tres diferentes sensores dentro de la unidad (43). Tendría un giroscopio, un acelerómetro y un termómetro. Todos ellos activas junto al **DMP** (*Digital Motion Processor*) consumirían como máximo 3.5mA. Lo que vamos a ver es cómo funcionan el giroscopio y el acelerómetro. Esto se debe a que son estos los que vamos a usar en nuestro proyecto. Cada uno de ellos tiene tres grados de libertad. Tanto el acelerómetro como el giroscopio están formados por **MEMS** (44). En el caso del acelerómetro funciona como un sistema de masas con resortes. Y en el caso del giroscopio funciona haciendo uso del efecto Coriolis. El rango del acelerómetro y del giroscopio puede ser ajustado (43). Por defecto, los valores del rango serían $\pm 2g$ y $\pm 250^\circ/s$ en el acelerómetro y giroscopio, respectivamente. Además, tiene unos conversores analógico-digitales de 16 bits.

Giroscopio	$\pm 250 \text{ }^\circ/s$	$\pm 500 \text{ }^\circ/s$	$\pm 1000 \text{ }^\circ/s$	$\pm 2000 \text{ }^\circ/s$
Acelerómetro	$\pm 2 \text{ g}$	$\pm 4 \text{ g}$	$\pm 8 \text{ g}$	$\pm 16 \text{ g}$

Tabla 3.4-1. Rangos de los sensores del MPU-6050



Ilustración 3.4-5. IMU MPU-6050 (45).

El sensor mide en analógico y lo pasa a digital para poder interpretarse a través de estos convertidores. Por lo tanto, las medidas van a ir desde los -32768 hasta los 32767 en binario. La medida del sensor en bruto daría un número entre esas dos cantidades. Para calcular la medida física, dividimos el rango completo entre el rango de la medida. Entonces esto nos diría cuántas unidades físicas vale un bit de la medida. Si os habéis dado cuenta esto significa que cuanto más rango dispongamos, menos exacta será la medida. Si lo multiplicamos por la medida en bruto nos daría la medida física del sensor. Sin embargo, en pocos casos nos interesa directamente estas medidas físicas. Sin embargo, a partir de ellas sí que podemos calcular otras medidas que nos interese más, en nuestro caos el ángulo. Para ello se aplica la ecuación (3.4-13) con los datos dados del giroscopio y posición inicial (44). Esto nos daría una medida teórica del ángulo, sin embargo, debido a *drift* esto no se puede aplicar directamente. El *drift* es un error sistemático de los giroscopios que hace que al cabo del tiempo las medidas no valgan. Para evitar esto se propone hacer un filtro paso de baja y de alta. Para realizar estos filtros hace falta que el. Para ello calculamos el ángulo a través del acelerómetro y del giroscopio y a cada uno le damos una ponderación. Estas ponderaciones son normalmente de 0.98 al giroscopio y 0.02 al acelerómetro. Estas medias son sensibles a fuerzas externas y al movimiento de la IMU. Sin embargo, su error no se acumula con el tiempo a diferencia del *drift*. Por lo tanto, poniendo ponderaciones bajas al acelerómetro, ponerle un filtro paso baja. De manera que cuando la variación de este sea muy grande debido a una fuerza externa, estos ángulos valgan poco. Sin embargo, los del giroscopio tenga un mayor peso.

$$\theta = \theta_0 + \omega\Delta t \quad (3.4-13)$$

Esta permite la comunicación a través de buses **I2C** (*Inter-Integrated Circuit*) y permite dos direcciones. Estas serían las direcciones 0x69 y 0x68. Permitiendo de esta manera que dos de este tipo de dispositivos puedan ser conectados en el mismo bus. Además, puede ser maestro o esclavo del bus. Lo que permite la adición de otros sensores a este. Sobre todo, lo que se conecta como esclavo sería un magnetómetro. Los datos de este pueden ser procesados dentro de la **IMU** a través del **DMP**. El **DMP** es un procesador que calcula los datos a partir de las medidas del acelerómetro y giroscopio. Además, puede fusionar hasta con 9 grados de libertad, si incluyéramos un magnetómetro (46). Este contiene menos errores que los filtros presentados anteriormente y son calculados internamente. El valor calculado en este es recibido a través de la lectura del registro asociado a este. Sin embargo, no hay mucha información acerca de este procesador. Esto es debido a que el fabricante no ha querido dar información acerca de este. Se debe de tener en cuenta a la hora de usarlo, que antes de empezar a funcionar debes dejar que se estabilice. Este proceso puede durar entre 10 a 40 segundos (44).

3.5 Protocolo I2C

I2C es un tipo de bus que permite una comunicación serial entre diferentes dispositivos. Se ha popularizado para la comunicación de los controladores con diferentes sensores, como es nuestro caso. También se le puede conectar a otros tipos de dispositivos como es el caso de las memorias. Este tipo de bus fue diseñado por la compañía Philips en 1982 (47).

3.5.1 Buses

La comunicación está formada por dos buses y por supuesto, una línea de tierra. Estos dos buses son de tipo de drenaje abierto (48) y están disponibles si están a un estado lógico alto. Por lo tanto, deben estar conectados a una fuente de alimentación a través de una resistencia cada una de ella. Las resistencias pueden variar de los 1,8 K Ω a los 47K Ω . Cuanto menor sea el valor de la resistencia, más aumenta el consumo. Sin embargo, la sensibilidad al ruido disminuirá y mejorará el tiempo de los flancos de subida y bajada. Los buses se llaman **SDA** (*System Data*) y **SCL** (*System Clock*). Como su nombre indica van referido a los datos y a la señal de reloj, respectivamente.

3.5.2 Dispositivos

En los buses se pueden conectar físicamente tantos dispositivos siempre que la capacidad máxima no supere los 400pF (48). La velocidad de transmisión estándar entre los diferentes dispositivos es de 100 Kbit/s. Aunque existen modos de alta velocidad, que tienen velocidades de 1Mbit/s, 3,4Mbit/s y 5bit/s. Los dispositivos pueden tomar dos roles principales, maestro o esclavo. Puede haber tantos dispositivos con capacidades de maestros y esclavos como se deseen. Pero solo puede haber un dispositivo con rol de maestro por conexión. El protocolo solo te permite acceder a 112 dispositivos esclavos conectados por maestro (47). Por lo tanto, por el protocolo no se podrían acceder más que a esa cantidad de dispositivo por maestro. Aunque en realidad son 7 bit de direccionamiento, lo que permitiría 128 direcciones. Pero de 112 hasta 128 está reservado para direcciones especiales. Se puede cambiar de maestro, habiendo terminado uno su conexión y empezando el otro la siguiente. El fabricante de los muchos dispositivos esclavos le ponen su propia dirección. Estos dispositivos pueden ser memorias o sensores, como es en nuestro caso.

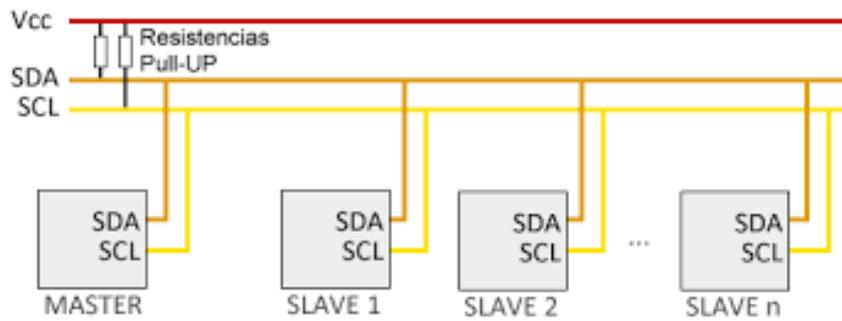


Ilustración 3.5-1. Colocación de los diferentes dispositivos dentro del bus I2C (49).

Los dispositivos con capacidades de maestro deben de poder realizar las siguientes funcionalidades (48). Es el encargado de empezar y finalizar las comunicaciones con los esclavos. Después debe de ser capaz de tener una señal de reloj que poner en el bus SCL. La señal del comienzo sería poner en nivel lógico bajo en el bus SDA con el bus SCL en nivel lógico alto. Y la señal de finalización sería subir el nivel lógico del bus SDA con el bus SCL en nivel lógico alto. Además de ser capaz de leer y escribir en el bus SDA. El esclavo solo tiene que aprender a sincronizarse la señal de reloj dada y escribir y leer en el bus SDA. No pueden ni escribir comandos ni realizar un pulso de reloj.

3.5.3 Protocolo

Una vez que el maestro pone la señal de comienzo, todos los esclavos se quedan atentos al siguiente byte (48). Debido a que en este se indican el esclavo con el que se va a comunicar el maestro. Los siete primeros bits del primer byte indican la dirección del esclavo para identificarlo en el bus. En el bus I2C, se pueden realizar dos funciones lectura y escritura. El maestro indica la función de escritura o de lectura en el octavo bit del primer byte enviado. La función de escritura o de lectura se refiere a si el maestro lee o escribe en el bus. En cualquier caso, el esclavo hará la función contraria para que funcione el intercambio de datos. El esclavo mandará un asentimiento si ha recibido el mensaje si se puede comunicar con él. Para ello, este pondrá en un estado lógico bajo el noveno bit, que será leído por el maestro. Ya sea el maestro o el esclavo quien escriba, siempre envía un byte y el noveno será de asentimiento. Se escribe de tal forma que el primer bit es el más significativo dentro del byte. Si el asentimiento se pone en alto, el maestro debería cortar la comunicación con una secuencia de finalización. El siguiente byte es la dirección del registro del esclavo en el que el maestro quiere leer o escribir. Esta dirección dependerá de cómo este diseñado el esclavo al que direcciones. Luego se transmiten los datos que se quiere introducir en el registro o los datos de este. Existen casos en el que el maestro escribe y tiene que introducir más de un byte en un registro. Entonces, el esclavo tiene que ir actualizando la dirección del registro donde se escribe. Esto se realiza para almacenar los datos y no sobrescribirlos. Cuando se transmiten todos los datos, el maestro tiene dos opciones posibles. O bien pone condición de finalización o la de inicio de nuevo e inicia otra comunicación.

En el caso de la lectura de un registro del esclavo en concreto, el maestro deberá indicar dicho registro. Para ello el esclavo tendrá un registro que indica la dirección interna del registro que el maestro va a leer. Entonces, el maestro escribe en el registro anterior, la dirección del registro que desea leer. Para ello tiene que realizar el proceso de escrito descrito anteriormente. A continuación, se envía una señal de inicio y se manda otro byte con la misma dirección. Pero esta vez el octavo byte indicará leer el registro que se ha indicado en el paso anterior. Hay un problema cuando el dispositivo que escribe es el esclavo y no está listo para enviar la información (48). Esto ocurre cuando este dispositivo es importante como un microprocesador. Y es que para que el maestro espere al esclavo el esclavo puede mantener el bus SCL en nivel lógico bajo. De esta manera, el maestro dejará de aplicar la tensión para poner el bus SCL a nivel lógico alto. Pero se encontrará que este no sube ya que lo mantiene abajo el esclavo.

3.6 Microcontroladores

Los microcontroladores son circuitos integrados que tiene todos los elementos básicos de un PC. Por supuesto, estos componentes son de menores dimensiones que los que tiene un ordenador. Y, por lo tanto, tampoco tienen las mismas capacidades que tendría un ordenador. Estos componentes serían una unidad de procesamiento, módulos de memoria y periféricos I/O (Entrada/Salida) (50). Este ni siquiera tiene interfaz de usuario, pero puede cumplir el rol de controlador del resto de dispositivos. Y, por lo tanto, él junto a algunos periféricos pueden formar un sistema por sí mismos. Este sistema solo tendría una función en particular, no como los ordenadores que tienen un uso general. Esto es perfecto para aplicaciones que necesiten una electrónica adecuada en un lugar específico. Además, estos son mucho más barato y hace que sea asumible en muchas aplicaciones. Estos antes eran mucho más difíciles de programar, ya que la mayoría de ellos se programaban en ensamblador (50). Al ser muchos de ellos lenguajes propietarios y a muy bajo nivel era difícil aprenderlo. Además, se ser difíciles de programar a ese nivel. Pero ahora podemos programar en lenguajes de más alto nivel como C. E incluso en un lenguaje interpretado como Python.

3.6.1 Componentes

Los elementos que forman un microcontrolador ya lo dijimos anteriormente. Estos son una unidad de procesamiento, módulos de memoria y periféricos I/O. Son los elementos básicos para que un dispositivo procese datos del exterior. Ahora vamos a explicar cada uno de ellos y las características que suelen tener en un microcontrolador (50).

- Unidad de procesamiento. Este es el encargado de ejecutar las instrucciones cargadas en memoria. Entre sus funciones incorporaría procesos aritméticos, lógicos y operaciones de entrada y salida. Aunque estas no son las únicas, también estarían operaciones de transferencia. Esta comunicación contendría comandos hacia otros dispositivos del sistema, como a periféricos. Normalmente estos son procesadores de 4 bits, 8 bits y hasta 16 bits. Y después existen microcontroladores muchos más avanzado de 32 e incluso de 64 bits (50).
- Módulos de memoria. Estos almacenan tantos datos que hay que procesar o almacenar. Además, también contiene las instrucciones que tiene la unidad de procesamiento. Por lo tanto, esto hace que existan dos tipos de memoria. Cada una de ellas son de un tipo diferente en función a sus necesidades. Estarían las memorias de programas que son las que almacenan las instrucciones. Estas memorias son no volátiles, es decir, mantienen la información sin necesidad de alimentación. Al principio, estas memorias serían del tipo **ROM** (*Read Only Memory*) (51). Es por ello por lo que el programa sería impuesto por el propio fabricante. Ya que el programa no se podía borrar, por lo que no era programable. Sin embargo, con la tecnología flash empezaron a poder borrarse y ser escribible los programas. Ahora también existen memorias tipo **EPROM** (*Erasable Programmable Read only Memory*) y **EEPROM** (*Electrically Erasable Programmable Read only Memory*). Después estarían las memorias para datos que son tipo **RAM** (*Random Access memory*). Es decir, a diferencia de la de programa estas son volátiles. Ya que esta solo quiere ser almacenada mientras que se está ejecutando el programa.
- Periféricos I/O. Estas son las fuentes y destinatario de la información procesada por la unidad de procesamiento. Por lo tanto, ellos son la interfaz con el exterior. Estos pueden ser de muy diversos tipos. Se suele utilizar la comunicación a través de pines para conectarlo al microcontrolador.

Estos son los básicos elementos que puede tener un microcontrolador. Además, puede incluir muchos más ya dependiendo de las características que tenga cada uno. Además de los básicos normalmente también viene incluidos convertidores. Estos son tanto del tipo de analógico a digital que de digital a analógico (50). El primero se necesita para sacar cualquier tipo de medida física del exterior. Ya sea temperatura, sonido, voz o aceleración. Después, tendríamos el segundo que sería para transmitir algún tipo de señal a partir de unos datos. Como puede ser girar algo un cierto ángulo. También incluyen normalmente puertos series para comunicarse entre dispositivos.

3.6.2 Procedimiento

Como ya se ha dicho los procesadores la mayoría del tiempo solo sirven para una sola funcionalidad. De tal forma que solo tiene un programa que ejecutar y este se almacena en la memoria de programa. Por lo tanto, el procesador accederá y descifrará a la siguiente instrucción a través de esta memoria. Una vez que sepa la instrucción, está la ejecuta y hará lo que proceda en ella. Esta puede incluir cosas como leer de algún periférico o escribir en el registro de algunos de ellos. En cualquier caso, se necesitaría almacenar las variables leídas o que van a punto de ser escrita en algún sitio. Estas como ya vimos se almacenan en la memoria RAM. Esta información es con la que suele trabajar el procesador mientras que el programa siga ejecutándose. Ellos tienen suficiente memoria como para ofrecer operaciones de transmisión en los periféricos de los pines I/O (50). A veces, se necesita más de un microcontrolador en un sistema. Normalmente, este caso ocurre en un sistema grande, en el que se tenga que ejecutar varias tareas. Entonces, cada microcontrolador se ocupa de una. Entonces tendrán alguna forma de comunicarse entre sí para saber que su tarea ha sido realizada con éxito.

3.6.3 Tipos de procesadores

Se pueden encontrar diferentes tipos de procesadores en los diferentes dispositivos. Vamos a ver dentro de que categorías puede entrar las **MCU** (*Microcontroller unit*). La más básica es la de si es un tipo **CISC** (*Complex Instruction set Computer*) o **RISC** (*Reduced Instruction set Computer*). Ya que elegir como va a ser el conjunto de instrucciones es fundamental para diseñar un procesador. La diferencia entre una y otra es, como su nombre indica, en el número de instrucciones que tiene una y otra. Y es que los **CISC** suele contener alrededor de 80 bits de instrucciones. Mientras que la **RISC** alrededor de 30 bits para instrucciones. Además, Los **CISC** tienen entre 12 y 24 modos de dirección, mientras que el **RISC** entre 3 y 5 (50). Esto influye en la arquitectura del procesador y en que está especializado. Las instrucciones de los **CISC** son instrucciones bastante complejas. Se necesita que la electrónica sea muy avanzada para ejecutar este tipo de instrucciones. Estos serán más lentos en ejecutar este tipo de instrucciones. (52) Esto es debido a que en muchos casos necesitarán varios ciclos de reloj para ejecutar una única instrucción. Sin embargo, el software es más sencillo de implementar y hacer un uso más eficiente de la memoria (50). Esto es debido a que en una sola instrucción puede cargarlo en memoria y realizar la operación conveniente. En cambio, los **RISC** es más complicado de implementar. Ya que hace uso de un software más complicado que hace más eficiente las instrucciones que ejecuta. Esto es debido a que las instrucciones que se ejecutan son más sencillas. Lo que hace que los bloques lógicos sean más pequeños en comparación a los **CISC**. Ya que estos últimos necesitan microcódigos dentro de la ejecución de la propia instrucción. Y esto hace que sea mucho más lento que un procesador **RISC** (52). Además, el diseño del propio microcontrolador será más sencillo que el de un **CISC**. Sin embargo, necesita que se pueda acceder rápido a memoria, lo cual hace implementar memoria cache. Para de esta forma almacenar varias instrucciones. Estos tienen menores consumo energético y necesita poca refrigeración por ello. Por lo tanto, las **MCU** utilizarán tipo de arquitectura **RISC**.

La segunda categoría es la arquitectura que implemente cada microcontrolador. Se va a indicar las principales arquitecturas implementadas. Estas serían la arquitectura de Harvard y la de Von Neumann. En la primera, se permite el bus de datos y de instrucciones van por separado. Mientras que en la de Von Neumann tanto las instrucciones como los datos van por el mismo bus (50). Por lo tanto, la de Harvard permite la transmisión de datos e instrucciones a la vez. Mientras que la de Von Neumann tiene que ser en diferentes tiempos.

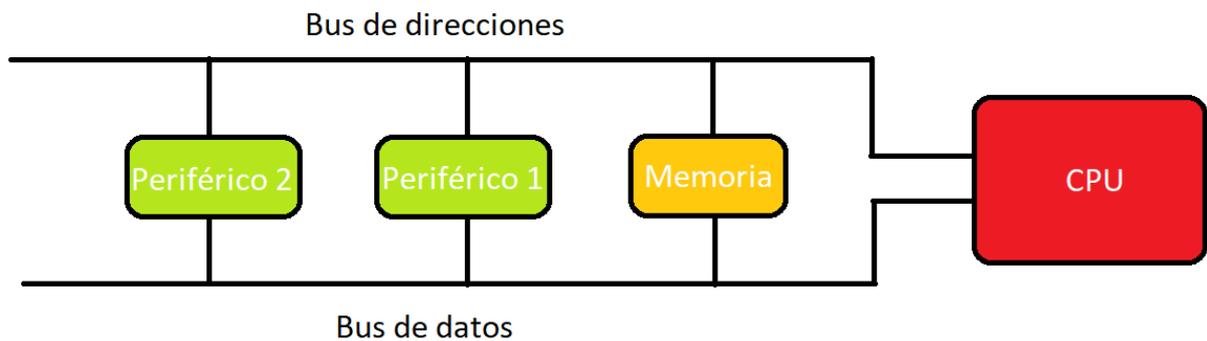


Ilustración 3.6-1. Arquitectura Harvard de un microcontrolador

3.6.4 ESP 32

Este **MCU** fue creada por el fabricante espressif Systems. Este contiene tanto un módulo wi-fi como un módulo Bluetooth. Su procesador puede incluir uno o dos core de **CPU** (*Central Processing Unit*) de 32 bits a una frecuencia de 240Hz (53). En nuestro caso, que es el modelo ESP 32 wroom-32, tiene dos core en la CPU (54). Proporciona de 16 a 24 bits para el conjunto de instrucciones. Lo que los convierte en una **MCU RISC**. Además, puede realizar operaciones con unidades de punto flotante. Esto es algo necesario para la exactitud de los cuaterniones. Incorpora unas memorias **ROM**, para las funciones del core y de reinicio. Incorpora una **SRAM** (*Statical RAM*) tanto para datos como para instrucciones. Y una memoria flash o **PSRAM** (*Pseudo-statical RAM*), el uso de estas memorias dependerá del modelo en concreto que usemos. En nuestro caso no incorpora ni memoria flash ni **PSRAM** en el propio microcontrolador. En nuestro caso nuestro modelo tiene. Además, a través de **QSPI** (*Quad Serial Port Unit*) se puede añadir múltiples memorias flash y **SRAM**. Dentro de los periféricos que incluyen en la propia **CPU**, hay unos cuantos que caben destacar. Lo primero serían los convertidores analógico-digitales y digitales-analógico, que vino en el apartado 3.6.1. También tendríamos 34 pines de entrada y salida de propósito general, para conectar más periféricos. Estos pines se le pueden asignar una función determinada programando el registro que se le corresponde. Además, tenemos una interfaz para las comunicaciones **I2C**, **I2S** (*Inter-Integrated sound*), **UART** (*Universal Asynchronous Receiver-Transmitter*) y **SPI** (*Serial Port Interface*). Tanto en **I2C**, **I2S** y **SPI** se permite que el dispositivo esta tanto en modo maestro como esclavo. También incluye una PWM (modulación por ancho de pulso) para las comunicaciones Bluetooth y wi-fi. Además, tiene un contador de pulso, que captura un pulso y mide los picos que están dentro de él. También, incluye un coprocesador para el modo de sueño profundo del dispositivo. Además, incorpora una interfaz **MAC** (*Media Access Control*) Ethernet, para comunicaciones en una **LAN** (*Local Area Network*) Ethernet.

Este microcontrolador como habéis visto es avanzado y permite la ejecución de los protocolos **TCP/IP** (*Transport Control Protocol / Internet Protocol*). Y Los protocolos wi-fi **MAC** 802.11 b/g/n al completo. Además, este implementa el conjunto básico de servicios de una estación y de un soft AP (punto de acceso limitado). Por otro lado, tenemos que ESP32 permite la versión de Bluetooth 4.2. Y se puede poner tanto en los modos clásico como el modo de baja energía. Permite los modos de conexión **ACL** (*Asynchronous Conection Less*), **SCO** (*Synchronous Connection Oriented*), **eSCO** (*extended Synchronous Connection Oriented*) y **AFH** (*Adaptative Frecuency Hop*). Permite formar y gestionar tanto piconet como scatternet, actuando en modo servidor. Aunque también se puede poner en forma de cliente, según este sea programado. Este puede realizar tanto múltiples conexiones, tanto el modo clásico como en el de baja energía.



Ilustración 3.6-2. Placa ESP 32 WROOM 32 (55).

Las aplicaciones que tiene este dispositivo son muchas y de espectro muy amplio (53). Para empezar, tendríamos el campo del **IoT** (*Internet of Thing*). Este se puede aplicar para un grupo de sensores y registro de datos (56). En el campo de la domótica, tendríamos ejemplos de lámparas, puertas y enchufes inteligentes. Todo esto se puede crear a través de este microcontrolador. También en un edificio se podría controlar la iluminación y la energía que se utiliza. En el caso de la industria se podría aplicar en robots industriales. Además, también podría servir para el control de forma inalámbrica de la maquinaria. En el caso de la agricultura se puede aplicar para regadío e invernaderos inteligentes. Para juguetes de control remoto o con sensores de proximidad. Luego en el apartado médico, nos encontraríamos que se puede utilizar para aplicaciones de monitorización. Además, de crear dispositivos inteligentes que puedas llevar contigo, como relojes inteligentes (57). También podríamos crear dispositivos de reconocimiento de imagen y sonido a través de ellos (58). Además, puedes crear cascos con ellos, o altavoces que transmitan radios y música sacada de internet.

3.7 Bluetooth

Bluetooth permite la transmisión de datos de forma inalámbrica de corto alcance entre diferentes dispositivos. Hoy en día lo podemos ver que muchos de los dispositivos que usamos a diario lo tienen integrado en alguna de sus versiones. Desde dispositivos móviles como son los smartwatches y los smartphones, hasta dispositivo más grande como ordenadores. Incluso el ordenador que incorpora nuestro coche implementa este protocolo. Y es que hoy en día este protocolo es fundamental para nuestras vidas. En este proyecto, lo vamos a utilizar para tener un control sobre los sensores que están alrededor del cuerpo del sujeto.

Para permitir que se produzca esta clase de comunicación entre los diferentes dispositivos se creó un estándar. Este estándar es lo que nosotros conocemos comercialmente como Bluetooth (59). Con estándar nos referimos a una serie de protocolos y reglas que tienen que implementar los dispositivos con Bluetooth. Esto permite que todos aquellos que usen Bluetooth puedan comunicarse entre sí. Esto es debido a que ambos implementan la misma torre de protocolo. Por lo que, los protocolos de las diferentes capas de un dispositivo se pueden comunicar con los del otro. Los creadores de esta tecnología son dos holandeses, Jaap Haartsen y Mattison Steven, que trabajaban en Ericson (60). Este estándar es desarrollado por el **Bluetooth SIG** (*Special Interest Group*) para poder ser utilizado por ellos y otros desarrolladores. Este es un grupo está formado por empresas son líderes mundiales en los mercados tecnológico. Que van implementando esta tecnología en sus dispositivos para diversos grupos. Además, este grupo se encarga de ir actualizando este estándar y tecnología para que se adapte a las tecnologías actuales. Para ello dan especificación sobre los protocolos que utilizan específicos y aquellos generales usados en otras tecnologías. Para que este estándar sea eficiente debe poder ser utilizado por diverso protocolo de capa superior. Esto es debido a que es un estándar del medio físico con lo cual se busca que sea accesible para la mayoría de estos.

Bluetooth se implementa en la banda de frecuencia de **ISM** (*Industrial, Scientific and medical band*) que va desde los 2400MHz hasta los 2483.5MHz (61). Esta banda de frecuencia es ampliamente utilizada por lo tanto

permite que sea compatible con muchas tecnologías. Además, esta banda proporciona un buen equilibrio entre la tasa de datos y el alcance de datos. Esto es debido a que a mayor frecuencia mayor es la dificultad de que este llegue a su destino. Dependiendo de los usos que vayas a usar del estándar así tendrás que ir variando los diferentes parámetros para ello. Estos son la ganancia de la antena, la potencia del transmisor y la versión utilizada de este. Ahora además de haber las diferentes versiones de Bluetooth existe el Bluetooth **LE** (*Low Power*). Esta tecnología permite transmitir datos de forma inalámbrica con muy poca potencia. Para ello hay que tener en cuenta la atenuación del medio y las características de los diferentes dispositivos usados. En ambos dispositivos es muy importante la alimentación que dispone de ella, ya está define la energía que puede gastar este. En el caso del transmisor esta será usada para aumentar la potencia de este y en el receptor la ganancia de la antena. También está la sensibilidad que pueda tener el receptor que define como tiene que llegar la señal transmitida.

3.7.1 Aplicaciones

A continuación, hablaremos sobre los diferentes usos frecuentes para los que se usa esta tecnología y estándar. Esta tecnología se usa para el intercambio de información entre los diferentes dispositivos. Pero diferenciaremos las aplicaciones principalmente según la arquitectura que haya en los dispositivos. También se clasificará según la forma en la que se transmiten esos datos y que clase de datos son los transmitidos. Dentro de esta clasificación general podríamos clasificar estas según su uso dentro de los sistemas.

Primero según la forma de transmitir los datos tenemos a la transferencia de paquetes de datos y de flujo de voz. Estos actúan de manera totalmente diferente ya que uno puede transmitir los datos cuando sin reservar recursos. Es decir, a medida que va recibiendo los datos puede ir enviándolos según una secuencia. Sin embargo, el flujo de voz necesita de una reserva de capacidades de canal. Esto se debe a que se está transmitiendo un flujo de información constantemente. Y estas dos formas de comunicación se deben intercambiar en el mismo medio que es la banda **ISM** de radio. Algunas aplicaciones prácticas donde podemos ver la transmisión de los diferentes datos por bluetooth son los siguientes. En el caso de la transmisión de datos, podemos ver que se utiliza en casi todos los ámbitos que se utilizan en bluetooth. Los usos que podemos ver diariamente son para el uso de periféricos de ordenadores y accesorios de este. Esto puede verse en el caso del ratón o incluso del teclado que son periféricos fundamentales en la interfaz con el usuario. En el caso anterior se utiliza bluetooth por comodidad, ya que es muy molesto tratar con cables. Pero en el caso de los dispositivos que ayudan a los deportistas a analizar sus ejercicios es fundamental. Esto es debido a que imposible que los sensores que miden al sujeto estén conectado al sujeto. Ya que si fuera así en muchos casos le impediría al deportista el uso de los sensores. Esta razón por la que en este trabajo utilizamos Bluetooth para conectar el dispositivo a los diferentes sensores. Para finalizar también tenemos que apreciar el uso de este en dispositivos médicos. Estos son de gran ayuda para monitorizar al paciente de una forma mucha más cómoda. Ejemplos de esto pueden ser monitores de presión arterial o sistemas de rayos x. Por otro, lado por el flujo de voz tenemos el uso en este tanto en altavoces como en auriculares. Ya casi la totalidad de estos dispositivos van por Bluetooth para la comodidad del usuario.

Se han hablado de sus usos según el tipo de datos que utilizaban y que se transmitían entre los diferentes dispositivos. Y es que tradicionalmente, la arquitectura de red que Bluetooth ha seguido es la siguiente. Un transmisor da un servicio que puede ser escuchado por uno o por muchos receptores de esa información. Esta arquitectura es la que siguen la mayoría de los usos, que se ha comentado anteriormente. Pero actualmente se ha añadido otra arquitectura de red diferente con la que los diferentes dispositivos se pueden comunicar. Esta consiste en que los diferentes dispositivos puedan enviar o recibir información entre ellos. Esto se hizo con la intención de adaptarse a la demanda de **IoT** y poder expandirse en esta rama. Esto es por lo que principalmente se utiliza para sistemas de monitorización, de control y automatización. Con el sistema de monitorización nos referimos a un conjunto de sensores que nos permiten ver diferentes variables que nos interesa. Estos sensores a la vez se podrían estar comunicando con una red de controladores. Este te permitiría ejercer diferentes acciones para poder mantener las variables en el estado que necesites. El sistema que se puede controlar sin ninguna ayuda a través de controladores y sensores sería un sistema de automatización. Un ejemplo de esto lo podemos encontrar en un sistema de luz inteligente que actuaría como sistema de control. Podríamos añadir sensores de luminiscencia en una habitación que funcionaría como un sistema de monitorización. Si hacemos que según el grado de luz actúe el sistema de luz inteligente ya tendríamos un sistema de automatización. Este es un ejemplo, sencillo pero la idea es tener diferentes sensores de diferentes tipos que interactúen entre sí. Esto es por lo que se ha diseñado esta clase de tecnología en Bluetooth.

Para finalizar, destacaremos un nuevo uso para esta tecnología que está disponible a partir de la versión 5.0. Y es usarlo para servicios de localización que tiene muchos tipos de aplicaciones que se pueden usar diariamente. Este uso consiste en encontrar dispositivos a través de las ondas de radio Bluetooth. Esto es una cosa que se ha inventado desde principios del siglo XX con otros tipos de onda de radio. Y es el mismo mecanismo que utiliza Bluetooth hoy en día para hacer este uso de su tecnología. Este en la actualidad tiene diferentes casos prácticos que serán descritos a continuación. Se puede utilizar para dar información sobre monumentos en tiempo real. De esta forma conseguimos mejorar la calidad de las visitas turísticas en punto importante. Incluso esto se podría poner en museos para poder obtener información acerca del objeto cultural, artístico, científico o histórico. Este ayudara a facilitar la comprensión de la cultura y ampliar tus conocimientos acerca del tema en cuestión. Para uso diario esto puede usarse para encontrar donde se encuentra un objeto que hallas perdido. Esta aplicación es muy útil y añadiéndole un módulo Bluetooth a los objetos que puedas perder como las llaves. Otro uso diario que pueden tener este uso es el de sistema de navegación en espacio cerrados. De esta forma podremos hacer que los pasajeros de aviones llegan a sus puertas de embarque fácilmente. O enseñar a la gente donde se encuentran las diferentes obras dentro del museo. Después tenemos otra aplicación que se utiliza en empresas que tienen un gran número de recursos. Esta aplicación es la localización de sus recursos materiales y humanos en tiempo real. Esto te permitirá conocer su posición en todo momento para poder facilitar su búsqueda tanto de herramientas o materiales como de personas. Ejemplos prácticos lo tenemos en las herramientas de un almacén, a los que hay que añadirle modulo Bluetooth. O también en hospitales para saber en qué planta están los pacientes. También puede detectar si ha habido un traslado de habitación o saber dónde está en caso de haber salido.

3.7.2 Perfiles Bluetooth

En el apartado 3.7.1, Se muestran algunas de las muchas aplicaciones que tiene Bluetooth. Evidentemente todas estas aplicaciones no se consiguen con un único protocolo. Sino que habrá que usar el protocolo adecuado para cada uso que se vaya a hacer de la tecnología. Como decíamos en la introducción, Bluetooth es un estándar que permite soportar múltiples protocolos. Para ello, para cada forma de utilizar esta tecnología se le identifica con un perfil. Este tiene que definir una serie de requisitos que tiene que cumplir el dispositivo para ser utilizado de esa forma. Entre ellos se define una serie de procedimientos que se tienen que adoptar para cumplir ese perfil, en específico. Y para aplicar estos procedimientos en muchos casos se tendrán que aplicar unos protocolos específicos. Por lo tanto, para poder comunicarse los diferentes dispositivos Bluetooth, estos deben tener el mismo perfil. Esto se debe a que cada uno de los protocolos que se adoptan tienen que hablar con su otro extremo. Además, deben de reconocer que procedimientos se debe seguir. A parte, un dispositivo puede adoptar múltiples perfiles o un único perfil dependiendo del uso del dispositivo (62). Por ejemplo, un ordenador o un smartphone probablemente dispongan de diferentes perfiles. En la mayoría de los casos estos se eligen automáticamente y en la mayoría de las veces el usuario no tiene que elegir. Después tenemos de periféricos que normalmente solo incluyen un perfil porque solo tienen una aplicación. Es el caso de los auriculares o de un ratón de un ordenador. A continuación, procederemos a describir los perfiles más usados e importantes (62) (63). También hablaremos en los usos en nuestro proyecto.

- **GAP** (*Generic Access Profile*): Es la base de todos los perfiles y sobre el que se construyen los demás.
- **A2DP** (*Advanced Audio Distribution Profile*): este es el perfil que se utiliza para el flujo de audio multimedia. Este es el que se utiliza para auriculares o altavoces.
- **AVRCP** (*Audio/Video Remote Control Profile*): este perfil es el que se utiliza para controlar el flujo de video y/o audio. Puede hacer que a través de un único dispositivo controlar a diferentes dispositivos multimedia.
- **FTP** (*File Transfer Protocol*): Este es un perfil creado para que el protocolo FTP se pueda utilizar con Bluetooth. Este es un clásico protocolo de transferencia de datos. Este es el perfil que normalmente se utiliza si hay que transmitir un fichero a otro dispositivo.
- **DIP** (*Device ID Profile*): Este perfil permite identificar a los dispositivos que se encuentra en su área. Este es el perfil usado a la hora de buscar dispositivos para posteriormente conectarte al que le interese. Este es uno de los perfiles que usamos en nuestros dispositivos para encontrar a los diferentes sensores.
- **HFP** (*Hands-free Profile*): Este es de los perfiles más utilizados hoy en día. Sirve para la transmisión de voz de un dispositivo a otro. Este es el que se utiliza para transmitir llamadas a un altavoz o a los

auriculares con micrófono.

- **HIC** (*Human Interface Device Profile*): Este es el que se utiliza en periféricos que se usan como entradas para el dispositivo. Estos dispositivos son los que crean la interfaz entre usuario y dispositivo. Este está especialmente diseñado para ello ya que tienen poca latencia. Por lo tanto, están diseñados para consumir poca energía.
- **PAN** (*Personal Area Networking Profile*): Es que utiliza para acceder a la WAN, LAN o red correspondiente. Y es la que te permite compartir red entre dispositivos. Este protocolo es la actualidad el que ha mejorado al anterior perfil LAN.
- **SPP** (*Serial Port Profile*): Este es el perfil que permite simular existe una comunicación con un puerto serie. Y es el que utiliza la comunicación a través del protocolo RFCOMM. Este es el protocolo que nosotros utilizamos para que nuestros sensores se comuniquen con el dispositivo principal.
- **SAP** (*Subscriber identify module Access Profile*): Es el perfil que hacen que los diferentes dispositivos se puedan conectar como esclavo a una tarjeta **SIM** (*Subscriber identify module*).

3.7.3 Funcionamiento

Bluetooth ha tenido varias versiones de este protocolo que ha ido variando y han ido añadiendo funcionalidades con el paso del tiempo. Aunque, como ya se ha visto en el apartado 3.7.2, cada perfil tiene sus propios protocolos con los que trabaja. Si que todos los perfiles suelen incorporar una misma base, en la capa física y la de enlace. Esta ha ido cambiando a lo largo de las diferentes versiones que ha habido. Esto es debido a que los protocolos de las primeras capas necesitaban ser modificados para las diferentes tecnologías. Bluetooth actualmente se encuentra en su versión 5.3 (64). En esta nueva versión, se han incluido cambios para la eficiencia energética que tan útil es por lo menos para Bluetooth. Con esto a parte de ahorro energético, podemos conseguir que disminuya el tamaño de las baterías. Con esto podemos conseguir que los dispositivos tengan menor peso y sean de menor tamaño. Además, mejora la fiabilidad en la transmisión por Bluetooth. También incluyen características que permiten mejorar la experiencia del usuario.

Desde la versión 4.0 (63), Los sistemas Bluetooth pueden implementarse de dos formas diferentes. Se implementa una u otra dependiendo de las necesidades que tenga que cumplir el sistema. Estas formas son las transmisiones **BR** (*Basic Rate*), es la forma clásica, y la forma **LE**, que es de bajo consumo. A la primera forma se le puede añadir una opción que suele estar que es el **EDR** (*Enhanced data rate*). Esta esta implementada desde la versión 2.0 y permite mejorar la tasa de transmisión un poco más de 2 Mb/s (65). Normalmente, al conjunto de ambas tecnologías se le llama **BR/EDR**, y es con la que nos referiremos aquí. Y después, vino la forma **LE** que permitía gastar poca energía por lo que permitía alimentar a pequeños dispositivos. Esto se hizo con la intención de tomar el mercado emergente del **IoT**. De hecho, en la versión 4.1 se incluyeron todas las características relacionadas con ese mercado (65). Por lo tanto, para disminuir el consumo energético se necesita adaptar los protocolos para que funcionen eficientemente.

En Bluetooth se ha hecho una abstracción de los protocolos que permite adaptar a las diferentes aplicaciones. Se han agrupado en dos grupos que se le denomina como host y controlador. El controlador lo forma las capas inferiores de la tecnología, fundamentalmente la capa física y la capa lógica. Y este puede ser uno u otro dependiendo de la forma en la que se implemente el sistema. Existen tres opciones, la primera es que el controlador implemente los protocolos inferiores de la forma **BR/EDR**. La segunda es que implemente los protocolos de la forma **LE** y la tercera es que implemente los de ambas formas. Estos últimos controladores permitían usar el medio físico de una u otra forma según le convenga a las aplicaciones. Después tenemos la parte del host, formada por el protocolo **L2CAP** (*Logical Link Control and Adaptation Protocol*) y por algunos protocolos de servicio básicos. Como hemos visto puedes existir diferentes tipos de controladores, pero solo existe un host. Para que el host de arriba se comunique con cualquier controlador se necesita una interfaz entre los dos. Este papel lo cumple **HCI** (*Host Controller Interface*) que permite la comunicación entre las dos partes. Aunque, como veremos, la separación entre ambas partes no es tan clara.

3.7.3.1 Controladores

Empecemos hablando sobre los controladores, en esta hablaremos sobre cómo hemos indicado de la capa física y lógica. Estas capas a su vez tienen diferentes protocolos que permiten implementar diferentes entidades. Estas

son abstracciones que permiten controlar el medio físico en el caso de la capa física y transportes en la lógica.

3.7.3.1.1 Capa Física

Bluetooth trabaja en la banda **ISM**, que se puede usar sin licencia. Por lo tanto, esta será el medio físico por el que se comunican los dispositivos. La capa permite colocar de ese flujo que viene a través de ese medio en el formato adecuado. Este posteriormente será entregado a las capas superiores para poder ser procesado. Esta banda de frecuencia se divide a su vez en canales con un ancho de banda determinado. Estos son compartidos por los diferentes dispositivos que están conectados entre sí, esto se llama una *piconet*. Para que dos dispositivos formen una *piconet* necesitan haberse sincronizado tanto en el reloj como la frecuencia. Al dispositivo que da su señal de reloj es al que vamos a llamar central. Y al dispositivo que se sincroniza según esa señal de reloj es el que llamamos periférico. Dependiendo de la implementación del controlador, el tiempo de uso y canales se organizarán de forma diferente.

3.7.3.1.1.1 Canales físicos

Para empezar, diremos que los canales físicos están definidos por una serie de parámetros. El primero de ellos es por un conjunto de frecuencias de la capa física que se puede utilizar en él. Y luego todas ellas comparten una serie de parámetros temporales que dependerá del tipo del canal. Y estos canales están limitado especialmente debido a su rango de acción. Luego la forma en la que se implementa depende de la forma de comunicación que se use en el controlador. A continuación, hablaremos de cada una de ellas

En el caso de **BR/EDR**, existen 79 posibles frecuencias física que son usados para las diferentes comunicaciones. Los diferentes canales físicos se dividen en unidades de tiempos que en este caso llamaremos slot. Al canal que le toca a cada conexión es determinado según el patrón de frecuencia de salto. Este es un parámetro que es calculado a partir de la dirección Bluetooth y el reloj de la central. Y lo que calculamos a partir de ahí es un número pseudoaleatorio entre esas 79 frecuencia que es la que usaremos. Existen diferentes tipos de canales dependiendo del uso que se le vaya a dar.

Primero, tenemos los canales que permiten que permiten la conexión. Dentro de estos tipos de canales existen dos tipos el tipo básico y el adaptativo. En el básico cada slot está identificado a través de una secuencia de salto. Por lo tanto, el mensaje de respuesta necesita ser recalculado a partir del reloj de la central, dependiendo del slot. En el caso del canal adaptativo la frecuencia que utilizan tanto central como periférico es la misma. También en el adaptativo puede indicar dentro del patrón de frecuencia cuales van a estar inutilizadas. A parte de estos dos canales que se utilizan para la transmisión de la información. También, están los canales que se usan para señalización, mantenimiento y gestión de los anteriores canales. Estos canales son el canal de escáner *inquiry*, el canal escáner page y el de escáner de sincronización. Todas ellas tienen una topología totalmente distinta de la que se han visto en los canales de conexión.

En la anterior se formaba una arquitectura bidireccional punto a punto. Que en el caso de los canales de escáneres también se forma, pero durante un pequeño instante. Ahora en algunos canales entra en juego otro tipo de comunicación para el envío a varios dispositivos a la vez. Esta arquitectura es unidireccional punto a multipunto. La respuesta que se espera en este tipo de arquitectura es punto a punto, si es que se espera. El primero de ello tiene la finalidad de descubrir los dispositivos detectables. Para ello manda un paquete de forma broadcast en varias frecuencias y espera respuesta. Los dispositivos lo escuchan y aquellos que quieran ser detectable ofrecen su información al transmisor. Luego, está el canal escáner page este funciona de una forma parecida al anterior, en este caso se usa para establecer conexión. Por lo tanto, los dispositivos que solicitan una conexión mandan una petición al dispositivo que se quiera conectar. Para ello, envía la petición por varias frecuencias dentro del canal para poder encontrar al dispositivo en cuestión. El dispositivo aceptará o rechazará la conexión en el mensaje de respuesta. Para finalizar tendríamos el canal escáner de sincronización. En este canal se envía información acerca de los relojes de los dispositivos y los canales de frecuencia. A parte si un dispositivo pierde la sincronización necesita este canal para volver poder a conectarse. El dispositivo envía los paquetes de forma broadcast unidireccionalmente. Y los que le interesa conectarse se quedan escuchando para captar la información para posteriormente poder conectarse.

En el caso **LE**, tenemos dos esquemas de acceso múltiple para acceso al canal diferentes. Una es **FDMA** (*Frequency Division Multiple Access*), que hace una división por frecuencias y **TDMA** (*Time Division Multiple Access*) que hace una división por tiempo. En el esquema del **FDMA**, se utilizan 40 canales físicos distintos. De los cuales, 3 de ellos son canales de anuncio primario y los restantes son de propósito general. A estos canales formados para el acceso múltiple lo forma la propia capa física. Los diferentes canales físicos, según el esquema

TDMA, se dividen en porciones de tiempo, llamadas eventos. Hay diferentes tipos de eventos para las diferentes comunicaciones que se pueden producir entre los dispositivos. Los tipos de evento que hay son de anuncio, anuncio extendido, anuncios periódicos, conexión e isocrónicos. En los diferentes tipos de eventos que existen puede usar uno o más canales de la capa física. Todo esto dependiendo del tipo de evento que se esté aplicando. En el caso de los canales de conexión se usa simplemente un único canal de la capa física entre los dispositivos. Sin embargo, en los canales de anuncio se puede utilizar diferentes canales de la capa física durante este evento. Esto es por si los dispositivos a los que tienes que avisar están en otros canales.

Existen cuatro tipos diferentes de canales, de **LE piconet**, de anuncios, periódicos y **LE isocrónico**. El primero es usado para la comunicación entre los dispositivos que están conectados entre sí en un evento de conexión. En el evidentemente se producen eventos de conexión, que tiene su respectivo canal de salto de la capa física. El central es el que controla la comunicación dentro de una comunicación. Por lo tanto, los eventos empiezan y cuando estos terminan. Este canal físico, al igual que pasa con el canal de piconet adaptativo esto puede indicar cuantos canales usa. Puede haber hasta 37 canales físico de *piconet LE*. El canal físico de anuncio se diferencia dependiendo del propósito. Existen 3 canales físico de anuncio primario y después se encuentra los canales físicos de anuncio secundario. Los primarios son aquellos que se utilizan para controlar la entrada de los dispositivos a los canales físicos. Estos dividen en eventos los cuales tienen 3 canales de la capa física fijos, que pueden ser reducidos. Los canales físicos de anuncio secundario tienen estos disponen de 37 canales de capa física fijos. Estos tienen como propósito contener toda la información que no tiene los anuncios primarios. Después están los canales físicos periódicos. Tienen el propósito de mandar información de configuración a equipos a los que no está conectados. Este se divide en eventos de anuncios periódicos los cuales se corresponden con un canal de capa física. Existen 37 canales de capa física fijos que pueden ser reducidos. Por último, nos encontramos con los canales físicos de **LE isócrono**. Este puede ser controlado y utilizado por una central o por un *broadcaster* sin conexión. En él se envía datos isócronos en eventos isócronos que ocurren en periodos regulares. Cada uno de estos eventos pueden estar a su vez dividido en más de un subevento. Cada uno de estos utiliza un canal de la capa física. En el caso de haber una central, cada uno de ellos existe una petición de la central y una respuesta del periférico. En el caso del *broadcaster*, solo envían la información en una dirección de la comunicación. En este caso, pueden recibirla todos aquellos que estén sincronizados con el flujo de la comunicación. Existen 37 canales de capa física isócronos que pueden ser reducidos.

3.7.3.1.1.2 Enlaces físicos

Los enlaces físicos son entidades virtuales que representa la comunicación entre dos dispositivos Bluetooth. Un canal físico soporta uno o más enlaces físicos. Están tienen unas características físicas sobre la comunicación, que se representa en forma de parámetros. Estos pueden ser modificable o no dependiendo de la tecnología de la que dispongamos. Sí se puede modificar, se utiliza el protocolo de gestión de enlace y **LL** (*Link Layer*) en las formas **BR/EDR** y **LE**, respectivamente. Estos no tienen ninguna relación con respecto a información de los paquetes que se transmite por el medio físico. Pero sí que tiene una relación con el transporte lógico, que es una de las capas superiores. Y es que cada transporte lógico para por un único enlace físico. Hay diferentes enlaces físicos dependiendo si estamos en la forma **BR/EDR** o en **LE**.

En **BR/EDR**, nos encontramos con que existe los enlaces físicos activos y los enlaces broadcast de periféricos sin conexión. Los enlaces activos son los que nos encontramos entre una central y un periférico con conexión. Uno de los parámetros que tiene es estar en modo *Hold* y *Sniff*. Esto modifica los periodos de actividad y el tráfico que lleva a través de él. Este es un ejemplo, de las características físicas que puede soportar que cómo vemos son de bajo nivel. En el caso de resto de canales el enlace físico dura poco tiempo. En el caso de **LE**, existe un enlace físico por cada canal físico que tiene con una relación directa. De forma que cada enlace físico permite adecuarse a las características del tipo de canal. Por lo tanto, en este caso la identificación del canal si que estaría identificando al enlace también.

3.7.3.1.2 Capa lógica

La capa lógica permite definir una serie de comunicaciones que será establecida en los diferentes dispositivos. Esto se hace con el propósito de dar soporte a la comunicación de los diferentes tipos de aplicaciones. Por lo tanto, ofrece un grado de abstracción de las capas inferiores, siendo la conexión con esta. Dentro de esta capa existe dos subcapas, una es el transporte lógico y otra es el enlace lógico.

3.7.3.1.2.1 Transporte lógico

Esta capa se encarga de dar soporte a los diferentes enlaces lógicos pudieron llevar a uno o a varios de ellos. En

el caso de que sean varios estos multiplican estos para poder ser enviados a través del mismo transporte lógico. Este transporte lógico tiene una serie de características asociadas a la fiabilidad de la comunicación. Todo esto dependerá de las características que tengan esta por eso existen diferentes transportes lógicos. Ya que estos están definidos en base a las necesidades que tenga la aplicación correspondiente. Estas son control de flujo, mecanismos de asentimiento y repetición, enumeración a través de secuencias y comportamientos programados.

Existen diferentes enlaces lógicos dependiendo de la forma que tenga el controlador. En la forma **BR/EDR** estarían los transportes dedicados a conexiones y otros para comunicarse por difusión. Para las conexiones existen los transportes **ACL**, **SCO** y **eSCO**. De los cuales **ACL** se utiliza para conexiones asíncronas y tanto **SCO** como **eSCO** se utilizan para conexiones síncronas. Después para comunicaciones por difusión, tenemos los transportes **APB** (*Active Peripheral broadcast*) y **CPB** (*Connectionless Peripheral broadcast*). Los cuales se diferencian en que el primero se envía a todos los equipos que estén sincronizados con un enlace físico con el transmisor. Y el segundo se envía a todos los equipos que permita la red en ese momento.

En cuanto a la forma LE, tendríamos como dijimos al menos un transporte lógico asociado a cada canal físico. Por lo tanto, para las conexiones tenemos el transporte lógico **LE ACL**, como podemos estas son asíncronas. Para cada canal de anuncio tiene un transporte lógico, esto incluye el de anuncio periódico. Estos son llamados **ADVB** (*Advertisement Basic*) y **PADV** (*Periodical Advertisement Basic*) que corresponde a sus respectivos canales. Luego para los canales isocrónicos existen dos tipos de transportes lógicos. Estos son los que están dedicados a flujos de dispositivos conectados y después los dedicados a flujos de difusión.

3.7.3.1.2.2 Enlaces lógicos

Cada uno de estos enlaces lógicos tiene un transporte lógico determinado. Hay diferentes tipos de enlaces lógicos para identificar los diferentes flujos de información de las aplicaciones. Esto esta formado por un conjunto de protocolo, que se utilizan dependiendo de la información enviada. Tampoco todos los enlaces lógicos puedes estar en todos los tipos de transportes lógicos. Por lo tanto, dependiendo del tipo de la información habrá una clase de transportes que podrás coger. Estos protocolos también varían dependiendo del plano en el que nos encontremos. Es decir, si nos encontramos en el plano de usuario o de control.

3.7.3.2 HCI

Hasta el momento, el apartado se ha centrado en la capa de protocolo se divide en dos abstracciones. Estas son el host y el controlador. Y que la razón por la que se hacía era con la intención es que el host funcionará independientemente del controlador. Para ello se necesita una interfaz que permita comunicar cualquier controlador con el host y permita adaptarlo.

Esta interfaz está dividida en dos partes cada una de ellas pertenece a una parte del sistema. Tenemos en primer lugar al **HCI driver** que forma parte del Host y por otro al **HCI firmware** del controlador. Ambas partes se pueden comunicar a través de diferentes protocolos de bus físico como **USB**, **UART**, **SDIO** (*Secure Digital I/O*), etc... Esto es así ya que ambas partes están en el mismo equipo, aunque sean componentes diferentes. Funciona de tal forma de que el controlador no tiene que saber el contenido del mensaje que envía el host. Ni el host tiene que saber cómo funciona el controlador siempre que conozca su estado. Para ello, el **HCI firmware** enviará eventos con información del controlador a las capas superiores del host. Esto es debido a que el host tiene el control de las conexiones que se mantiene con otros equipos. Para ello **HCI driver** le envía unos comandos al controlador para que sean ejecutados ahí. Esto se hace mediante diferentes comandos de un protocolo según la forma que tenga de implementarse el Bluetooth. En la forma **BR/EDR** se aplica como dijimos anteriormente el protocolo gestión de enlace en control mientras que LE aplica LL. Por lo tanto, si la forma es **BR/EDR** estos comandos estarán formados por comandos **LMP** (*Link Management Protocol*). En el caso de LE, estos estarán formados por los paquetes de control de este protocolo. Además de comandos, los eventos se comunican a través de estos mismos protocolos anteriormente mencionados. Estos informan acerca de la situación de los controladores y de las conexiones que tiene con otros.

3.7.3.3 Host

El host es el encargado de recibir la información del usuario, procesarla y enviársela al controlador. A parte este controlara a sus correspondientes controladores y las conexiones a la que esté conectado a este. Este está formado

por las capas superiores de lo que sería un equipo Bluetooth. Como ya se menciono anteriormente, estas capas pueden variar dependiendo de los perfiles que se usen. Ya que estos hacen que se añadan o eliminen unos ciertos protocolos. Pero sí que existe un numero de protocolos que es esencial a la hora del uso de los demás. Estos son los que veremos en las siguientes secciones.

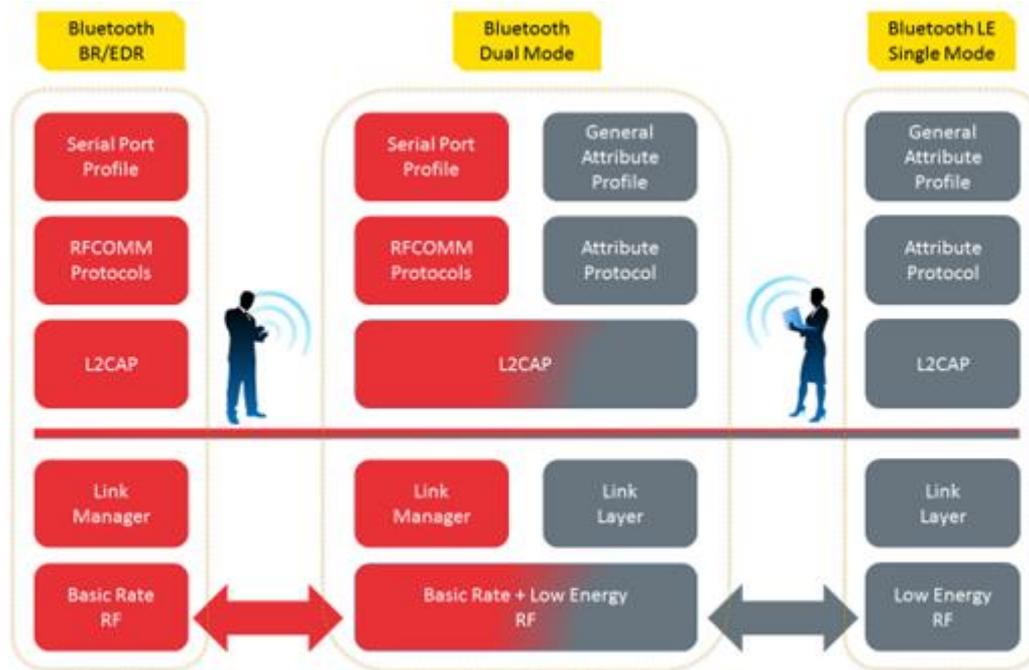


Ilustración 3.7-1. Torres de protocolos de Bluetooth. A la izquierda nos encontraríamos la **BR/EDR** (rojo) y a la derecha la torre de protocolo de Bluetooth **LE** (azul) (66).

3.7.3.3.1 L2CAP

L2CAP es el primer protocolo que normalmente nos encontramos en la torre de protocolo tras **HCI**. Este protocolo se encarga de crear canales lógicos en los que se envían los mensajes de capas superiores. Este canal da servicio de multiplexado y de fragmentación y ensamblado. Esto permite admitir un mayor tamaño de paquete a las capas superiores, entre otras cosas. Este protocolo necesita del soporte del controlador para saber cómo hacer su función. Esto es debido a que como controlador ofrece capacidades diferentes. Por lo tanto, necesita saber qué controlador va a realizar la conexión para así saber qué capacidades ofrecer. Por lo tanto, aunque se ha trabajado mucho en que con la interfaz ofrecida hacer independiente host y controlador. Aquí vemos un claro ejemplo en que el host necesita saber el controlador que utiliza y no es del todo independiente.

Además de las capacidades de adaptabilidad que se han observado que ofrecen. Además, también para cada canal ofrece unas ciertas capacidades de control de la comunicación. En particular, tenemos el control de flujo por cada canal y el control de errores. El control de flujo se aplica a cada canal, es decir, cada canal tiene su ventana, aunque se manden por el mismo controlador. Por lo tanto, el control de errores debe estar asociado al canal también para cambiar la ventana en función de este. Este se realiza haciendo retransmisiones cuando se detectan errores en los mensajes. Sin embargo, esto no se aplica cuando estamos hablando de flujo de datos. Esto es debido a que el transmisor debe estar constantemente enviando datos no le da tiempo a realizar retransmisiones. Debido a que si lo hiciera no podría enviar los datos correspondientes al tiempo de la retransmisión. Por supuesto, debido a que divide la información en canales, puede ofrecer calidad de servicio. Es tan sencillo, como dar a cada canal una prioridad y variar los parámetros de cada una de las capacidades que ofrece. De esta forma se mantendrá la calidad de servicio que se halla especificado.

3.7.3.3.2 SDP (Service Discovery Protocol)

Este protocolo es un protocolo cliente servidor cuyo servicio es informar de las capacidades y servicios del servidor. Este está en la capa superior a **L2CAP**, que es el protocolo explicado anteriormente. Este permite al

cliente saber qué servicios son ofrecidos por los dispositivos cercanos a él. De tal forma que el cliente pueda buscar los servicios que le interese según una serie de parámetros. Esta búsqueda no se puede realizar sobre cualquier parámetro, sino que solo los que tenga un **UUID** (*Universally Unique Identifier*). Y el significado de alguno de estos parámetros varían dependiendo del tipo de servicio al que nos referimos. Por lo tanto, a la hora de querer entender estos hay que saber qué tipo de servicio presta el dispositivo. Además, se pueden buscar todos los servicios cercanos sin información de los atributos previa. Por eso, toda la información no se le pasa inmediatamente al cliente para evitar colapso de información en el cliente.

El motivo de este protocolo es recolectar la información necesaria del dispositivo que nos interese conectar. Por ejemplo, en el descubrimiento del servicio no se le pasan todos los parámetros sino solo los que sirven para reconocerlo. Estos pueden ser parámetros como el nombre asociado al servicio o el tipo de servicio, entre otros. Así eliminamos la preocupación del cliente de que se sepa una serie de parámetros que son necesarios para la conexión. Esta información es almacenada en base de datos dentro de los diferentes servidores. Cuando el cliente se asocia al servicio del servidor es cuando se envía el resto de información del servidor. En Bluetooth, este protocolo es especialmente útil debido a la existencia de los perfiles. Esto es debido a que un mismo host puede adoptar diferentes perfiles cada uno con sus protocolos. Por lo tanto, aparte podemos saber si nuestro dispositivo puede acceder a todos los servicios a partir de nuestros perfiles. A parte, de esto si implementa varios perfiles el servidor **SDP** puede manejar las peticiones de los diferentes proveedores de servicios. Lo mismo ocurre con los dispositivos clientes tienen diferentes aplicaciones que son clientes. El cliente **SDP** puede realizar peticiones a los diferentes servidores asociados a través del cliente **SDP**.

3.7.3.3.3 GAP

Es el perfil que describe los mensajes y procedimientos que deben ser ejecutados por los dispositivos para comunicarse. Este debe de ser implementado para que no haya ningún problema en la comunicación. Como sería el caso de comunicación multiperfil o no poder establecer enlaces o canales entre ellos. Los procedimientos de este perfil pueden ser de dos tipos, obligatorios u opcionales. Aunque eso si si un dispositivo cumple un procedimiento opcional es obligatorio que el otro también lo cumpla. Principalmente, este perfil describe procedimientos de descubrimiento, establecimiento de enlace y seguridad. A parte, también se encarga de poner roles a los diferentes dispositivos. De esta forma cada uno sabe que tiene que realizar para comunicarse con el otro, ya que no es simétrico. Este tiene diferentes procedimientos según el tipo del controlador que tenga cada dispositivo. Pero básicamente tienen los mismos tipos de procedimientos, aunque cada uno lo implemente de distinta forma. Además, en el caso **LE** s que añaden un tipo de procedimiento para las conexiones isocrónicas y de aviso periódico. Esto es debido a que estas conexiones no existen para el caso de los controladores **BR/EDR**.

3.7.3.3.3.1 Modos y Procedimientos en BR/EDR

Existen procedimientos para el descubrimiento, establecimiento de enlace y la vinculación de los dispositivos. Además, Habla sobre la creación de canales a través de **L2CAP** y finalmente sobre su conexión o a través de broadcast.

3.7.3.3.3.1.1 Descubrimiento

Los dispositivos se pueden poner en dos modos, que son descubrimiento general o limitado. Esto hace alusión a si los dispositivos desean ser encontrados por otros o no. También existe la posibilidad de que los dispositivos no sean descubiertos. Una vez aclarado esto, el procedimiento es el siguiente a la hora de un cliente descubrir servicios. Y es que el cliente envía un mensaje tipo *inquiry* según el tipo del procedimiento así será el mensaje. Estos tipos puede ser de tipo general o limitado. En el caso, de que sea general todos los equipos que se puedan descubrir responderán al mensaje. En el caso de que sea limitado, solo aquellos cuyo escáner este configurado con el código de acceso a *inquiry* limitado. En cualquier caso, le responderá a quien corresponda con la dirección Bluetooth, reloj y la clase del dispositivo. Una vez descubierta la dirección ya se puede preguntar por el nombre puesto al dispositivo. Para ello existen dos posibles procedimientos, la petición de nombre o el descubrimiento de nombres. En uno se solicita el nombre de aquel dispositivo al que se pueda conectar que tenga una dirección determinada. Para ello se necesita hacer el mecanismo de page antes de conectarse.

3.7.3.3.1.2 Vinculación

En el mecanismo de vinculación pretende crear una conexión entre dos dispositivos mediante una clave que se intercambian ambos. Esta clave está asociada al enlace que comparten ambos dispositivos. Uno de los dos dispositivos crea la clave y lo envía hacia el otro dispositivo. Y se utiliza para autenticar los diferentes dispositivos en futuras conexiones. Existe dos formas de vinculación, la forma general y la dedicada. Lo único que se diferencia la una de la otra es cuando se realiza este intercambio de clave. Mientras en la general se realiza en el proceso de establecimiento de un enlace o un canal. En de la segunda se realiza antes o después de la configuración de estos de tal forma que es independiente del mecanismo page.

3.7.3.3.1.3 Establecimiento de enlace

Dependiendo del modo de seguridad este se realizará de un modo u otro. Pero lo que se produce siempre es un page donde se determinará la frecuencia de salto. En este se enviará el código de acceso del dispositivo que se ha hallado tras el descubrimiento de este. Luego ambos se pasarán las configuraciones del enlace para ponerse de acuerdo. Posteriormente pasarán una petición de establecimiento del enlace que deberá ser respondido. Y posteriormente hay una autenticación a través de la clave de enlace de la vinculación. Y después la negociación de la encriptación de los mensajes. Para finalizar se completará la configuración del enlace.

3.7.3.3.1.4 Establecimiento del canal y de la conexión

Existen diferencia según el modo de seguridad escogido, pero consiste en una petición y una respuesta. En esta se pasarán los parámetros adecuados del canal **L2CAP** del que se va a hacer uso. En algunos modos de seguridad además piden autenticación y negociación de encriptación. La conexión se hace de la misma manera que el establecimiento del canal. Pero en este caso lo hará la aplicación de usuario que se corresponda y esta enviará los datos que deban mandarse.

3.7.3.3.1.5 Broadcast

En este caso no existe establecimiento de enlace ya que no existe una conexión como cual entre ellos. Esta comunicación puede de necesitar de una sincronización de ambos relojes o no. Todo esto dependerá de las configuraciones que se encuentre el transmisor y el receptor. Por lo tanto, el transmisor enviará cada cierto tiempo un tren de sincronización durante un determinado tiempo. Luego el receptor en algún momento se pondrá a escanear y recibirá el tren de sincronización. Tras lo cual, se sincronizará a él y podrá recibir la información del perfil. Esto se hará a través de un mensaje **CPB** y ya podrá recibir mensaje de este tipo.

3.7.3.3.2 Modos y procedimientos en LE

Ya se ha hablado sobre los modos de procedimientos con los controladores **BR/EDR**. Ahora vamos a ver los modos en los que se puede poner un dispositivo que tienen un controlador **LE**. Aunque como dijimos los procedimientos tienen el mismo objetivo cambia la forma de hacerlo. Esto es debido a que se debe de adaptar a el modo de baja energía que se quiere conseguir con estos controladores. Los procedimientos nuevos son para las conexiones isocrónicas y los avisos periódicos.

3.7.3.3.2.1 Descubrimiento

Los modos son los mismo que en **BR/EDR**, es decir, pueden ser descubiertos o no. Y cuando son descubribles se pueden clasificar en el modo general o en el limitado. Los que están en el modo general descubrimiento solo pueden ser descubiertos a través del procedimiento de descubrimiento general. Sin embargo, aquellos que estén en el modo limitado descubrimiento serán descubiertos en ambos procedimientos. Tanto en el procedimiento de descubrimiento general como en el limitado. Por lo tanto, se está filtrando los dispositivos en modo limitado en

el procedimiento de descubrimiento limitado. Esto es debido a que los del modo general no se escogen en este procedimiento. Esta va a ser la diferencia que va a existir entre los dos procedimientos. El procedimiento consiste en que aquellos periféricos que estén en algún modo de descubrimiento mandan un aviso. Este aviso es periódico y será captado por una central que está escaneando durante un tiempo en el proceso de descubrimiento. Este aviso contendrá la dirección del dispositivo e información interna como su modo de descubrimiento. Una vez que acaba el tiempo el dispositivo recoge todos los avisos y los guarda según el procedimiento. Si es general los guarda todos si es limitado solo aquellos avisos que provenga de dispositivo modo limitado. También procedimiento de descubrimiento de nombre, sino se ha hecho en los anteriores protocolos. Y para ello necesitamos hacer uso de dos procedimientos. El primer paso es realizar un establecimiento de conexión a través de su procedimiento correspondiente. Y después usar el procedimiento de lectura definida en el perfil **GATT** (*Generic Attribute Profile*) para leer el nombre con su **UUID** correspondiente.

3.7.3.3.2.2 *Conexión*

Un mismo dispositivo puede tener uno o más roles de conexiones. Estos son el papel que desempeñan estos dentro de una conexión y un dispositivo puede tener varios. Ya que este tendrá un papel u otro dependiendo de las funciones que quiere que este implemente dentro de la conexión. Los diferentes modos de conexión que existen son los que explicaremos a continuación. Para empezar, tenemos el modo no conectable, el cual no admite conexiones de ningún dispositivo. Después, tenemos el modo de conexión directa o indirecta, dependiendo de a quien le permita conexión. Estaríamos en el modo directo, sí las conexiones que admite son solo de dispositivos peer ya conocidos. Sin embargo, sería modo indirecto, sí las conexiones que admite son de cualquier dispositivo. En ambos casos, el dispositivo tiene que ejecutar el procedimiento de establecimiento de conexión general o automático. Estos dos procedimientos dependen de los parámetros que adquiera el host. En el primer caso, solo se pueden hacer este procedimiento a un conjunto de dispositivos peer ya conocidos. Esto es debido a que antes los controladores han negociados para tener los mismos parámetros. En el segundo caso, se añade a la lista de filtro de aceptación aquellos dispositivos a los que puedes conectarte. Y el controlador cambia su configuración para poder adaptarse a algunos de estos dispositivos. De esta forma hacemos posible la conexión con todos ellos.

Aparte de esos procedimientos de conexión, también tenemos el selectivo y el directo. En el cuál uno, puede establecer conexión con cualquier dispositivo de la lista de filtro de aceptación. Y el otro, permite el establecimiento de conexión con un único dispositivo peer. Además, existen procedimientos para actualizar los parámetros de la configuración durante la conexión. Y también existen procedimientos para terminar la conexión. También se describe para las conexiones síncronas, procedimientos de temporización antes y durante la conexión. También existen procedimientos para las conexiones síncronas, tanto como para empezar de cada lado como para terminarla.

3.7.3.3.2.3 *Vinculación*

En esta fase solo se utiliza para intercambiar entre los dos dispositivos información de identidad y de seguridad. De esta forma una vez que los dispositivos han sido vinculados se puede decir que hay una conexión segura. Los dispositivos pueden tener dos modos, estos son vinculable o no lo son. De tal forma que se propone un mecanismo de vinculación. En el que un dispositivo no vinculable no puede acceder a información de uno vinculable. Esto es debido a que se exige la vinculación para que así no pueda entrar cualquier dispositivo.

3.7.3.3.2.4 *Aviso periódico*

Este te permite el envío de información periódica unilateralmente de sincronización u otra. Para ello existen dos modos el de aviso periódico y el de aviso de sincronización periódico. Y existen dos procedimientos, que están relacionados con la sincronización. Esta los procedimientos de establecimientos de la sincronización y de transferencia de esta sincronización. En el primero, el dispositivo que adquiere la información debe de sincronizarse. Este proceso se da antes de que ocurra la comunicación e una conexión síncrona. Y el segundo, es el envío de nueva sincronización una vez ocurrida la conexión entre ambos dispositivos.

3.7.3.3.2.5 *Broadcast isocrónico*

Los dispositivos envían información de isocrónica a través de un flujo unidireccional **BIS** (*Broadcast isochronous stream*). Para ello los dispositivos toman dos roles fundamentales, difusor y observador. Uno envía la información y los observadores que estén asociados a ese **BIS**, obtienen esa información. Los diferentes observadores de un **BIS** forman un grupo que comparten la misma información, **BIG** (*Broadcast Isochronous group*). El difusor puede estar en dos modos uno que es de difusión y otro que es de sincronización. El modo de difusión permite enviar los paquetes de difusión isocrónica de los subeventos del flujo al grupo. El modo de sincronización necesita que el modo de difusión este activado. Este sirve para transmitir la información de sincronización dentro de un **BIG**. En este modo, los observadores tienen que transmitir además su información de sincronización. Para finalizar, hay que decir que para que esto funcione se proponen dos procedimientos. El primero, permite al difusor usar y transmitir un mapa de canales al resto de observadores. Y esto lo recibirán y usarán para saber a quién pueden escuchar. El segundo procedimiento es para indicar a los observadores que se va a dejar de enviar información a ese **BIG**.

3.7.3.3.4 Atributos

3.7.3.3.4.1 *ATT (Attribute protocol)*

En los dispositivos **LE**, tendríamos un protocolo que permite transmitir un conjunto atributos de las comunicaciones. Este protocolo se llama **ATT** y seguiría una arquitectura servidor-cliente. Donde es el servidor el que aporta la información de los atributos. Estos pueden ser descubiertos y leídos por el cliente e incluso pueden ser modificados por este. Cada atributo tiene un **UUID** que identifica su tipo, un manejador de este y un conjunto de permisos. Estos permisos describen que capas o protocolos deben de hacer uso de estos atributos. No se puede acceder a estos permisos a través del protocolo **ATT**. El manejador es único en el servidor lo que permite al cliente referencia al atributo a través de él.

3.7.3.3.4.2 *GATT*

GATT es un perfil que declara las estructuras de datos de los perfiles para ser intercambiadas. Por lo tanto, aquellos perfiles que vayan sobre este perfil tendrán que utilizar esta estructura. La estructura que define para los perfiles es que estos contienen servicios que a su vez estos contienen características. Los servicios son un conjunto de datos y comportamientos asociados a estos que cumplen una función en particular. Esto es para que el dispositivo cumpla con los requisitos que impone ese perfil en particular. Los servicios pueden ser de dos tipos, primarios o secundarios. Los primarios proveen funcionalidades que se consideran fundamentales para el dispositivo. Estos no pueden ser modificados para que no haya incompatibilidad con clientes ya conectados. En su lugar se crean servicios secundarios que están referidos a uno primario o a otro secundario. Estos incluyen de esa forma funciones adicionales. Además, este servicio junto a todos los referidos a él forman un servicio independiente. Las características de los servicios es un valor e información acerca de ese valor. Esta información además de sus propiedades incluye como esta es accedida, mostrada y representada.

GATT especifica una comunicación cliente-servidor para obtener las características de los diferentes servicios. Para ello utiliza el protocolo **ATT** que permite enviar este tipo de datos. El servidor será aquel que tenga toda la información de los perfiles que se puedan implementar. Y el cliente será aquel que haga peticiones acerca de esos datos que el servidor debe aceptar. Se describe los procesos con el cual el cliente descubre, lee, escribe y obtiene indicaciones de esos atributos. Como este perfil va sobre **GAP**, se definirá un servicio **GAP** dentro del servidor **GATT**. Este implementará todas las funcionalidades impuestas en **GAP**.

3.8 Raspberry Pi

Este es un ordenador de pequeñas dimensiones como los microcontroladores vistos en el apartado 3.6. Este es de mayor tamaño que un microcontrolador debido a que incluye más funcionalidades. Ya que, a diferencia de un microcontrolador, este si es completamente como un ordenador real. No tiene un propósito único, sino que

es de propósito general como los PC. Este incluso incorpora un sistema operativo adaptado a la Raspberry pi (67). En 2012, se creó el primer producto final de este dispositivo, aunque ya había prototipo de este desde 2006 (68). Este se convirtió en uno de los primeros microrordenadores. Sus aplicaciones son muchas y muy diversas ya que como se ha dicho antes tiene propósito general. Pero principalmente este es usado en el campo de la robótica y de la ingeniería en general. Se pueden hacer servidores que consuman pocos recursos de una gran variedad. Por sus dimensiones, también es muy usado en el internet de las cosas (69). También están muy de moda para utilizarlos en inteligencias artificiales (70).

La Raspberry pi sigue la arquitectura **ARM** (*Advanced RISC Machine*) que la solemos encontrar en los dispositivos móviles. Aunque ahora se ha aplicado a los primeros ordenadores también. Como dice en el nombre de **ARM**, este sigue también la ideología de arquitectura **RISC**. Esta arquitectura solo permite instrucciones que ocupen un ciclo de memoria. Esto, aunque limite el tipo de instrucciones que pueda haber, sí que acelera mucho la ejecución del procesador. Este tipo de arquitectura solo permite hacer operaciones entre registro (71). Y como es **ARM** solo hay algunas funciones que permiten sacar un dato de memoria y ponerlo en registros. Este tipo de arquitectura permite que con poca electrónica realizar un buen desempeño del procesador. Además, está diseñada para bajo consumo, lo que evita que tenga que preocuparte de refrigeración (72). Esto es debido, a que las instrucciones son simples. Por lo tanto, el procesador no tiene que hacer funciones muchos microcódigos. Además, la falta de dificultad en la electrónica abarata los costes de producción. Esta tecnología al principio era de 32 bits, pero en la actualidad se está consiguiendo arquitectura de 64 bits. Esto hace que cada vez más este a menos distancia de los procesadores con arquitectura **CISC**. En nuestro caso las Raspberry 3 tiene un **SoC** (*System on Chip*) Quad-core multimedia BroadCom BCM2837 (67). Tiene cuatro núcleos, usa una arquitectura ARMv8 de 64 bits y va a 1,2 GHz.

Los sistemas operativos que usan las Raspberry pi están basados en UNIX/Linux. Esto es debido a que el código de Linux es totalmente abierto. Es por ello, que se ha podido cambiar el código de este para adaptarlo a esta tecnología. Es decir, se ha realizado la portabilidad de estos sistemas operativos a la Raspberry pi. Ya se han portado diferentes distribuciones de esta. Como Debian, Fedora Remix y Arch Linux. Sin embargo, la fundación de la Raspberry pi recomienda usar la distribución que parte de Debian. A que es un sistema operativo debido a que la carga del sistema operativo es ligera. El que vamos a utilizar es este que es el recomendado. Este incorpora una interfaz gráfica que el usuario podrá usar y el interpretador de Python. A parte incorpora programas para programar en ese lenguaje. Es por lo que todo el código de la Raspberry pi de este trabajo está programado en este mismo lenguaje. Además, incluye un navegador para conectarse a la red. Esta funcionalidad de poder conectarse a la red permite que sea de forma inalámbrica desde la versión 3. Este es instalado sobre una memoria externa. En el caso más frecuente de la Raspberry pi, este es instalado sobre una memoria microSD (*Secure Digital*) (67). Ya que la Raspberry pi toma a esta memoria como fuente de almacenamiento principal.

Para el almacenamiento y los periféricos incorporados en la placa dependería de la Raspberry pi utilizada. En nuestro caso, una Raspberry pi 3 modelo B, el almacenamiento constaría de 1 GB de memoria **RAM**. La capacidad de esta memoria puede variar dependiendo de la capacidad que desees. También incluye un BCM43438, que permite el uso de redes locales wi-fi 802.11 b/g/n. Además, tiene las capacidades necesarias para usar Bluetooth de baja energía. Incorpora Ethernet base 100 para las conexiones cableadas. Y para comunicarse con la parte más electrónica tiene 40 pin de entrada y salida. Además, incorpora cuatro puertos USB v2, para periféricos, y un puerto microSD, para almacenamiento. Para las salidas de audio tendríamos cuatro puertos Pole estéreo. En el caso de la salida de video, tendríamos un puerto de video compuesto y un puerto **HDMI** (*High-Definition Multimedia Interface*). Para la entrada de video tendríamos un puerto cámara **CSI** (*Camera Serial Interface*) que permite conectar una cámara Raspberry pi. Y como entrada y salida de video, tendríamos un puerto **DSI** (*Display Serial Interface*), para conectar una pantalla táctil Raspberry pi. Para finalizar como fuente de alimentación tendríamos un puerto micro **USB**. El último modelo de la Raspberry pi es la Raspberry pi 4 modelo B. Esta mejora el procesador incluyendo un **SoC** Broadcom BCM2711 con Quad-core Cortex-A72 de 64 bits. Este procesador va a 1.5GHz. Además, incorpora wi-fi a 5GHz y la versión 802.11 ac y también, Bluetooth versión 5.0. Tiene dos puertos **USB** v3.0 y dos puertos **USB** v2.0. Y en las salidas de video tiene dos puertos micro **HDMI**, dos puertos **MIPI** (*Mobile Industry Processor Interface*) **DSI** y dos puertos **MIPI CSI**.



Ilustración 3.8-1. Raspberry pi 4 modelo B (73).

3.9 BBDD (Base de datos)

Una base de datos es un conjunto organizado de información. Estos conjuntos de información tienen un **DBMS** (*DataBase Management System*) para gestionarla. La información más el **DBMS** forman el sistema de la base de datos. Estas fueron creadas al principio de la década de los 60. Y había solo una clase de **BBDD** para datos eran estructurado. Estas eran las bases de datos de navegación, que a su vez se podían dividir en más clases. Estas eran las jerárquicas y las de redes (74). Se fueron creando diferentes tipos de base de datos según las necesidades de las personas y negocios.

3.9.1 Tipos de Base de datos

Los conjuntos pueden tener la información de forma estructurada o no. Dependiendo de cómo se organice la información se pueden diferenciar diferentes tipos de **BBDD**. También esta evolución en los tipos se ha propiciado gracias a la creación de nuevas tecnologías. Vamos a comenzar a hablar sobre los tipos de **BBDD** dependiendo de la organización de información almacenada.

- Relacional: la información está organizada en un conjunto de tablas con sus filas y columnas correspondientes. Las filas serían los valores de cada campo para cada una de las entradas. Y cada columna es el valor de un campo para todas las entradas disponibles. Estas empezaron a usarse en la década de los 80. Además, es la forma más eficiente y flexible de acceder a unos datos estructurados (74).
- Orientado a objeto: Trata a la información como un objeto de un lenguaje orientado a objeto.
- Distribuidos: tiene la información organizada en dos o más ficheros repartidos en diferentes lugares. Esta puede estar en la misma subred o dentro de una red más profunda. Y todos los archivos siguen formando parte de una misma base de datos lógica. Tiene que haber alguna forma de conseguir que el **DBMS** pueda gestionar todos los ficheros a la vez (75). Una forma de realizarlo es centralizando el **DBMS** de forma que guarda una copia de todos los datos. Y va actualizando los ficheros de cada ordenador donde está distribuido periódicamente.
- *Warehouse*: Este guarda todos los datos de forma centralizada. Y está diseñado para hacer peticiones y análisis rápidamente.
- *NoSQL*: son las bases de datos que almacenan datos no estructurados o semi estructurado. Estos se pusieron de moda cuando las aplicaciones web alcanzaron más popularidad.

- Grafos: Son aquellas cuyos datos son almacenados en términos de entidades y relaciones entre ellas (74). Está diseñada para trabajar con datos altamente interconectados. Y para tratar conjuntos de datos grandes y complejos (76).
- Nube: es una base de datos que se encuentra en una nube privada, pública o híbrida. Los datos que almacenan pueden ser tanto estructurado como no estructurado. Y tiene un tipo especial de modelo de base de dato, que consiste en base de dato como servicio. Este consiste en presentar las funciones administrativas y de mantenimiento como un servicio.
- Documento: organizan la información orientada a documento en formato **JSON** (*JavaScript Object Notation*). Este es un formato estándar para representar datos. Esta se utiliza para la comunicación de estos mismos

A parte de estos tipos existen muchos más dependiendo de la organización de los datos. Pero la mayoría son propietario o son desarrollado para una función en específico. Luego tendríamos modelos nuevos de estas bases de datos creados a partir de fusionar los anteriores. Y ahora también están muy en auge la base de datos autónoma. En la que estas están basadas en el modelo de la nube, en la que se instala una inteligencia artificial. Esta se carga de todas las tareas de gestión como, actualizaciones, puesta a punto, seguridad y recuperación.

3.9.2 Lenguaje de base de datos

Ahora, se comentarán los lenguajes de programación que sirven para editar la **BBDD**. Estos son los usados en las **DBMS** para realizar operaciones sobre la base de datos. Estas operaciones pueden cumplir tareas de control, gestión y mantenimiento de esta. Pueden permitir desde la creación y diseño de una **BBDD**, hasta realizar consultas y monitorización de esta. Además, una vez creadas y rellenas estas bases de datos debe permitirse usar sus datos. Estos lenguajes pueden ser de diferentes tipos según las funciones que permitan estos (77).

- Están los lenguajes de definición de datos, es el que permite el diseño de una base de datos. Permitiendo crear o eliminar bases de satos o sus estructuras e incluso cambiar o renombrar estas.
- Están los lenguajes de manipulación de datos que permiten la gestión de datos en esta. Debe de tener funciones para crear, borrar, actualizar y consultar datos.
- Estaría los lenguajes de control de datos, que controla los permisos de acceso de los usuarios. Este debe tener funciones para dar o quitar permisos de una base de datos.
- El lenguaje de control de transacciones, que gestiona las transacciones de una base de datos. Las transacciones es una tarea que está relacionado con un conjunto de tareas.

Algunos ejemplos de estos lenguajes de base de datos serían **SQL** (*Structured Query Language*) y **XQuery**. **SQL** es un lenguaje de definición y manipulación de datos. Este trabaja con bases de datos relacionales y es de los lenguajes más utilizados. **XQuery** es un lenguaje de manipulación de datos mediante el formato **XML** (*Extended Mark up language*). Esto permite la comunicación a través de internet.

3.9.3 DBMS

DBMS sirve como interfaz entre la base de datos y el usuario, ya sean persona o programas. Este está formado por un programa que organice la información de forma óptima y organizada. Este tiene que ser capaz de realizar tareas como actualizar o recuperar información de la **BBDD**. También debe tener funciones de supervisión y control de la **BBDD**. Existen varios **DBMS** famosos entre los que vamos a destacar unos cuantos. Primero, nos encontraríamos a **MySQL** y **PostgreSQL** como ejemplo de **DMBS** de código abierto. **MySQL** es un sistema de gestión de base de datos basado en **SQL**. Esta diseñado y optimizado para aplicaciones web y puede ejecutarse en cualquier plataforma. Ya que está diseñado para procesar millones de peticiones. **PostgreSQL** también trata con las bases de datos relacionales y extiende el lenguaje **SQL**. Este es conocido por su integridad en los datos y conjunto de características robustas (78). Además, este es conocido por tener una gran comunidad capaz de buscar soluciones ingeniosas. También tenemos **DBMS** que no tengan código abierto, por ejemplo, **Microsoft Access**. Este tiene una interfaz gráfica para usuarios no experimentado y permite crear gráficas fácilmente. Aunque también permite programar a través de un lenguaje de base de datos. En este caso, el programa que se utiliza es el **VBA** (*Visual Basic Application*) (79).

3.9.3.1 BBDD autónoma

En el caso de las bases de datos autónoma no se necesita la ayuda al administrador con todas estas tareas. Ya que todas estas funciones que eran tan tediosa ahora pueden ser realizadas por una inteligencia artificial. Las empresas tienen más carga de datos debido a que hay más sensores dando una gran cantidad de información. Esto se debe a que el internet de las cosas ha estado revolucionando el mundo empresarial. Pero esta tecnología necesita una buena **BBDD** que dé cabida a todos estos datos y que se adapte a la empresa. Y esta base de datos necesita ser mantenida, escalable y ágiles (74). Esto permitirá adaptarse al crecimiento del negocio, y así no se quedará desactualizada. Todo esto es una gran carga de trabajo que se puede hacer cargo una inteligencia artificial en vez del administrador. Además, las inteligencias artificiales te ayudan a seleccionar la información que pueda ser de utilidad. Ayudándote así a la toma de decisiones, de forma que, este proceso llega a ser más ágil. Además, permite aumentar la seguridad del sistema. Estos además les dan libertad a los usuarios, ya que la base de datos que se creen depende de ellos. Por lo tanto, obliga que el usuario sea más activo en la creación de la base de datos sin perder en seguridad.

3.9.4 Servidor de base de datos

Esta clase de servidor ofrece un servicio de base de datos. En el que los clientes pueden realizar ciertas operaciones con la información almacenada en él. Estos clientes pueden ser tanto locales como remoto. El servidor será un software que permite la organización de la información. Y además se encargará, como todos los servidores, de atender las peticiones de sus clientes. Estos servidores deberían ser equipos especializados que pueden almacenar una gran cantidad de información. Además, lo ideal sería que fuera capaz de acceder y operar con esta de la forma más rápida y eficiente. Al almacenar una gran cantidad de información es importante tenerla protegida. Para ello los clientes que acceden a esta base de datos normalmente se tendrán que autenticar. Y cuando se autenticen, se le asignarán un permiso, que indicarán que operaciones pueden realizar. Por ejemplo, la mayoría de los clientes normalmente solo necesitan acceder a la información sin modificarla. Estos no se les da permiso de modificación de la **BBDD** porque pueden introducir datos erróneos. Los servidores actuales tienen simultaneidad en las consultas. De tal forma que clientes remoto puedan leer y modificar datos a la vez sin perder integridad de los datos.

Los servidores de base de datos que están más de moda son **MariaDB**, **PostgreSQL** y **MongoDB** (80). **MariaDB** es la nueva versión del servidor **MySQL**, es muy utilizado en base de datos en desarrollo web. Es un servidor que da acogida a base de datos relacionales. Y permite simultaneidad en las tareas tanto de escritura como de lectura. Este está hecho con software libre, al igual que lo es su **DBMS**. Es bastante rápido y soporta varios motores de almacenamiento. El servidor **PostgreSQL** contiene base de datos relacionales, de código libre y orientado a objeto. Esta caracterizada por su robustez, estabilidad y velocidad en la administración de datos. Por ello es por lo que es muy buena opción para grandes empresas y entornos gubernamentales. Por último, hablar de **MongoDB** que tiene el motor **NoSQL** orientado a documentos más utilizados. No guarda su contenido en tablas como las anteriores, sino que sigue la estructura de **BSON** (*Binary JSON*) dinámicas. Esto hace que su acceso sea sencillo, ágil y es multiplataforma. **BSON** está basado en **JSON**, este a su vez es un formato estándar para intercambio de datos (81). La diferencia entre uno y otro es que en **BSON** tiene estructuras binarias. Esto permite codificarlo más rápido el tipo y la longitud de la información. Otras funciones interesantes es que permite configuración de privilegios de usuario y **SSL/TLS** (*Secure Socket Layer / Transport layer security*). Es por esto que es muy segura y confiable ante fallos. Otras funciones que la hacen robusta son replicación de datos, balanceo inteligente de carga. Por último, tenemos algunas funciones de la eficiencia del almacenamiento. Que serían la agregación de datos y la indexación.

3.10 Inteligencia Artificial

Una inteligencia artificial consiste en una serie de algoritmos encargados de imitar el comportamiento humano. Sin embargo, esta tiene la potencia de procesamiento y la velocidad de ejecución de un ordenador. Esto es completamente diferente a lo que hace un ordenador convencional. Ya que este solo ejecuta lo que se le dice. Sin embargo, una inteligencia artificial es capaz de hacer más que eso. Es capaz de pensar por sí sola e incluso hacer procesos de la mente humana como aprender. Ya que la tecnología de la nube ha permitido que la alta capacidad de cálculo necesario sea asequible (82). Además, ahora disponemos de una cantidad de datos que una

inteligencia artificial necesita para funcionar. Ahora, disponemos de una gran cantidad de datos que se pueden tratar con facilidad. Esto es debido a que es asequible tener la capacidad de almacenar y procesar estos datos con facilidad. Estos datos pueden ser estructurado o no estructurado. Estos pueden ser almacenado gracias a herramientas como la que se ha desarrollado en el apartado 3.9. Esta cantidad de datos con la llegada del internet de las cosas ha llegado a tener incluso más información. Hay diferentes lenguajes de programación que permiten implementar esta clase de tecnología. Tenemos a Python, R, Matlab y muchos más.

3.10.1 Aplicaciones

El uso de esta tecnología se está expandiendo en todo el ámbito empresarial y científico. Esta nueva tecnología tiene mucha utilidad en multitud de campos variados. Nos lo podemos encontrar para usos del día a día. Como para el reconocimiento de voz, que tienen inteligencias artificiales como Siri y Alexa. Además, también nos lo podemos encontrar en los algoritmos de búsqueda de navegadores como Google. En el campo de la medicina es fundamental para el análisis de señales médicas. Además, incluso podría predecir ciertas patologías en los pacientes o dar diagnósticos. Este último es el caso de Gyant, que es un asistente médico (83). Incluso este puede ayudar con operaciones, en la fase de preparación de esta (84). Y este es muy importante sobre todo en el campo de imagen médicas (85). En el campo de la ingeniería también nos los podemos encontrar en el desarrollo de sistemas electrónicos (86). Como ya se ha visto en el apartado 3.9, esta también sirve para crear base de datos autónomos. Además, está impulsando en la tecnología del big data (87). Esta permite reconocer patrones dentro una inmensa cantidad de datos. En los entornos empresariales, se están usando como asistentes de chat (82). Además, también permite supervisar tanto web, aplicaciones, experiencia de usuario y registros. Permitiendo ahorrar mucho tiempo al equipo técnico de la empresa y ponerlo a trabajar en tareas más relevante. Por último, permite hacer análisis empresariales sin contar con un experto. Simplemente alguien sin conocimientos empresariales con esta clase de herramienta. Podría sacar un análisis de esta sabiendo cómo funciona el programa. También existen otras ramas como en educación, logística, transporte, climáticas ... Y la lista es casi interminable y es que esta tecnología se puede aplicar a una inmensa cantidad de ámbitos (83). Y el uso de esta es una tendencia al alza.

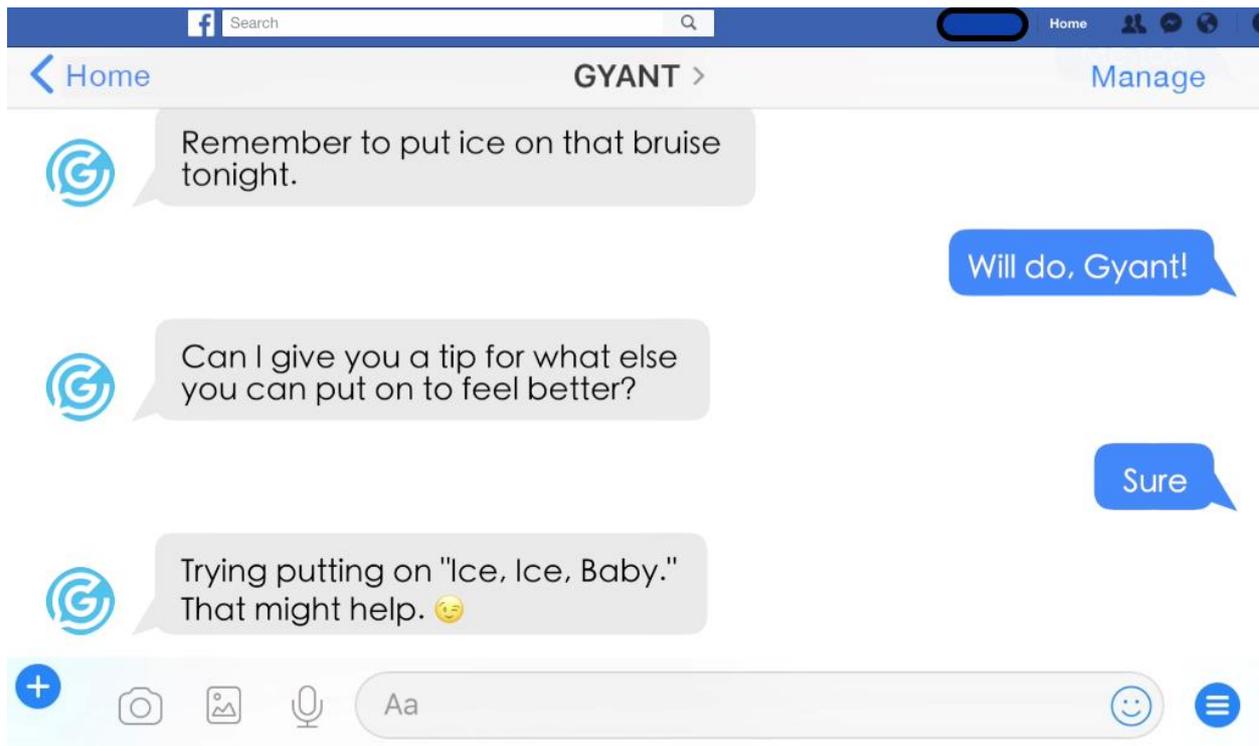


Ilustración 3.10-1. Chat de texto de una conversación con Gyant (88).

3.10.2 Tipos y algoritmos de machine learning

Existen muchos tipos de algoritmo de inteligencia artificial, ahora se va a explicar algunos de ellos. Pero antes se va a clasificar estos según el tipo de procedimiento que sean capaces de alcanzar. Esta clasificación está realizada por dos expertos en ciencias de la computación, Peter Norvig y Stuart Rusell (83) (89).

- Sistemas que piensan como humanos. Estos piensan como un humano. Automatizando algunos procedimientos típicos que nos interesan de nuestra mente. Estos en concreto son la toma de decisiones, el aprendizaje y la resolución de problemas.
- Sistemas que actúan como humanos. Estas máquinas realizan tareas tal y como lo haría un humano. Estas máquinas serían los robots y andróides.
- Sistemas que piensan racionalmente. Estos intentan emular el pensamiento lógico y racional de un humano. Intentan que los sistemas perciban algunos datos del mundo, razone acerca de ellos. Y a posteriori que actúe en consecuencia. En este grupo está los sistemas expertos, que intentan emular ser un humano experto en algún tema.
- Sistemas que actúan racionalmente. Estas máquinas intentan imitar el comportamiento racional de los humanos. Un ejemplo de ellos sería este serían los agentes inteligentes.

Los algoritmos que más nos interesa son los de *machine learning*. Estos entrarían dentro de los sistemas que piensan como humanos. Los algoritmos de *machine learning* son herramientas que proponen soluciones a problemas planteados. Para resolver estos problemas, estos algoritmos se caracterizan por aprender cómo se solucionan por sí solos. Y no existe un algoritmo que sea el que mejor encaje para todos los problemas que se le planteen. Si no que los algoritmos darán peor o mejor resultado dependiendo de las características del problema. Estas son como de que tipo de problema estamos hablando o con cuantas variables trata. Dentro del *machine learning* hay tres paradigmas a la hora de aprender a resolver estos problemas (90). De los problemas más importantes que se ayudan a resolver con esta herramienta sería los siguientes. Estarían los problemas de regresión, clasificación y *clustering*. Los primeros serían los que se encargan de crear una función continua a

partir de unos puntos dados. Los problemas de clasificación, identifica a que categoría pertenece la entrada. Por último, los problemas de *clustering* consisten en agrupar las entradas con las mismas características. Estos son los algoritmos supervisados, los no supervisados y los reforzados. Que veremos que esta forma de realizar el aprendizaje está presente en los algoritmos que vamos a ver.

- Algoritmos supervisados: Estos algoritmos se le pasa dos clases de datos. Aquellos que sirva para la maquina aprenda y aquellos que queremos que nos prediga la salida. Por lo tanto, necesita que los datos sean divididos entre los de entrenamiento y los que hay que probar. Este trabajo debe de ser realizado un asistente a la inteligencia artificial. A través de los datos de aprendizaje podrá aprender patrones de estos. Y estos serán utilizados en los datos que hay que probar para generar las salidas. Estos algoritmos están en el paradigma que tiene su mismo nombre. Algunos algoritmos importantes de esta clase serían los siguientes.
 - **Árbol de decisión.** Este algoritmo crea un diagrama de árbol formado por nodos y relaciones entre nodos. Los nodos serían eventos o elecciones. Y las relaciones serían decisiones o condiciones. Este algoritmo es muy utilizado para problemas de clasificación

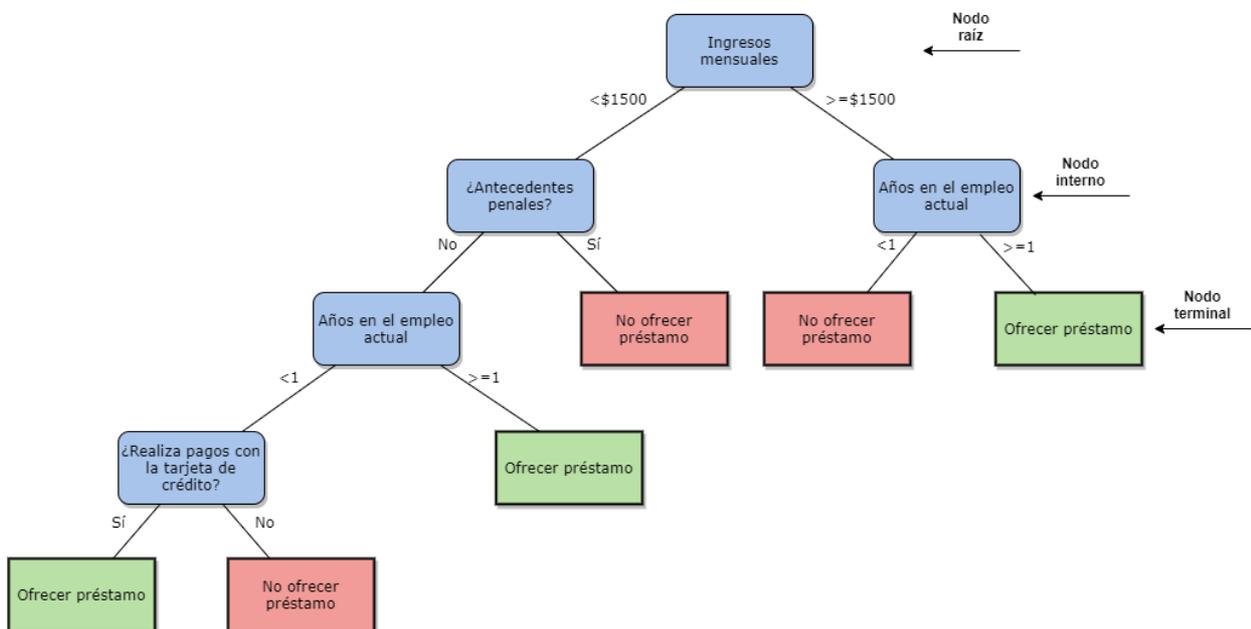


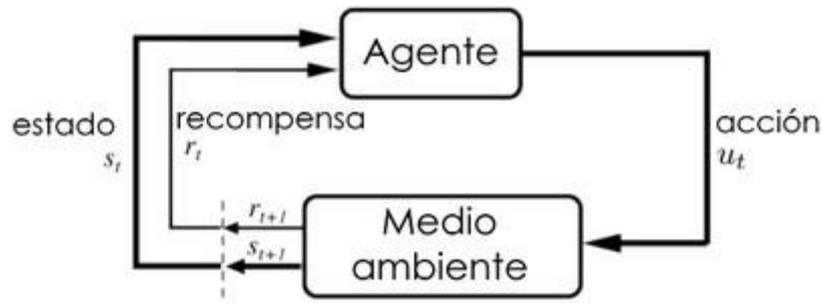
Ilustración 3.10-2. Ejemplo de un algoritmo en forma de árbol de decisión (91).

- **Navie Bayes.** E un algoritmo basado en el teorema de bayes, expresado en la ecuación (3.10-1). Este se basa en que las variables de entradas sean independientes entre sí. Se utiliza en los problemas de clasificación y *clustering*.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \quad (3.10-1)$$

- **SVM (Support Vector Machine).** Este algoritmo crea un margen entre puntos que pertenecen a diferentes grupos. Y se dibuja de tal forma que la distancia entre el margen y la clase sea el mayor posible. Esto hace que haya más diferenciación entre las diferentes clases. Esto sirve para problemas de clasificación y de regresión. Sobre todo, para modelo lineales. Aunque existe la posibilidad de aplicarlo en modelo no lineales, con el problema del Kernel. Esta técnica por ejemplo es usada en *CanoeSense*. Es utilizado como clasificación de cada una de las fases del movimiento.

- K vecino más cercano. Este algoritmo utiliza los datos de entrenamiento como método de predicción. Para ello únicamente miramos la entrada más parecida a la que vamos a predecir. Y le asignamos la salida de la entrada del entrenamiento que más se parezca. Como podéis observar es totalmente diferente a las anteriores, este método es directo.
- Algoritmos no supervisados: A diferencia del otro paradigma esto no necesita clasificar los datos de entrada. Estos por ellos mismos aprenden a reconocer patrones entre los datos aportados. Y los utilizan para los datos que se quieran predecir. Esta clase de algoritmos están diseñados para una gran cantidad de datos. Vamos a ver los algoritmos más usados dentro de esta clase.
 - Componentes principales. Este algoritmo crea variables independientes a partir de variables dependientes entre sí. Para ellos hace uso de transformaciones ortogonales. Esto permite reducir dimensiones de los datos haciendo su ejecución más rápida y fácil. Además, explica la estructura de varianzas y covarianzas en una combinación lineal de variables.
 - K-means *Clustering*. Este algoritmo funciona de forma que se busca los centros de cada grupo a clasificar. Una vez encontrado se asocian las entradas a su centro más cercano. Y a cada centro se le asocia un grupo. Este proceso se repite hasta que se alcance un estado de convergencia.
- Algoritmos semi-supervisados: Esta clase de algoritmos combinan las entradas con entrenamiento y pueden aprender sin ellas. Esta clase de algoritmos nacen de los problemas con pocos datos de entrenamiento. Además, tampoco tenemos una gran base de entradas. Por lo tanto, se hace una mezcla de las dos. Vamos a ver los algoritmos de este estilo más famosos.
 - SVM transductiva. Se basa en un SVM, y aún no se entiende muy bien el todo el funcionamiento. Pero se basa en crear un margen entre las variables etiquetadas y las que están sin etiquetar. De tal manera, que la distancia entre los dos grupos sea máxima.
 - Modelos generativos. Este algoritmo genera datos a partir de unos datos. Estos modelan tanto su característica como a la clase a la que pertenecen. Estos generan los datos a través de la distribución de probabilidad de los datos. De manera, que cualquier función que modele esa distribución sirve para generarlos.
 - Autoaprendizaje. Se utilizan los datos de aprendizaje para crear los patrones necesarios. Una vez hecho se introducen los nuevos datos como un algoritmo supervisado. Pero una vez predichos estos se añaden a la lista de datos de aprendizaje. De esta manera, se van actualizando los patrones a medida que vas añadiendo datos. Este algoritmo funciona para problemas de clasificación.
- Algoritmos reforzados: Estos algoritmos están basados en que existen dos entidades que serían el agente y el entorno. El entorno tiene un estado que es la entrada del agente y el agente actúa en consecuencia. Para aprender a partir de este mecanismo, se le ofrece una recompensa por su actuación al agente. El agente tendrá en consideración esta recompensa en la actuación en adelante. Este es el último de los paradigmas de la inteligencia artificial.



[Figure source: Sutton & Barto, 1998]

Ilustración 3.10-3. Algoritmo básico de inteligencia de refuerzo (92).

- Algoritmos de redes neuronales: Este no es un tipo de paradigma, sino que es un tipo de tecnología dentro de *machine learning*. Y es la utilización del funcionamiento del cerebro en una inteligencia artificial. El funcionamiento del cerebro es una red de neuronas, en las que cada neurona realiza un proceso. Después de realizar el proceso pasa la información a la siguiente neurona. Y como podemos observar, este puede adaptarse a los cambios en la entrada producidos. Este tipo de *machine learning* funciona igual a la que cada neurona se le asocia una función. Y se conectan entre sí. El funcionamiento de la red global se convierte en una especie de caja negra. A esta tecnología se le aplica los tres paradigmas. El aprendizaje profundo toma esta misma tecnología y la aplica en varias capas. De tal manera que hace que todos trabajen para solucionar un único problema. Con cada capa se va añadiendo un nivel más de abstracción. De tal forma que separamos un problema difícil en otros más sencillos y cada vez más abstractos (93).
 - Red neuronal supervisada. En esta se ve la diferencia entre la salida de entrenamiento y la obtenida por la red. A partir de esta diferencia se cambia los parámetros de la red neuronal.
 - Red neuronal no supervisada. Esta red simplemente busca similitudes entre los datos de entrada de esta. Y se adapta la red para agruparse en ese grupo de datos.
 - Red neuronal reforzada. Esta funciona mediante objetivos que cumpla la actuación del agente. Se le dará a este una recompensa u otra en función al número de objetivo o a su mejor intento.

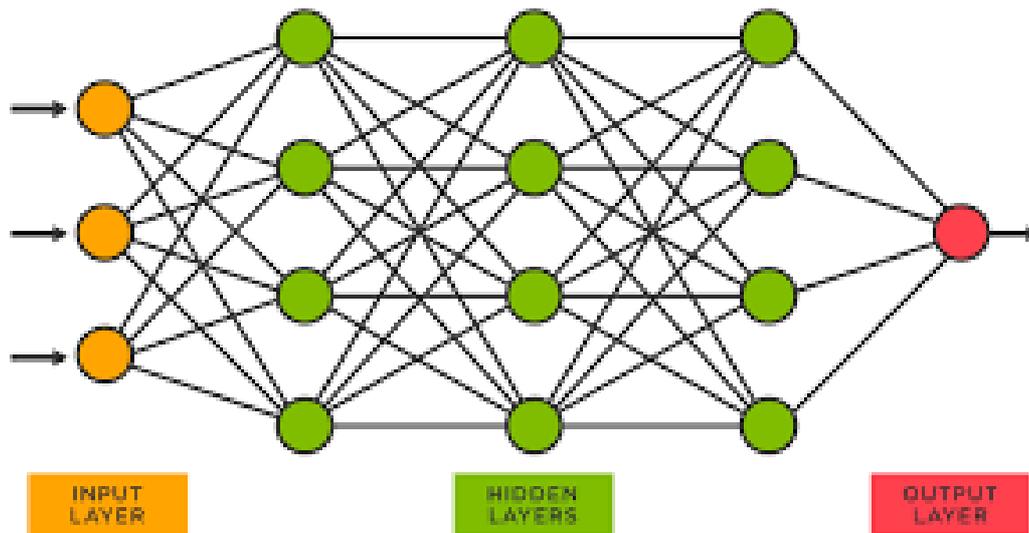


Ilustración 3.10-4. Muestra de una red neuronal y sus partes (94).

Se ha visto los algoritmos de *machine learning* más utilizado y su clasificación. Además de esta forma de crear algoritmos, también se puede crear algoritmos a través de otros. Esto se hace si quieres aumentar el alcance del *machine learning* o aumentar el número de entradas. Esto se hace añadiendo *machine learning* que se dedican a diferentes problemas. Este algoritmo se llama *machine learning multitask*. Por último, hablar que se puede crear un algoritmo con menos fallos mezclando a partir de otros. Esto se consigue mezclando algoritmos que den muchos errores. Esta forma de crear *machine learning* se llama *ensemble learning*.

3.10.3 Scikit learn

Este es el módulo de Python creado para usar algoritmos de *machine learning*. Este está construido sobre las librerías **NumPy**, **SciPy** y **matplotlib** y es de código abierto (95). Además, tiene una gran comunidad para solucionar los errores que pueda haber en el código. Este módulo tiene se puede usar para multitud de aplicaciones. Para ello este permite solucionar multitud de problemas que nos encontramos en este campo. Algunos se han mencionado mencionado ya en el apartado 3.10.2. Para los problemas de clasificación, se hace uso de algoritmos **SVM** y el vecino más cercano entre otros. Para los problemas de regresión, se hace uso de algoritmos de vecino más cercano y **SVM**. Los problemas de *clustering* son resueltos mediante algoritmos K-means entre otros. También tenemos forma de eliminar variables de entradas a considerar. De esta forma aumentaremos la eficacia de la solución. Para ellos usaremos algoritmos de **PCA** (*Principal Component Analysis*) y de selección de características entre otros. Tenemos forma de tratar los datos para estar preparado para una inteligencia artificial. Para ello podemos utilizar algoritmos como la extracción de características. Para finalizar, estaría el problema de validar, comparar y elegir los parámetros y los modelos elegidos. Para ello tenemos algoritmos como la validación cruzada.

Nos vamos a centrar en el caso del modelo lineal que se usa, por lo menos, en los casos más sencillos (96). Cualquier modelo lineal se puede describir mediante la ecuación (3.10-2). La inteligencia artificial nos puede dar los parámetros que nos hace falta para terminar describir el modelo. Para ello este módulo utiliza una inteligencia artificial supervisada. El modelo lineal más sencillo que podemos aplicar es la siguiendo la ecuación de mínimos cuadrados. Para ello debemos de minimizar la ecuación (3.10-3). Pero esta ecuación se basa en que todas las variables de entrada son independientes. Para solucionar podemos ajustar la ecuación, sacando la ecuación (3.10-4). que depende del parámetro α . Este modelo para la regresión se llama *Ridge*. Que se debe ser

ajustado para el buen funcionamiento del modelo. Este puede ser calculado mediante validación cruzada, que es uno de los métodos mencionados antes. Aquí esta librería utiliza la validación cruzada dejando uno fuera. Este método de calcular los parámetros consiste en coger todos los datos como entrenamiento excepto uno. Se calcula los parámetros entrenando a la inteligencia artificial y se aplica dicho modelo al que es de prueba. Y se calcula el error cuadrático medio. Se repite este proceso dejando fuera cada una de las entradas. Y se calcula la media del error cuadrático medio. Esta media nos da una medida si el modelo funciona. Se elige el parámetro α que menos EMS tenga. Existen parámetros que permiten saber lo bien que se ajusta a los puntos dados. Por ejemplo tenemos el caso del error cuadrático medio. Que es la media de la diferencia entre el valor introducido para aprender y el valor predicho para cada entrada. Y también está el valor r^2 . Este nos indica cuanto esta de relacionada la variable de entrada con la de salida en la regresión lineal (97). La expresión vista en la ecuación (3.10-5) es la que describe matemáticamente este parámetro.

$$\omega_0 + \omega_1 x_1 + \dots + \omega_m x_m \quad (3.10-2)$$

$$|X\omega - y|^2 \quad (3.10-3)$$

$$|X\omega - y|^2 + \alpha |\omega|^2 \quad (3.10-4)$$

$$\frac{cov(x, y)^2}{\sigma_x^2 \sigma_y^2} \quad (3.10-5)$$

3.11 Conclusión

Se han explicados todos los conocimientos que vamos a utilizar en el trabajo. Se ha hablado sobre los dispositivos que tratamos en el trabajo y de los protocolos que hacemos usos. Se ha mencionado el estado del arte de los trabajos que están también aplicados al deporte. A parte se han visto, la tecnología de almacenamiento de la BBDD y de la inteligencia artificial. Esta se aplicará para crear un modelo mucho más sencillo. En el siguiente apartado, se verá cómo aplicar estas tecnologías para conseguir nuestros objetivos.

4 APLICACIÓN

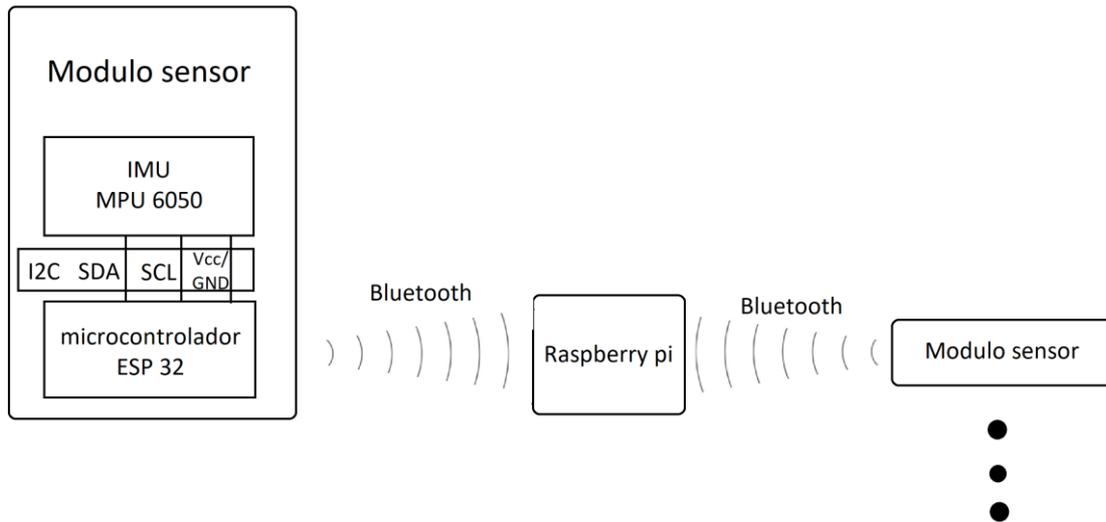
4.1 Introducción

La aplicación debe de ser capaz de crear un modelo sobre la ejecución de una salida. Para ello lo primero que se debe calcular los ángulos de las salidas de los deportistas. Para ellos se usará los **IMU** (*Inertial Measurement Unit*) explicados anteriormente. Se hará uso de los microcontroladores para controlarlos. Y finalmente los datos que se irán muestreando llegarán a la Raspberry pi a través de Bluetooth. Y estos serán almacenados en una BBDD (base de datos). Posteriormente, se utilizarán los conocimientos matemáticos visto en 3.4.3.2 para tratar los datos. Y posteriormente se creará un modelo con la ayuda de la librería *Sci-kit learn* vista en 3.10.3.

4.2 Arquitectura

Como se ha explicado anteriormente, la arquitectura básica estaría formado por un módulo central y sensores. El módulo central es una Raspberry pi, en este caso en concreto una Raspberry pi 3 modelo B. Los sensores se necesitan que estén conectados inalámbricamente. Esto es debido a que debe permitirle libertad de movimiento a los palistas. Los protocolos cable limitarían el movimiento que pueda realizar el deportista. Además, estos podrían tener problemas con los cables y el agua que podría ser peligroso para el deportista. Para realizar este tipo de comunicaciones se necesitan que ambas partes tengan el módulo de la tecnología correspondiente. Ya que los sensores por sí mismo no suelen contener este tipo de comunicaciones. Estos serán conectados a unos controladores, que disponga este módulo, a través de buses. En nuestro caso la **IMU** utiliza el bus **I2C** (*Inter-Integrated Circuit*) para conectarse al controlador. Ahora se debe de preguntar qué tipo de comunicación debe haber entre la Raspberry pi y las **IMUs**. Un parámetro para tener en cuenta es la distancia de la comunicación entre los diferentes módulos. Debido a que se desea que dicha distancia no afecte a los datos, esta distancia se decidió que sea corta. Con esa intención, se deja la Raspberry pi en el barco del piragüista. De tal forma que la distancia de los módulos será de unos metros como máximo. Además, la carga de datos es baja, simplemente se envían datos numéricos de ángulos cada 100ms. Por esas razones es por las cuales se ha elegido Bluetooth como protocolo de comunicación.

El controlador debe esperar a que la Raspberry pi inicie la conexión Bluetooth. Esto es debido a que es la Raspberry pi desde la que vamos a controlar el sistema. Por lo tanto, una vez iniciada cada una de las conexiones, este mandará los comandos a ejecutar. Estas instrucciones son poco frecuentes y sirven para iniciar algún proceso en los controladores. Este controlador no necesita tener grandes capacidades técnicas. Tan solo debe ser capaz ejecutar el protocolo Bluetooth y saber leer datos a través de I2C, para leer datos de la **IMU**. Este controlador interesa que pese poco y que sea de pequeñas dimensiones. Esto es debido a que cuanto menos pese, más cómodo le resultará al deportista llevarlo. Además, el movimiento del palista será más fidedigno a la realidad de su palada. Y cuantas menos dimensiones, mejor se podrá acoplar este a la parte del cuerpo correspondiente. Habiendo dicho todo esto, lo lógico sería escoger un microcontrolador con las funcionalidades necesarias. Además de las razones de escasez de funcionalidades necesitadas y del tamaño necesitado, está el coste. Y es que en este caso se necesita 5 microcontroladores para su respectiva **IMU**. Se ha escogido el microcontrolador ESP 32 como microcontrolador de la **IMU**. Ya que tiene tanto **I2C** como Bluetooth, es pequeño y es relativamente económico. En la siguiente imagen se puede ver como este es usado dentro del sistema. Y la función de intermediario que este tiene.



Esquema 4.2–1. Se muestra como están las conexiones dentro del módulo sensor y con el módulo central.

Por último, debido a que la Raspberry pi está en el barco del piragüista, se necesita alguna forma de interactuar con ella. Esto es debido a que considero que se complicaría la sincronización entre las salidas y el programa. Esto es debido a que el piragüista debe cumplir unos requisitos en las salidas. También, se puede retrasar por factores humanos y que el programa no considera. De forma que es mejor que el usuario de la aplicación indique cuando hay que medir. Esto lo puede hacer perfectamente un usuario desde una lancha o con algún vehículo desde la orilla. Se podría pensar tener una tarjeta **SIM** (*Suscriber identify module*) en la Raspberry pi y que esta se conectara a la red es la mejor solución. Pero no se puede garantizar que en cualquier parte del río esté conectado a la red. Por lo tanto, la solución que se ha dado es conectar la Raspberry pi a una red wi-fi de un dispositivo portátil. Se ha elegido la tecnología wi-fi debido a que tiene gran alcance y es de fácil uso. Además, está incluido en muchos dispositivos, como los que se ha usado la Raspberry pi y un teléfono. En este caso, limitamos la distancia en la que se encuentra ambos dispositivos, pero aseguramos conectividad. Por lo tanto, se ha utilizado a través de un móvil para tener mayor movilidad. Una vez que esté en nuestra misma red, abriremos una sesión **SSH** (*Secure Shell*) con la Raspberry pi. De esta forma se podría ejecutar los comandos que nos hagan falta de la Raspberry pi.



Esquema 4.2–2. Esquema de la arquitectura seguida a la hora de realizar las medidas al deportista.

4.2.1 IMU

La **IMU** que se ha utilizado es el **MPU (Motion Process Unit) 6050** debido a que es económico. Además, es una **IMU** muy usado hoy en día debido a lo económico que es y para la funcionalidad que tiene. A parte, su uso es sencillo para gente que se inicia en el mundo de sensores inerciales. Este incorpora tres sensores de los que podemos leer sus datos. Estos son el giroscopio, el acelerómetro y un termómetro. De los cuales, se va a utilizar los dos primeros a través del módulo **DMP (Digital Motion Process)**. Este te proporciona los datos del ángulo en forma de cuaternión. Esto es justo lo que necesitamos para calcular los ángulos de las articulaciones más tarde. Y este pone el valor de ese cuaternión en un registro. El valor de este va decayendo en el tiempo. Por lo que, si se mantiene el suficiente tiempo parado el valor del cuaternión se ira a 1. Es decir, el dispositivo detectará que no ha habido un giro desde el sistema inicial. Por lo tanto, se ve la importancia de que si alguien se detiene en medio de la salida se perderá los valores iniciales.

4.2.2 Microcontroladores

Como se ha dicho la función antes de este microcontrolador es de intermediario entre el **IMU** y la Raspberry pi. De forma que este controla cuando leer la información del **IMU** y cuando enviarla al módulo central. El lee directamente de un registro del **IMU** que tiene los datos del cuaternión cada aproximadamente 100 ms. Y lo manda a través de Bluetooth a la Raspberry pi. En la parte inicial, prepara los sensores y la conexión con la Raspberry pi. Cada uno de los microcontroladores actúa como servidor de Bluetooth. Se considera que esta debería ser la topología ya que quien tienen los datos necesarios son los microcontroladores. A parte es la Raspberry pi la que solicita el envío de dichos datos, por lo tanto, toma el rol de cliente. Estos microcontroladores han sido programados en C a través del **IDE (Integrated Development Enviroment)** de Arduino. Y esta alimentada con una fuente de alimentación de 3,7 V y 650mA.



Ilustración 4.2-1. Imagen del módulo sensor. Se puede observar cómo está conectado el microcontrolador ESP 32 (microchip negro) a la **IMU MPU 6050** (atornillador al recipiente) mediante I2C.

4.2.3 Raspberry pi

La Raspberry pi es la que inicia la comunicación Bluetooth con cada uno de los microcontroladores. Además, es la que da la orden de empezar a leer datos cuando empieza la salida. Para ello, la Raspberry pi inicia todas las conexiones con los microcontroladores paralelamente. Esto debe ser así ya que necesitamos que se muestren los

datos de todos los sensores simultáneamente. Esto es necesario debido a que hay que fusionar los datos para obtener los ángulos de las articulaciones. Una vez obtenidos todos los datos de los sensores, estos son almacenados en una base de datos. Además, se almacenan en la base de datos los datos posteriormente fusionados a partir de los medidos. Todos los archivos de la Raspberry pi están programados en Python 3 excepto los archivos de la **BBDD**. Estos están escritos en **SQL** (*Structural Query Language*). Como se ha comentado anteriormente la Raspberry pi está conectada al dispositivo móvil a través de wi-fi. De forma que se ha creado una red wi-fi en mi teléfono y luego, se ha conectado la Raspberry pi a esta red. Ya que no se disponía de ninguna salida en el lugar de las medidas, comprobaba a través del teléfono si estaba conectado. Este dispositivo va alimentado con una fuente de alimentación de 5V y 3A.

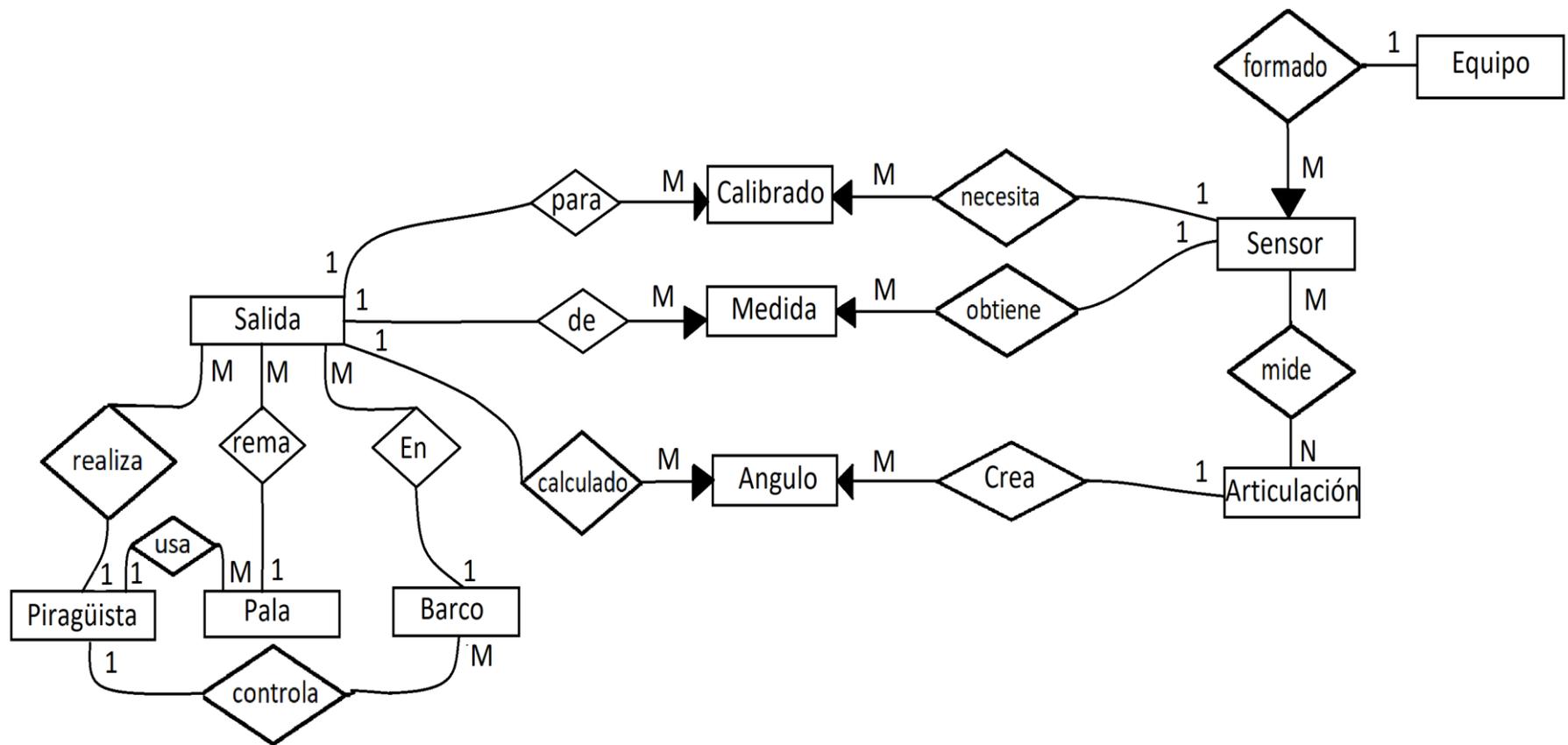
4.3 Base de datos

Como se ha dicho anteriormente se necesita una base de datos, para almacenar todas las mediciones. Además de datos adicionales con los que se puede trabajar posteriormente como los ángulos de las articulaciones. Para ello se crea un servidor de base de datos **MariaDB** en la Raspberry pi. Dentro de este servidor se ha creado la base de datos *tfg* y un usuario llamado *admin_tfg*. El cual puede acceder a esta con todos los permisos de un administrador. Para acceder a ella se ha utilizado su **DBMS** (*DataBase Management System*), que es **MySQL**.

4.3.1 Diseño

Para empezar una medición pertenece tanto a un **IMU** como a una salida realizada. Ya que una medida es realizada por un **IMU** en una prueba en concreta y en un tiempo en concreto. Por lo tanto, estos tres parámetros serán las que identificarán a una entrada y formarán la clave primaria. De los cuales, tanto la salida como la **IMU** son referencias a sus tablas correspondiente. La salida debe indicar datos fundamentales como quien ha realizado la salida y en qué condiciones se ha realizado. Estas condiciones pueden ser el barco y la pala con la que se realizó y el tiempo o distancia que recorrió. Ya que se pueden hacer medidas por distancia o por tiempo, una será fija y la otra dependerá del deportista. En el caso que sean por tiempo, se necesitaría un dispositivo capaz de medir distancias o se mediría con marcas. También se incluye el ritmo con la que se realizó la salida y la fecha para ser localizable más fácilmente. Como ya se ha indicado, se indican se indican tanto el piragüista como su barco y su pala. Aquí se ha considerado que el deportista es propietario de su barco y pala o al menos que lo usa regularmente. En los datos del piragüista estaría datos para identificarlo como sus nombres y apellidos. Además, se incluyen medidas del como peso y altura y finalmente su categoría y nivel. Estos datos se usarán posteriormente en caso de que se quieran comparar datos de diferentes piragüistas. Tanto el barco como la pala tienen datos de características de ellas como su marca.

Por otra parte, estaría la **IMU** correspondiente a una cierta medida. Debido a que no se si en el futuro se utilizaran otro equipo de sensores en esta base de datos. Se añadió una entidad llamada equipo a la cual estaría formada por un número de sensores. Esta contendrá datos acerca de lo que está midiendo y propiedades del sistema. El sensor indica el tipo de sensor y su modelo como datos informativos acerca de él. Aparte este indica que es lo que está midiendo. Por otra parte, los sensores necesitan ser calibrados al principio de su funcionamiento. Por lo tanto, guardamos es información en la tabla calibrado. Donde almacenamos los offsets del giroscopio y el acelerómetro correspondiente a una salida en concreto. Esta tabla referenciará tanta a la salida como al sensor del que sea el calibrado. A partir de estas medidas se pueden sacar los ángulos que realizan las articulaciones mediante la fusión de estos. Por lo tanto, una articulación será medida a partir de varios sensores y un sensor puede servir para medir varias articulaciones. Y finalmente estarían los ángulos formados por las articulaciones. Estos ángulos dependerán tanto del tiempo como de la salida en concreto en que se haya realizado la medición. Ya que la medición es la utilizada para formar ese ángulo. Y además esta dependerá de la articulación correspondiente. Este ángulo tiene dos formas tanto la forma de Euler extrínseca como con cuaterniones. Los cuaterniones representan el giro que realizan los sistemas de referencia. Por otra parte, los ángulos de Euler nos dan una concepción mejor sobre el ángulo que se gira en cada eje.



Esquema 4.3-1. Tablas de la base de datos y las relaciones que guardan unas con las otras.

4.4 Protocolo

Ya se ha visto en el apartado 4.3, como se ha realizado la conexión entre los diferentes dispositivos. En este apartado hablaremos de como la aplicación funciona en el caso de realizar las medidas. Además, se verá el paso de mensaje en cada uno de los casos y los posibles errores de la aplicación.

4.4.1 Funcionamiento

Se han realizado tres archivos que el usuario que ejecutará desde la Raspberry pi para diferentes funciones. Dos de ellos son usados comunes por el usuario, uno funciona para medir y otro para mostrar datos de las medidas. Por último, el archivo que es tendría que ser usado por un administrador. Este archivo permite configurar las bases de datos a través de un archivo en Python. Al final, los tres archivos son necesario para realizar todas las funciones de la aplicación.

4.4.1.1 Procedimientos para medir

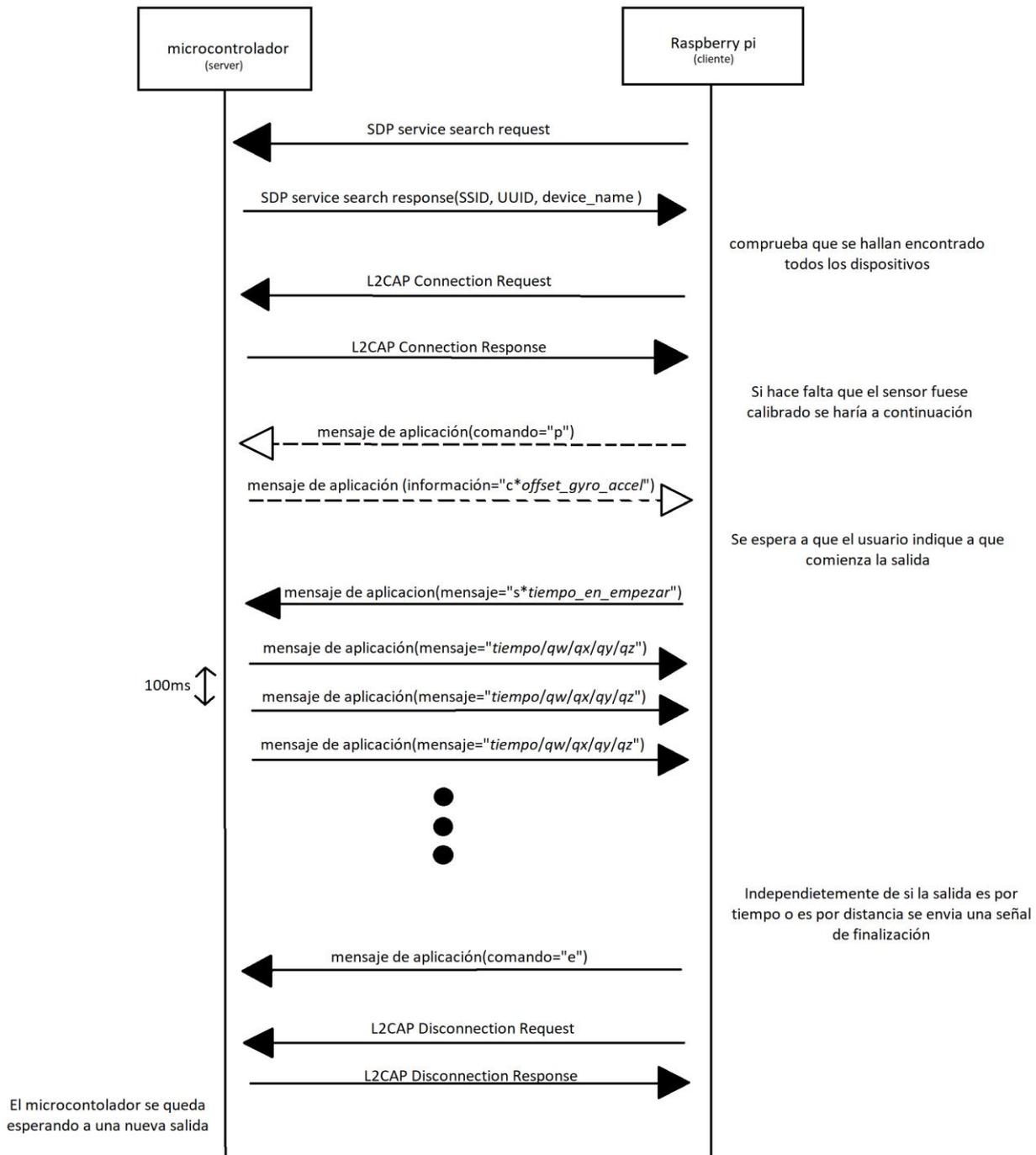
La aplicación comienza preguntando si la salida necesita que se haga una calibración previa o si no. También da la opción de eliminar los *offsets* de la calibración anterior y que los sensores esperen a ser calibrados. En cualquier caso, la aplicación va a buscar los dispositivos Bluetooth cercanos. Para identificar a cada **IMU** correctamente, los nombres Bluetooth de todos los dispositivos siguen un patrón. El patrón sería ESP32testx_y donde *x* e *y* sería el número de la **IMU** y del equipo relativamente. Por lo tanto, al nombre se le pondrá los números de la **IMU** asociado a cada microcontrolador. El número de equipo será el mismo para todos, ya que todos sirven para el mismo fin. Una vez buscado todos los dispositivos con ese patrón, miran si estos los cinco. Ahora que ya se tiene identificados a los dispositivos y su dirección, ya se puede conectar a ellos. Esta conexión va sobre **RFCOMM** (Protocolo de comunicación de radiofrecuencia). Deben estar conectados los cinco **IMUs** si no, no es posible fusionar los datos para hallar los ángulos buscados. Una vez que se compruebe que las 5 **IMUs** están conectados, datos sobre el funcionamiento de la salida. Así que preguntará cuanto tiempo tardará la salida y cuando tiempo hay que esperar para que empiecen. Si no se le introduce ningún tiempo, el dispositivo medirá hasta que tú se lo indiques. Esto sirve para realizar medidas de distancia en vez de tiempo, aunque la distancia es medida de forma manual. El tiempo de espera es por si quieres realizar una salida en barco en marcha y dar el aviso de que va a comenzar. A continuación, se calibran los sensores si se ha indicado que se haga.

Para calibrar los sensores, el programa crea hilos para las comunicaciones con cada microcontrolador. Todos los hilos tanto como pasarle parámetros como para recibir resultados son comunicados a través de una cola. Cada hilo le manda el comando para calibrar al sensor correspondiente. Los sensores son calibrados al principio del programa del microcontrolador. Estos no se vuelven a calibrar hasta que se reinicia el programa del microcontrolador. Por lo tanto, el comando solo se puede enviar mientras que los microcontroladores están en la fase inicial. En caso contrario, el microcontrolador simplemente ignorara la orden y esperara a la siguiente salida. Cuando acabe el proceso de calibración, el microcontrolador devolverá un mensaje junto a los *offsets*. Estos serán almacenados como información de la salida. En caso contrario, si algo no se ha inicializado bien en la **IMU** o que la comunicación se caiga enviará un error. Si salta un error en el calibrado, entonces se detendrá la ejecución de los demás hilos y del procedimiento. Sin embargo, el microcontrolador se quedará esperando que la Raspberry pi lo reinicie. Y no aceptará ninguna orden más.



Ilustración 4.4-1. Interfaz creada para realizar una medida. Se puede ver como todos las **IMUs** han tenido que estar conectados para calibrar los sensores y empezar a medir.

Después del posible calibrado, el dispositivo se queda esperando a que se le indique que ha comenzado la salida. Cuando indica que comience la salida, se crean los hilos clientes de las conexiones de cada microcontrolador. Por lo tanto, a cada hilo se le pasa el socket de la conexión creada anteriormente y el tiempo donde empieza y termina. Los hilos mandan la orden de que empiece a transmitir los datos de la **IMU** correspondiente. El microcontrolador empieza a leer datos de la **IMU** cada 100ms. Estos datos de la **IMU** son los cuatro números que forman un cuaternión. Posteriormente, este empaqueta los datos del tiempo en que se ha leído y los datos de la **IMU** y los envía a su hilo. A partir de aquí, el comportamiento será diferente si la salida es por tiempo a que si la salida es por distancia. Si la salida ha sido por tiempo, simplemente se van almacenando los datos hasta que llegan al tiempo de finaliza. Entonces, envía el comando de finalización a cada uno de los microcontroladores. Estos se ponen a esperar la orden de hacer otra salida de nuevo. Sin embargo, si la salida es por distancia, se espera hasta que el usuario indique la finalización de la salida. Entonces, este le marcará a cada hilo que en ese instante cuando ha acabado y eliminará los datos posteriores a este. En cualquier caso, se puede producir un error en la transmisión de los datos. Si eso llega a ocurrir, se eliminarán todos los datos enviados por ese microcontrolador. Si los datos de un microcontrolador no llegan se eliminarán los datos de la salida completa. Para finalizar, se rellena la **BBDD** con todos los datos correspondiente a la tabla salida. Esto es realizado a través de un formulario. Una vez creadas esa entrada, se almacena en una entrada de la tabla medida los ángulos suministrados. También almacenaríamos los datos de calibración que se ha usado para cada sensor. En el caso de no haberse calibrado desde el principio este utilizará la calibración de otra salida ya creada. En el siguiente esquema se puede observar el paso de mensaje entre el controlador y la Raspberry pi.



Esquema 4.4–1. Diagrama de paso de mensaje entre un microcontrolador y la Raspberry pi teniendo un funcionamiento correcto.

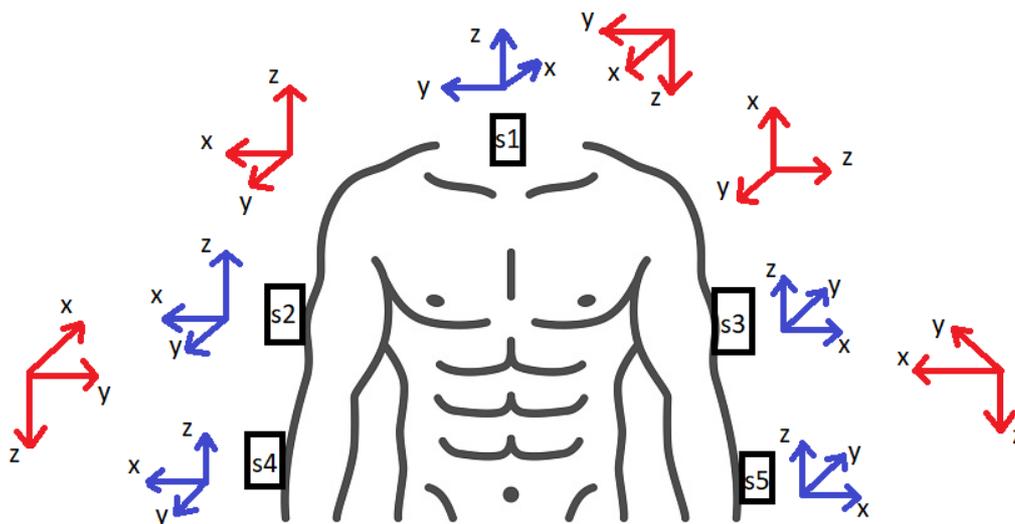
Si el microcontrolador no pudiera leer del registro del DMP, entonces informaría a la Raspberry pi. Enviaría una letra “F” en el mensaje, que haría que se detuviera la lectura de la salida. Ocurriría igual en el caso de que no se pueda inicializar el DMP en el caso de la calibración. Para finalizar se hablará sobre el reinicio ya que se ha visto que es necesario en la medición. Debido a que, si se produce un error en la calibración o queremos calibrar de nuevo, es necesario este paso. Se buscan los dispositivos en Bluetooth. Y para los dispositivos que se encuentran con los nombres Bluetooth dicho anteriormente. Entonces se realiza una conexión con ellos y se le manda la orden de reinicio. Esta puede ser suministrada en cualquier punto en el que se encuentre el microcontrolador. El

mensaje que enviaría en caso de reinicio sería “r”.

4.4.1.2 Configuración de la base de datos

Esta funcionalidad sirve para introducir o cambiar los datos de la base de datos. Estos son solo los datos que puedan necesitar ser introducidos o modificados por el usuario. Por ejemplo, ni los sensores ni las articulaciones tienen que ser modificados manualmente por los usuarios. Las opciones que permiten modificar e introducir serían los piragüistas, los barcos y las palas manualmente. Esto es debido a que el piragüista y su equipamiento son datos que se obtienen del administrador. Los datos de la salida solo se pueden modificar manualmente. Ya que la aplicación solo permite crear salidas en el momento de realizarla. Y los datos de los ángulos te permite generarlo automáticamente dando simplemente la salida a la cual pertenecen. En el caso de introducir datos simplemente se tiene un formulario con los datos de la base de datos. Y una vez que este es rellenado, se crea una entrada en la entrada con esos datos en la tabla correspondiente. Los datos no rellenados del formulario simplemente tomarán su valor por defecto. Para modificar una tabla primero buscas la entrada de la tabla, para ello se rellena otro formulario. En este se introducen valores que sepas de la entrada que estás buscando. Estos formularios variarán dependiendo de la tabla en la que la estés buscando. Una vez rellenado el formulario, este te pasará una lista con todas las entradas que cumplen esos valores en la tabla. Una vez se le indique la entrada buscada, se pasará otro formulario en el que modificaras los parámetros que quieras. Y estos serán los que cambian en la **BBDD**. Solo se cambiarán los modificados, los demás se quedarán igual. Las interfaces que se describieron se pueden ver ejemplos de ellas en el anexo 7.1.

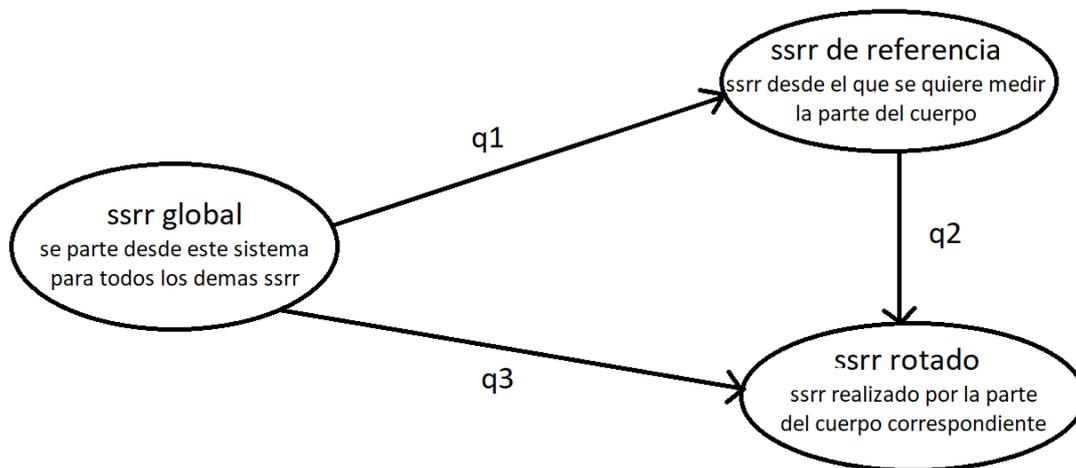
Por último, se va a explicar cómo se crean las entradas de la tabla ángulo. Para empezar, el usuario tendrá que realizar el mismo de búsqueda que cuando modificas una tabla. Una vez encontrada la entrada, busca todas las mediciones almacenadas en la tabla medida de esa salida. Una vez obtenido esos datos se procederá a realizar la fusión de los datos. Para la fusión de los datos, se ha creado una máquina de estados. Donde a partir de las medidas de las **IMUs** y de la posición inicial nos dará los valores de los ángulos. La posición inicial de los sensores se puede ver en el Esquema 4.4-2. La posición inicial que se ha supuesto es la de torso extendido y brazos estirados hacia abajo. En esta imagen se puede ver los sistemas de referencia de cada sensor en azul. Y el sistema de referencia desde el que queremos partir para hallar nuestro ángulo de esa articulación en rojo. Todo esto teniendo en cuenta que nos encontramos en la misma situación.



Esquema 4.4-2. Esquema de tren superior en estado inicial con los sensores colocados en la parte correspondiente del cuerpo. Los ssrr (sistemas de referencias) azules corresponden a los ssrr iniciales de los sensores y los rojos a los ssrr que deben tener las articulaciones,

Como se ha mencionado estos ángulos son construido a través de una máquina de estado. La máquina de estado general se puede encontrar en el Esquema 4.4-3. Aunque esta solo es una simplificación de las demás máquinas de estado que se aplicó. Para empezar, la máquina de estado se basa en un principio. Y es que todos los sistemas

de referencia deben de partir de un único sistema de referencia. A este se le ha llamado sistema de referencia global. Ya que se necesita tres máquinas de estado diferentes, estas la podemos adaptar para cada caso. Son tres debido a que el patrón se repite si es izquierda o derecha.

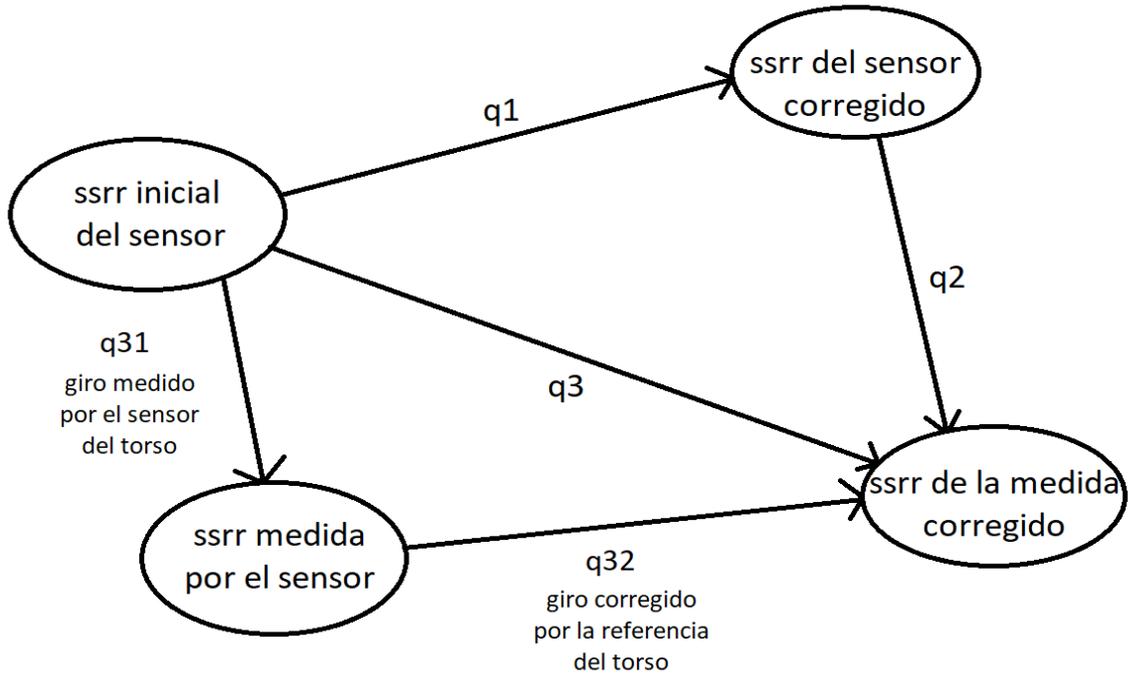


Esquema 4.4-3 . Máquina de estado general que se ha seguido a la hora de fusionar los datos.

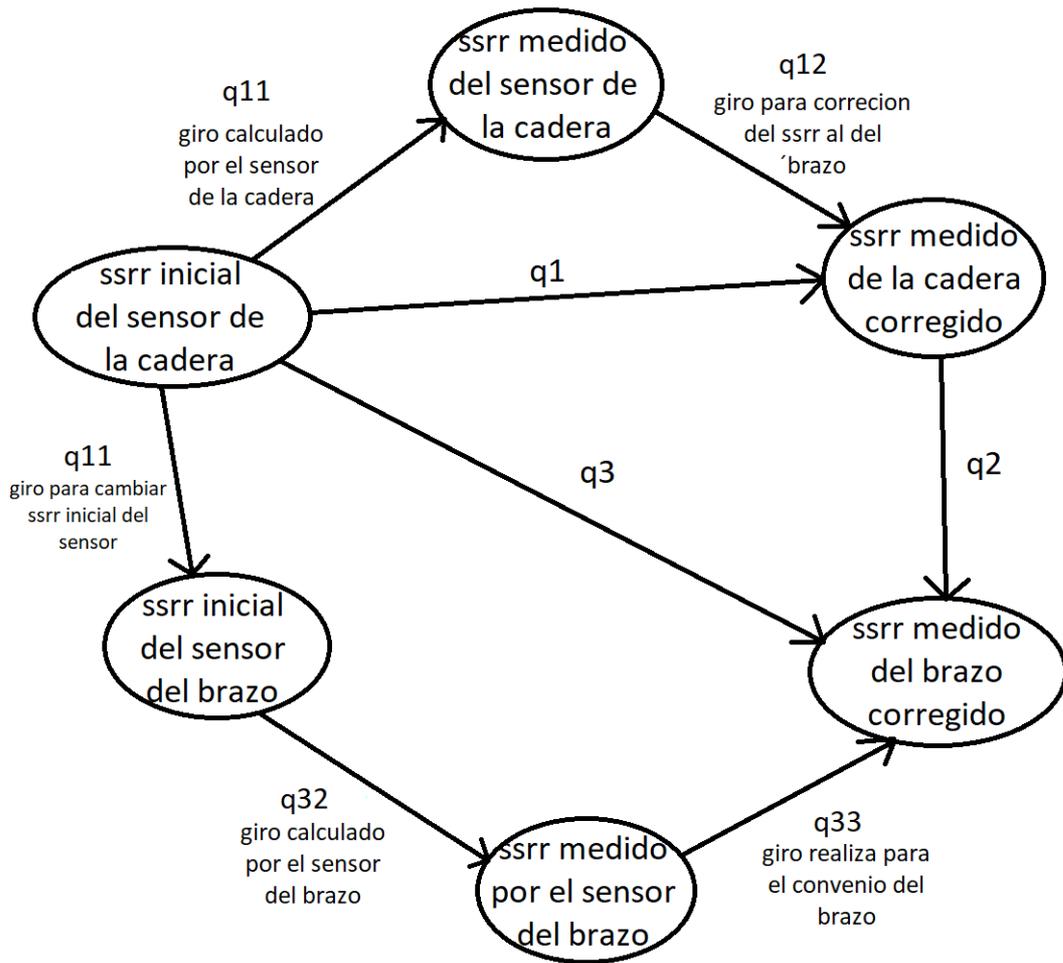
Para la cadera se toma como sistema de referencia global el suyo inicial. Para el caso de los hombros se toma como sistema de referencia global el del sensor de la cadera inicialmente. Y para los codos se toma los sistemas de referencia de sus respectivos sensores del brazo inicialmente. A partir de ahí se saca por una parte el sistema de referencia desde donde partimos. En el caso de la cadera, este será el sistema de referencia inicial modificado por llevar una nomenclatura. En el caso de los hombros, este dependerá de la medida de la cadera y será modificado por la misma razón. Y en el de los codos, este dependerá de las medidas de los brazos correspondiente y será modificada también. Y por otra parte se tiene la medida que se ha realizado en cada parte desde el sistema de referencia global. Por lo tanto, las máquinas de estado quedan como reflejan las siguientes imágenes. Los cuaterniones se han supuesto que hacen girar a los cuaterniones. Eso hay que tenerlo en cuenta para los cálculos que se vayan a realizar. El giro que interesa es aquel que va desde el sistema de referencia de referencia hasta el medido. Es decir, interesa el cuaternión q_2 , que podéis ver en el Esquema 4.4-4, Esquema 4.4-5 y Esquema 4.4-6. Se sabe tanto q_1 como q_3 . Es decir, se conoce el giro necesario para llegar al sistema referencia medido y de referencia desde el global. Por lo tanto, para llegar a q_2 , se debe tener en cuenta la formula (4.4-1).

$$q_2 q_1 = q_3 \rightarrow q_2 = q_3 q_1^{-1} \quad (4.4-1)$$

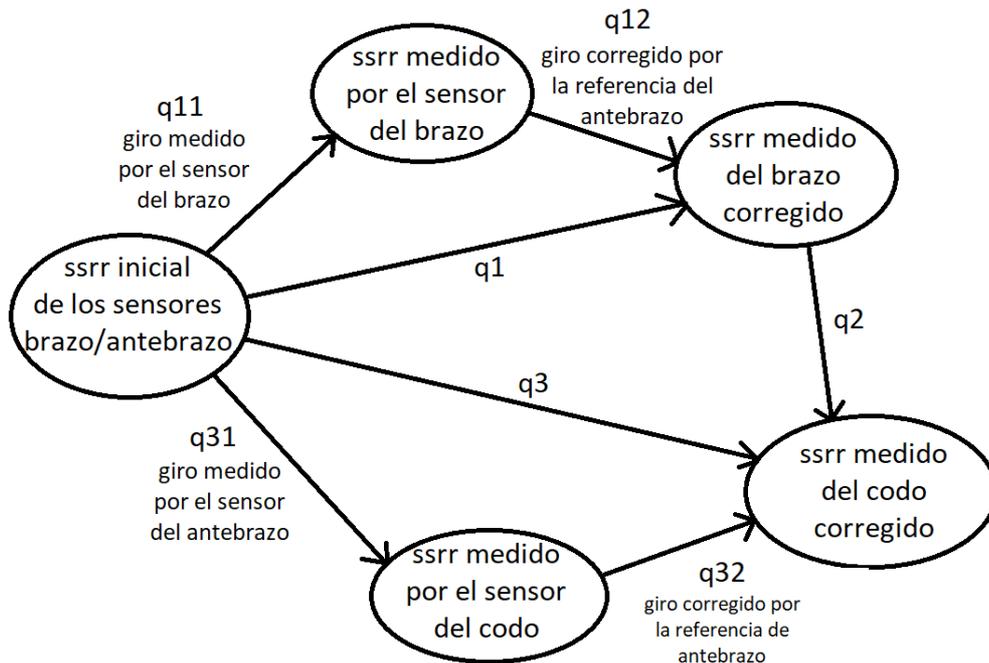
Teniendo en cuenta que los cuaterniones deben ser unitario para considerarse de rotación. Tenemos que la inversa de un cuaternión en este caso es igual a su conjugado. Para finalizar se calcula el ángulo que ha girado cada eje mediante los ángulos de Euler extrínsecos (98). Esto se realiza para facilitar la representación en las gráficas. Estos son hallados mediante las ecuaciones (3.4-10), (3.4-11) y (3.4-12)



Esquema 4.4-4. Máquina de estado para adquirir los ángulos de la cadera



Esquema 4.4-5. Máquina de estado seguido para adquirir los ángulos realizados por cada hombro.



Esquema 4.4-6. Máquina de estados seguido para adquirir los ángulos realizado por cada codo.

Las fórmulas que se ha aplicado para las anteriores máquinas de estado serían las siguientes. La primera ecuación pertenecería a la cadera, la siguiente al hombro y la siguiente al codo. Estas ecuaciones se corresponden con Esquema 4.4-4, el Esquema 4.4-5 y el Esquema 4.4-6, respectivamente. Aunque para calcular el ángulo de Euler no se han aplicado estas sino la conjugada. Ya si te fijas en los esquemas lo que gira siempre es el sistema de referencia. Sin embargo, para los cuaterniones lo normal es que gire el vector. Entonces las ecuaciones (3.4-10), (3.4-11) y (**3.4-12**) se refieren a las componentes de un cuaternión. Considerando que ese cuaternión gire un vector en el espacio.

$$q_2 = q_{32}q_{31}q_1^* \quad (4.4-2)$$

$$q_2 = q_{33}q_{32}q_{31}q_{11}^*q_{12}^* \quad (4.4-3)$$

$$q_2 = q_{32}q_{31}q_{11}^*q_{12}^* \quad (4.4-4)$$

4.4.1.3 Analisis de los Datos

Para realizar este paso como dije antes, sé crea un archivo dedicado a este procedimiento. En él, se muestran varias gráficas cuyo objetivo es mostrar las características de la palada en una salida. En las primeras gráficas están los ángulos de los movimientos de las articulaciones en el tiempo. Dichos ángulos son los ángulos de Euler extrínsecos, los cuales representan un movimiento cada uno de ellos. Estos datos se han sacado de la tabla ángulo que debe haber sido rellena anteriormente. Cada una de las articulaciones realizan unos tipos de movimientos u otro. Las articulaciones que se han medido tienen seis grados de libertad. Con lo que se puede medir los movimientos con solo tres ejes del movimiento. Sin embargo, el movimiento del hombro tiene seis grados de libertad. Por lo tanto, se necesitaría un sensor más por cada hombro para poder medir los seis. Pero en nuestro

trabajo solo se utilizan tres de estos movimientos. Debido a que se puede representar adecuadamente el movimiento del hombro con esos tres. De estos datos, se puede ver datos básicos como la rotación de cadera que ha realizado el palista. Y sobre todo se puede observar el recorrido que ha medido el sensor.

Después, se procede a sacar nuevas graficas que den más información acerca de la palada. Se ha considerado que una palada es el resultado de un conjunto movimiento actuando paralelamente. Estos movimientos son cada uno de los realizados por cada una de las articulaciones. Debido a que el deportista tiene un comportamiento similar en todas sus paladas. Algunos movimientos de las articulaciones dependerán de otros ya que el movimiento sigue un patrón. Las dependencias de los distintos movimientos seleccionados describirán el patrón que sigue el deportista. Con este pensamiento en mente y pensando en la técnica de piragüismo se ha considerado posibles gráficas a estudiar. Se empezará con el movimiento del codo lo hacemos depender de los movimientos del hombro. En concreto, se puede ver la dependencia de la flexión del codo con la abducción y flexión del hombro. Esto es debido a que a través de dichos movimientos del hombro se puede hallar la fase en la que se encuentra. Ya que cuando este lanza el brazo hacia delante y cuando lo retrae se quiere ver lo estirado que esta. A parte interesa ver, la posición de este durante la fase del ataque aérea. Luego a partir del movimiento de abducción del hombro se puede observar las fases de salida y el ataque aéreo. Se espera que las gráficas tengan un comportamiento cíclico en algún aspecto.

La siguiente gráfica mide como se realiza la rotación de cada brazo en la palada. Para ello se hace depender del movimiento de rotación de la cadera y flexión del brazo. Esto se debe a que esos dos movimientos dan una referencia acerca de cómo se ha realizado las dos fases. Tanto la fase de rotación como la de ataque, ya que estos dos movimientos son fundamentales en cada uno de ellos. Y el resultado da una descripción del carácter cíclico de la palada en estas dos fases. Por lo tanto, se espera un movimiento ovalado en la que se pueda ver ambas fases. Para finalizar se obtendría, por así decirlo, la eficiencia de la rotación. Y es que se supone que se rota para hacer más uso de nuestros músculos del torso y menos del brazo. Por lo tanto, cuando se rota la flexión del brazo debería ser menor para de esa forma cargar más trabajo al torso. Este al tener más musculo hace que se distribuya más la carga. Entonces se toma el movimiento de la cadera como referencia de la rotación. Y a continuación, se ve cuanta flexión se ha realizado sobre todo en las fases de ataque y rotación. Y hay que decir que hay dos gráficas por cada modelo uno para el brazo derecho y otro para el izquierdo.

Para obtener información más sencilla acerca del modelo anteriores, se realiza una regresión. De tal manera que mediante este se permita describir las características básicas de las gráficas anteriores. Estas regresiones se realizan mediante una inteligencia artificial que permita sacar los parámetros de esta. En el caso de que haya dos variables independientes, la regresión será un plano y el modelo tendrá dos coeficientes. En el caso de una variable independiente, la regresión será una recta y solo habrá un coeficiente. En cualquiera de los casos, habrá el punto que corta al eje de la variable dependiente. Este punto es llamado punto de intercepción. Con estos datos, se podrá ver una representación lineal más fiel a los puntos dados. Sin embargo, para ver cuánto se parece a la realidad se calcula el error cuadrático media y r^2 . El error cuadrático medio da una medida de cuál es la diferencia de los puntos dados a los dados por la regresión. Y r^2 dice la proporción de varianza de la variable dependiente dados los datos de la variable independiente. Con todos estos parámetros se puede describir como ha sido esa característica en concreto. Se ha hecho uso la librería *Scikit learn* de Python para implementar dicha inteligencia artificial. De tal forma que la utiliza para resolver un problema de regresión. Se ha aplicado un modelo lineal aplicando la regla del mínimo cuadrado para el cálculo de parámetros. Esta se supone que podría ser débil ante variables de entradas corraladas. Por ello se aplicó el modelo *Ridge* con validación cruzada para el cálculo del parámetro α . Sin embargo, este dio un mayor EMS que la primera solución. Es por ello por lo que se ha utilizado el método de mínimos cuadrados para crear la regresión lineal.

4.5 Conclusión

En este apartado, se ha visto el funcionamiento de nuestro trabajo paso por paso. Además, se muestra dónde y cómo se almacena la información. A parte, se puede observar la arquitectura seguida de los diferentes dispositivos y la comunicación entre ellos. Y como se muestran los resultados obtenidos. En el siguiente apartado, se hablará sobre las pruebas realizadas a este sistema y de los resultados obtenidos.

5 VALIDACIÓN DEL TRABAJO

5.1 Introducción

En este apartado, se hablará sobre las pruebas realizadas a los deportistas. En estas pruebas se siguen el procedimiento visto en 4.4. Sobre todo, cuáles se han hecho y qué procedimiento se han seguido a la hora de realizarlas. Luego, se verán los resultados obtenidos de las diferentes salidas realizadas. Y se comentará dos casos de todos los resultados.

5.2 Medidas

Las medidas son el proceso necesario para tomar los datos necesarios que se analizarán más adelante. Para ello, se han realizado unas salidas en el río a diferentes piragüistas con las unidades de medidas equipadas. Se va a describir el procedimiento de las pruebas realizadas.

5.2.1 Preparación

Para la realización de las medidas se ha utilizado una especie de traje para colocar los módulos sensores. Estos módulos sensores estarían formados por la conexión **I2C** (Inter-Integrated Circuit) del microcontrolador con el **IMU** (*Inertial Measurement Unit*). Como se puede ver en la Ilustración 5.2-1 y la Ilustración 5.2-2 se ha hecho uso de una licra y unas cintas para crear estos trajes. Tanto la licra como las cintas incorporan pequeños bolsillos para introducir los módulos sensores. La licra lo incorpora justo en la parte de detrás del cuello. Estos bolsillos deben ser estrechos para no permitir el movimiento de los módulos dentro de estos. Estos módulos sensores están instalados en un pequeño recipiente cerrado para impedir la entrada de agua. El **IMU** está fijado mediante tornillos y tuerca al recipiente para no permitir el movimiento dentro de este. Y el módulo está alimentado mediante una pequeña batería. La Raspberry pi está metida dentro de una bolsa y en un recipiente para evitar ser mojada. Y es alimentada con una batería externa para dispositivos móviles que permite carga rápida. Esto es debido a que es necesario 2.5 A para alimentar la Raspberry pi. De esa forma alcanzamos la intensidad que necesita la Raspberry pi para funcionar. Antes de ponerla en el barco se tiene que asegurar que la Raspberry pi esté en la red wi-fi del dispositivo móvil.



Ilustración 5.2-1. Traje realizado para fijar los sensores a cada parte del cuerpo.

El deportista debe vestir la licra, como se puede ver la Ilustración 5.2-2. Y las cintas deben ir colocados debajo de los hombros y de los codos ya ambos en su lado exterior. Se toma el convenio de que todos los sensores deben de tener los sensores mirando hacia arriba. Y se colocan de manera que el primero vaya en el bolsillo de la licra. El segundo y el tercero van justo debajo del hombro derecho e izquierdo relativamente. Y el cuarto y el quinto van justo debajo del codo derecho e izquierdo relativamente. Una vez que todo está listo se procede a realizar la calibración de los dispositivos. Para ello, se indica al deportista que se quede en la posición que puede verse en la Ilustración 5.2-2. Esta postura es la que se puede ver en el Esquema 4.4-2. Es decir, con el torso recto y los brazos extendido hacia abajo con las palmas hacia dentro. Entonces se iniciará el programa para medir, marcando la opción de calibrado. Y cuando termine el calibrado el programa indicará que está listo para realizar las medidas. Entonces el deportista podrá subirse al barco.



Ilustración 5.2-2. Deportista equipando los sensores. Y está tomando la postura inicial para calibrar los sensores

5.2.2 Realización

Las salidas se realizarían manteniendo al principio de la salida la posición de calibrado anterior. Es decir, con el torso estirado y con los brazos estirados hacia abajo. Esta posición debe de mantenerse antes de empezar a remar. Una vez empezado a remar no se podrá detener en ningún momento o la salida será invalida. Esto es debido, al error proveniente de los sensores si se queda quieto durante un tiempo. Ya que el valor del ángulo de giro tenderá a cero. Por lo tanto, la postura que tome a continuación será la nueva posición inicial. Y los cálculos para fusionar los datos fallaran.

Se realizaron medidas a cuatro piragüistas de mi club, 2 con experiencia y 1 con poca experiencia. Las salidas son de dos minutos de duración y se realizaron entre 2 a cuatro salidas a cada uno. En el caso del novato se ha contado con la ayuda de un barco que iba a su lado para seguirle de cerca. En el caso de los dos con más experiencia se fue con una bicicleta por la orilla y usando el móvil. En ese caso se tenían que quedar pegados a la orilla o si no se perdía la señal del wi-fi. En el siguiente video podéis ver a una de las personas realizando una de estas salidas. Se ha podido realizar un video en el caso del sujeto que tenía menos experiencia mientras realizaba la prueba. Este video es el Video 5.2-1, que puede verse a continuación. En la Ilustración 5.2-3, la Ilustración 5.2-4, la Ilustración 5.2-5 y la Ilustración 5.2-6 se puede ver los datos de las salidas. En concreto, se ven los datos de los piragüistas, barcos, palas y las propias salidas, respectivamente.



Video 5.2-1. Realización de una salida del primer sujeto haciendo un seguimiento desde un barco.

En las salidas, en muchas de ellas no se ha partido de la postura inicial que se ha pensado. Esto ha dado que se halla la referencia haya variado. Y hace que en muchas de las salidas los datos hayan salido erróneos. Sin embargo, en algunos casos se pueden ver que se ha mantenido una postura similar a la de referencia. Además, se ha cometido un error en el rango de los sensores.

id	nombre	apellidos	altura	peso	long_torso	categoria	club
6	Eloy	Parra Boyero	1.66	68	0.48	veterano	club kayak Torneo
7	David	Rodriguez	1.76	80	0.5	senior	club kayak Torneo
8	Juan Antonio	Écija Fernández	1.64	71	0.48	veterano	club kayak Torneo

Ilustración 5.2-3. Piragüistas que han ejecutado las pruebas.

id	modelo	fabricante	propietario
2	Viper Racer	Nelo	8
3	Vanquish 3	Nelo	7
4	4	Nelo	6

Ilustración 5.2-4. Barcos de los piragüistas que han realizado las pruebas.

id	modelo	fabricante	propietario
2	IV	Braca-sport	8
3	IV	Braca-sport	7
4	ExtremeXL	Roman	6

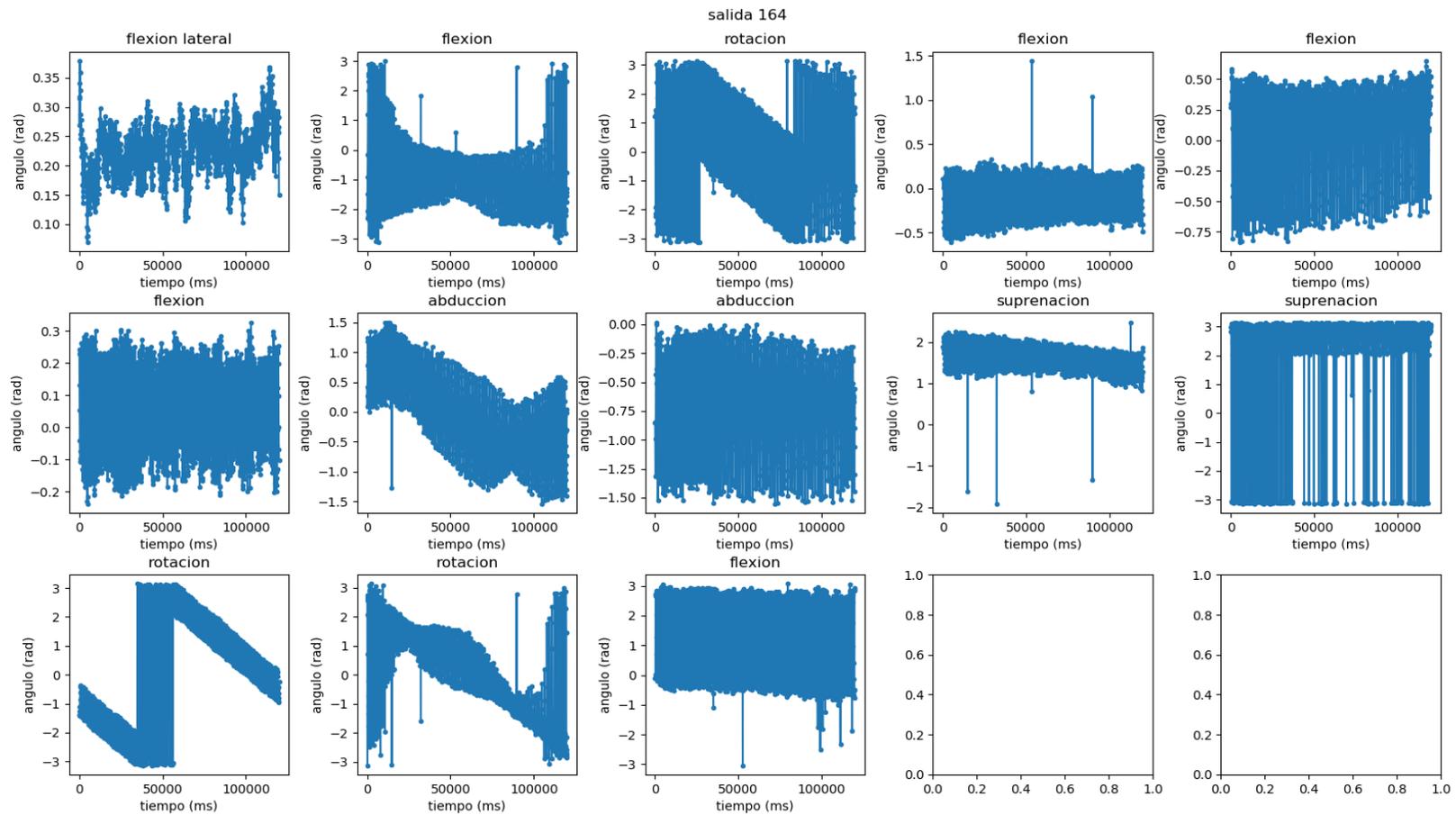
Ilustración 5.2-5. Palas de piragüistas que han realizado las pruebas.

id	id_piraguista	id_barco	id_pala	tiempo	ritmo	fecha	descripcion
164	8	2	2	120000	R1	2021-07-15	Juan Antonio 1
165	8	2	2	120000	R1	2021-07-15	Juan Antonio 2
167	7	3	3	120000	R2	2021-07-16	David 1
168	7	3	3	120000	R2	2021-07-16	David 2
169	6	4	4	120000	R2	2021-07-16	eloy 1
170	6	4	4	120000	R2	2021-07-16	eloy 2
171	6	4	4	120000	R2	2021-07-16	eloy 3
172	6	4	4	120000	R2	2021-07-16	eloy 4

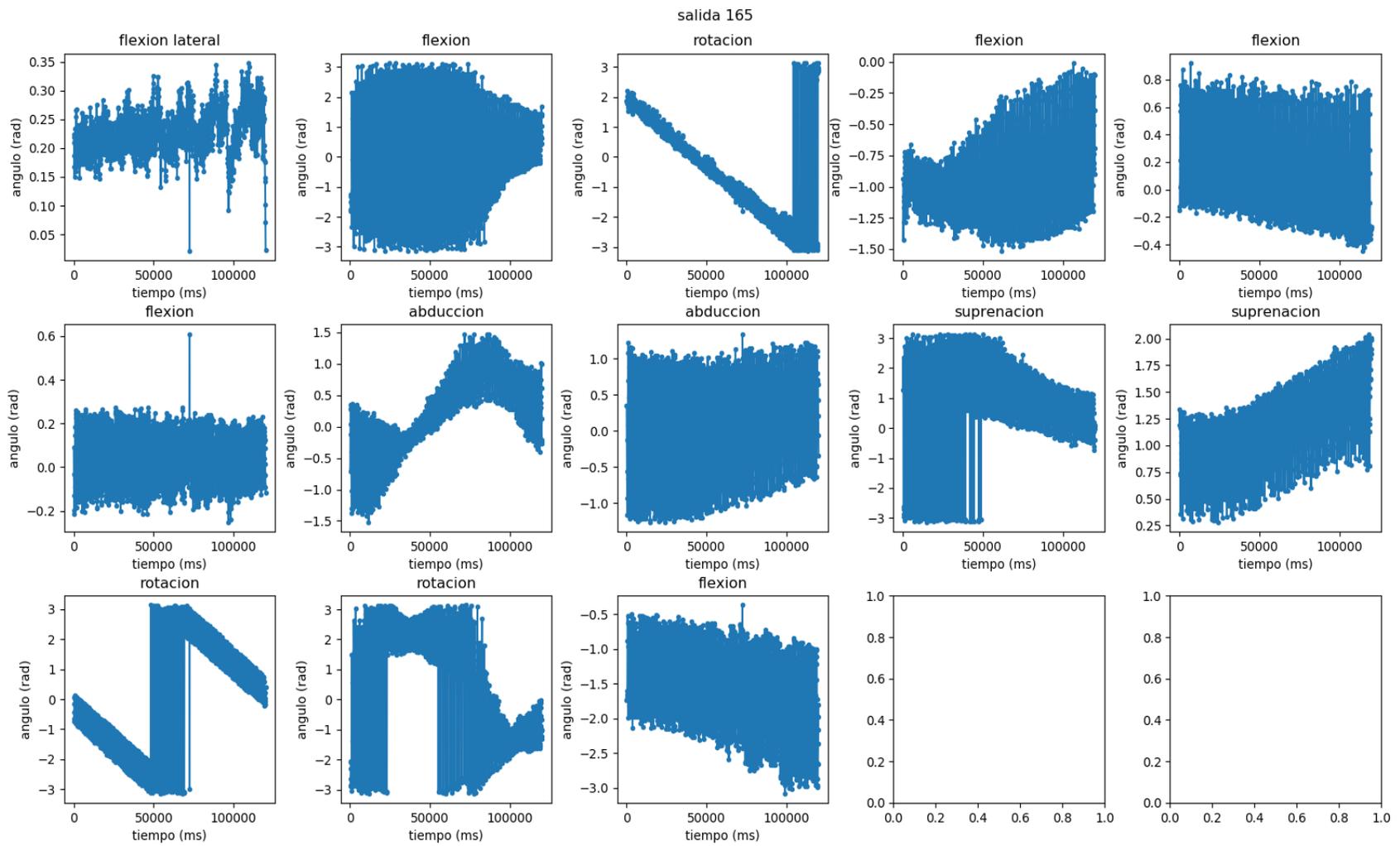
Ilustración 5.2-6. Pruebas que se han realizado.

5.3 Resultados

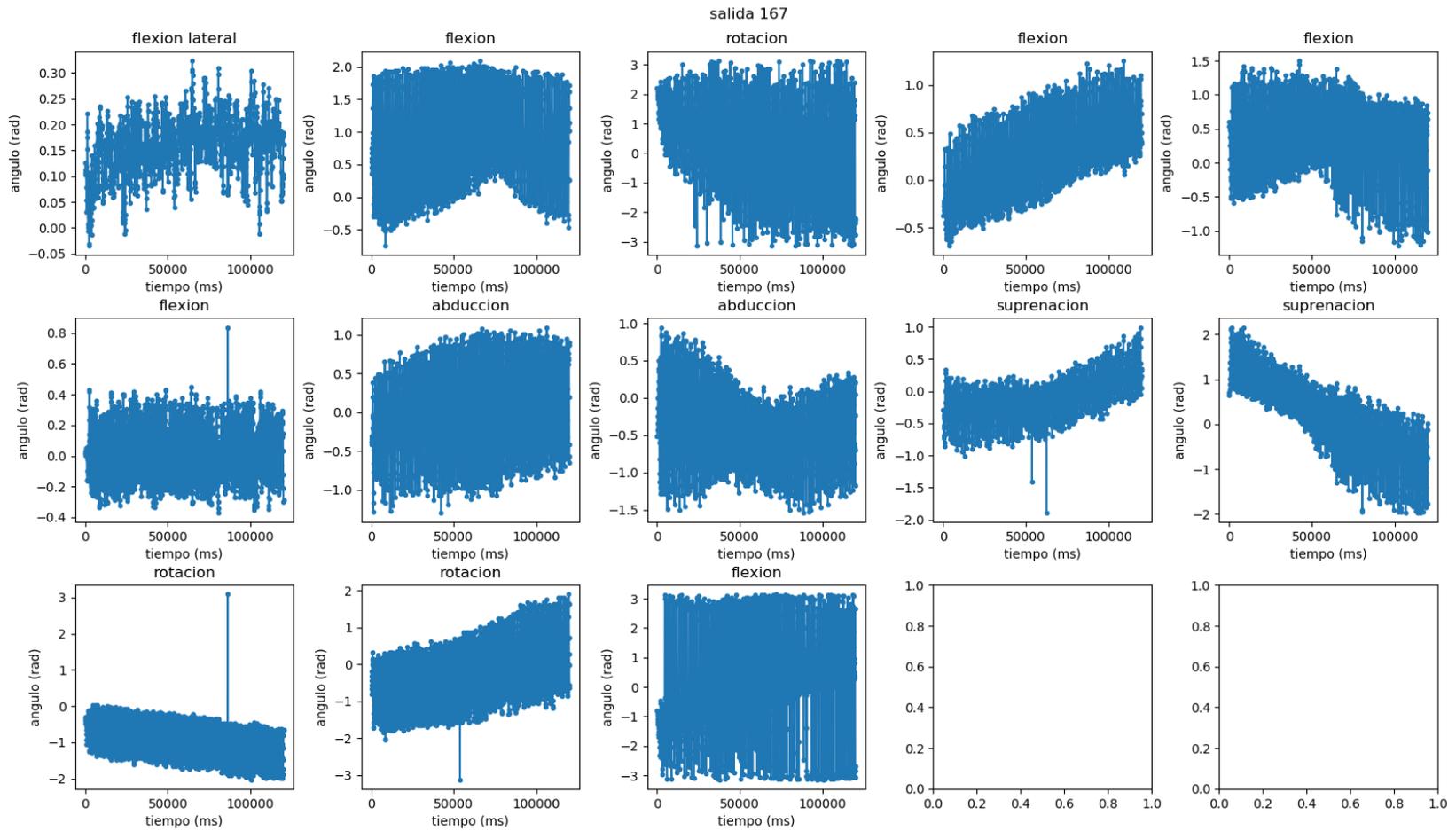
Una vez realizada las mediciones en el río. Se procedió a calcular los ángulos realizados por las articulaciones de cada uno de los participantes. Esto se realizaría siguiendo los pasos vistos en el apartado 4.4.1.2. Estos datos son representados a través del tiempo, para observar cómo han salido estos. Estos pueden verse en las gráficas que aparecen a continuación. En la mayoría de las salidas, se pueden múltiples errores que se comentarán al final del apartado. Pero en la salida 171 se puede apreciar cómo el brazo izquierdo está dentro de los límites físicos. Y lo mismo ocurre para el caso del brazo derecho de la salida 167. Y es que para imprimir las gráficas comparativas entre los ángulos mencionadas en el apartado 4.4.1.3. Se necesitaría que las gráficas de un brazo u otro de todas las gráficas dentro de los límites establecido. Se pueden saber estos límites viendo los sistemas de referencia de las articulaciones del Esquema 4.4-2. Por lo tanto, nos vamos a centrar en estudiar estos dos casos.



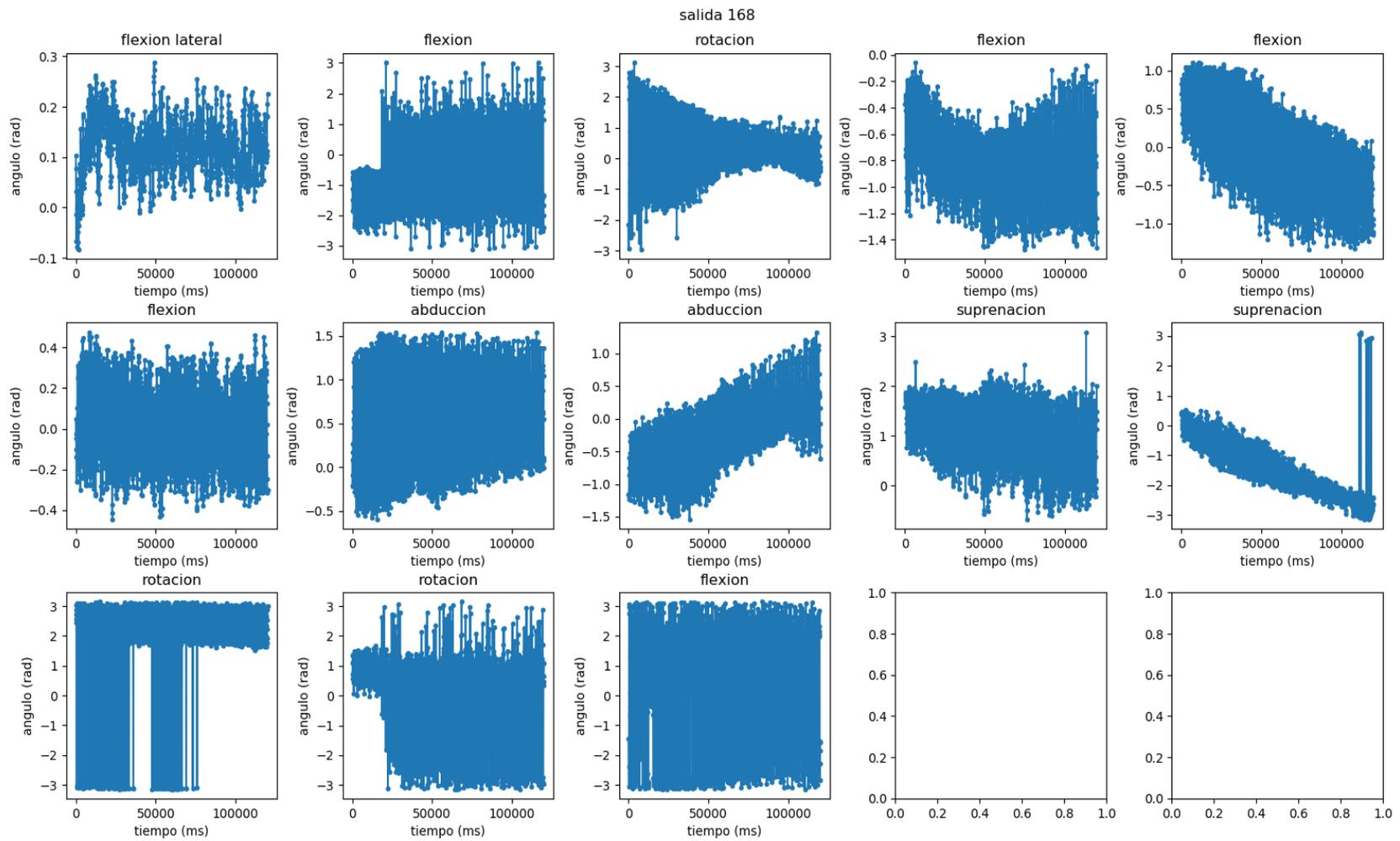
Gráfica 5.3-1. Evolución de los ángulos de los movimientos de las articulaciones a lo largo del tiempo de la salida 164. Aclaración. Las columnas son de derecha a izquierda la cadera, el hombro derecho e izquierdo, relativamente, y el codo derecho e izquierdo, relativamente. Y en el título esta el nombre de cada articulación.



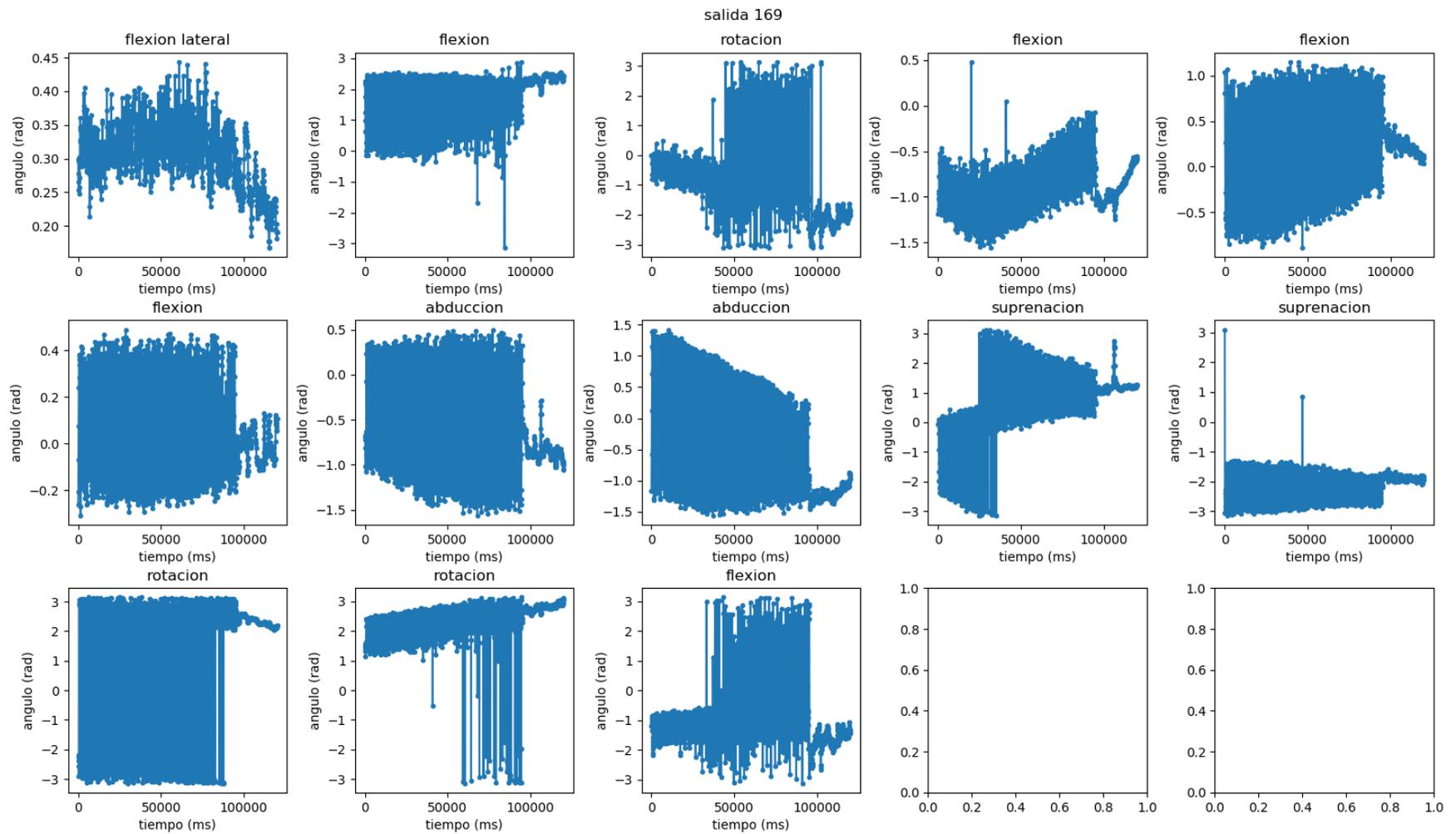
Gráfica 5.3-2. Evolución de los ángulos de los movimientos de las articulaciones a lo largo del tiempo de la salida 165. Aclaración. Las columnas son de derecha a izquierda la cadera, el hombro derecho e izquierdo, relativamente, y el codo derecho e izquierdo, relativamente. Y en el título esta el nombre de cada articulación.



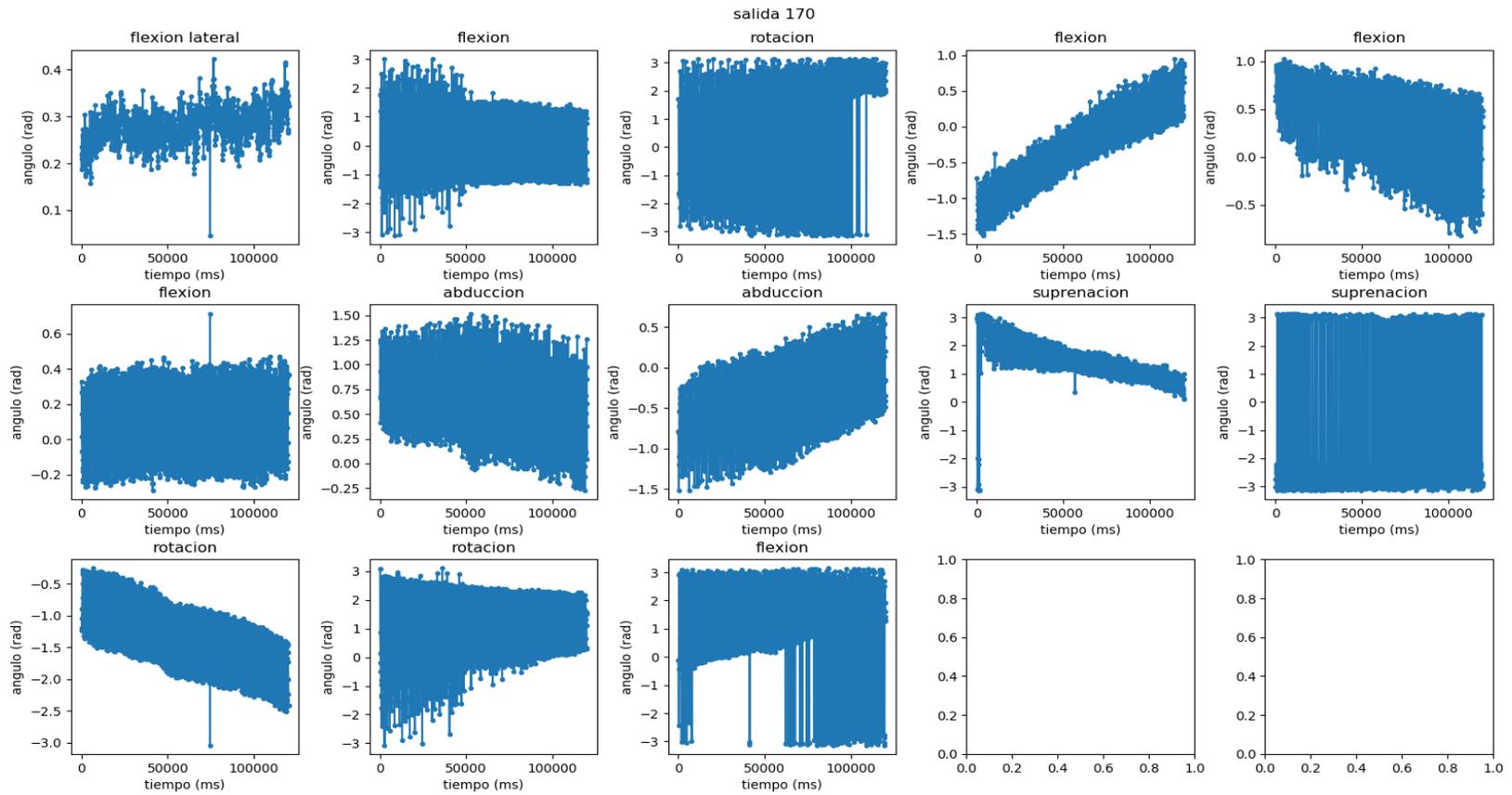
Gráfica 5.3-3. Evolución de los ángulos de los movimientos de las articulaciones a lo largo del tiempo de la salida 167. Aclaración. Las columnas son de derecha a izquierda la cadera, el hombro derecho e izquierdo, relativamente, y el codo derecho e izquierdo, relativamente. Y en el título está el nombre de cada articulación.



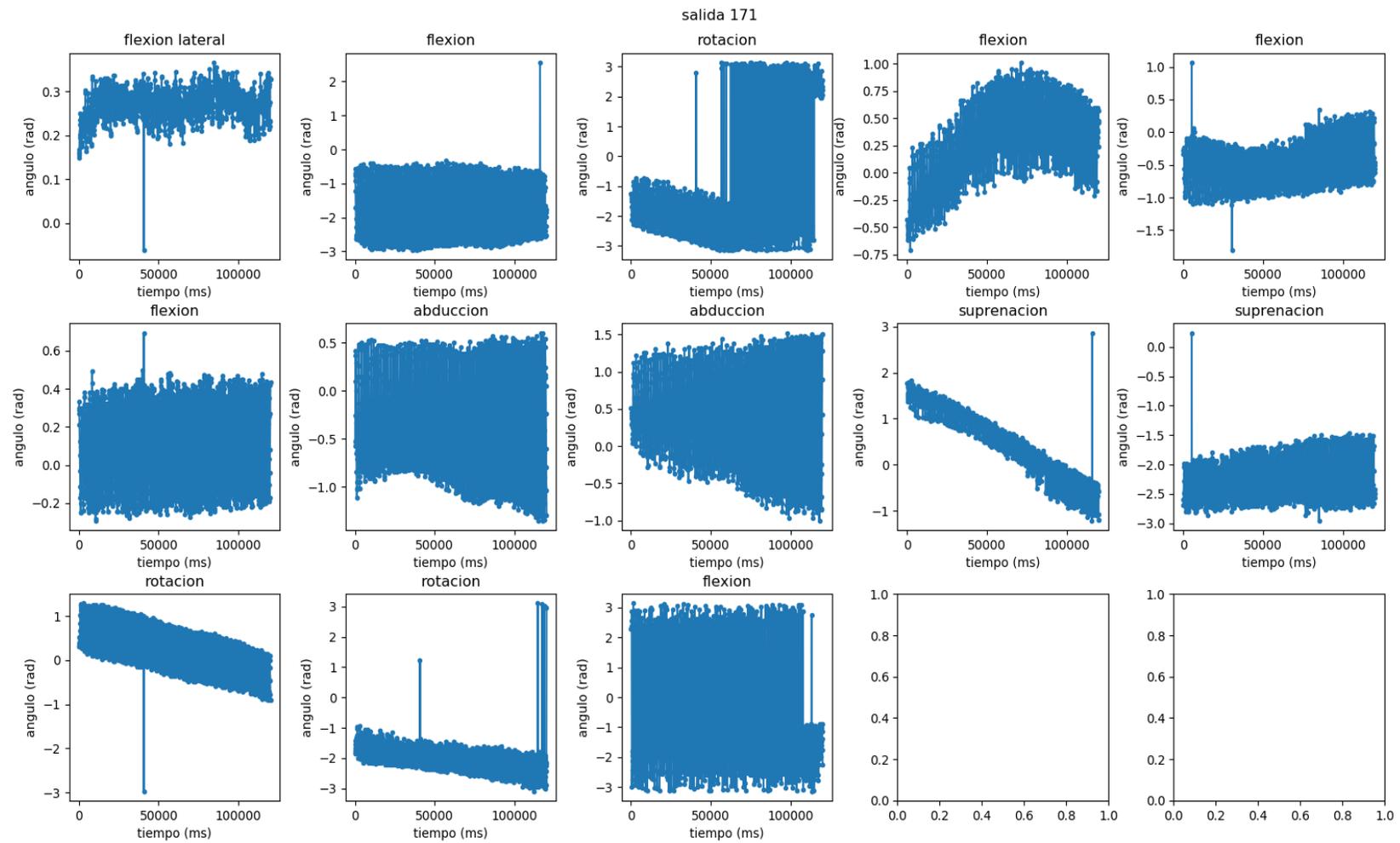
Gráfica 5.3-4. Evolución de los ángulos de los movimientos de las articulaciones a lo largo del tiempo de la salida 168. Aclaración. Las columnas son de derecha a izquierda la cadera, el hombro derecho e izquierdo, relativamente, y el codo derecho e izquierdo, relativamente. Y en el título está el nombre de cada articulación.



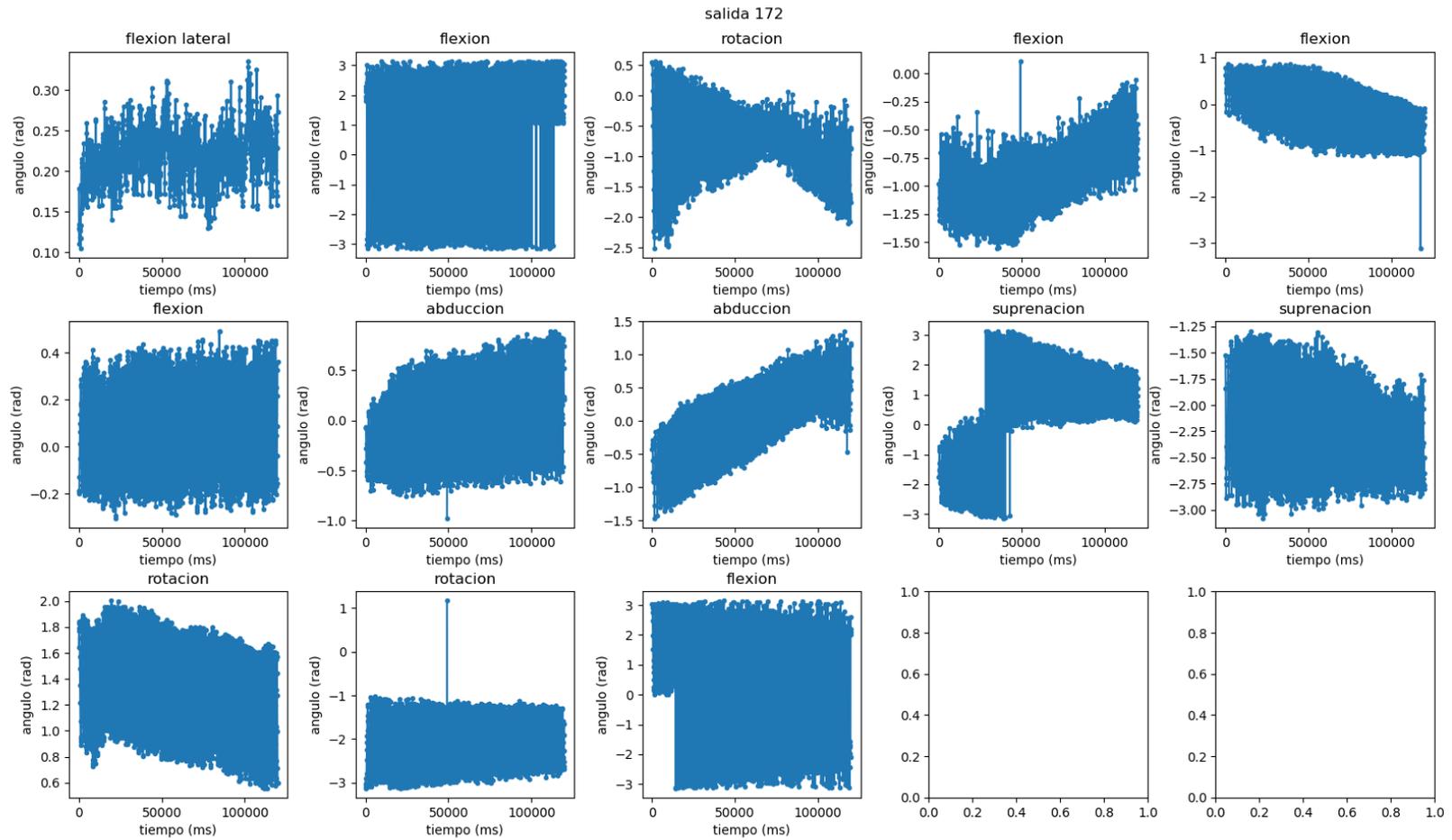
Gráfica 5.3-5. Evolución de los ángulos de los movimientos de las articulaciones a lo largo del tiempo de la salida 169. Aclaración. Las columnas son de derecha a izquierda la cadera, el hombro derecho e izquierdo, relativamente, y el codo derecho e izquierdo, relativamente. Y en el título esta el nombre de cada articulación.



Gráfica 5.3-6. Evolución de los ángulos de los movimientos de las articulaciones a lo largo del tiempo de la salida 170. Aclaración. Las columnas son de derecha a izquierda la cadera, el hombro derecho e izquierdo, relativamente, y el codo derecho e izquierdo, relativamente. Y en el título está el nombre de cada articulación.

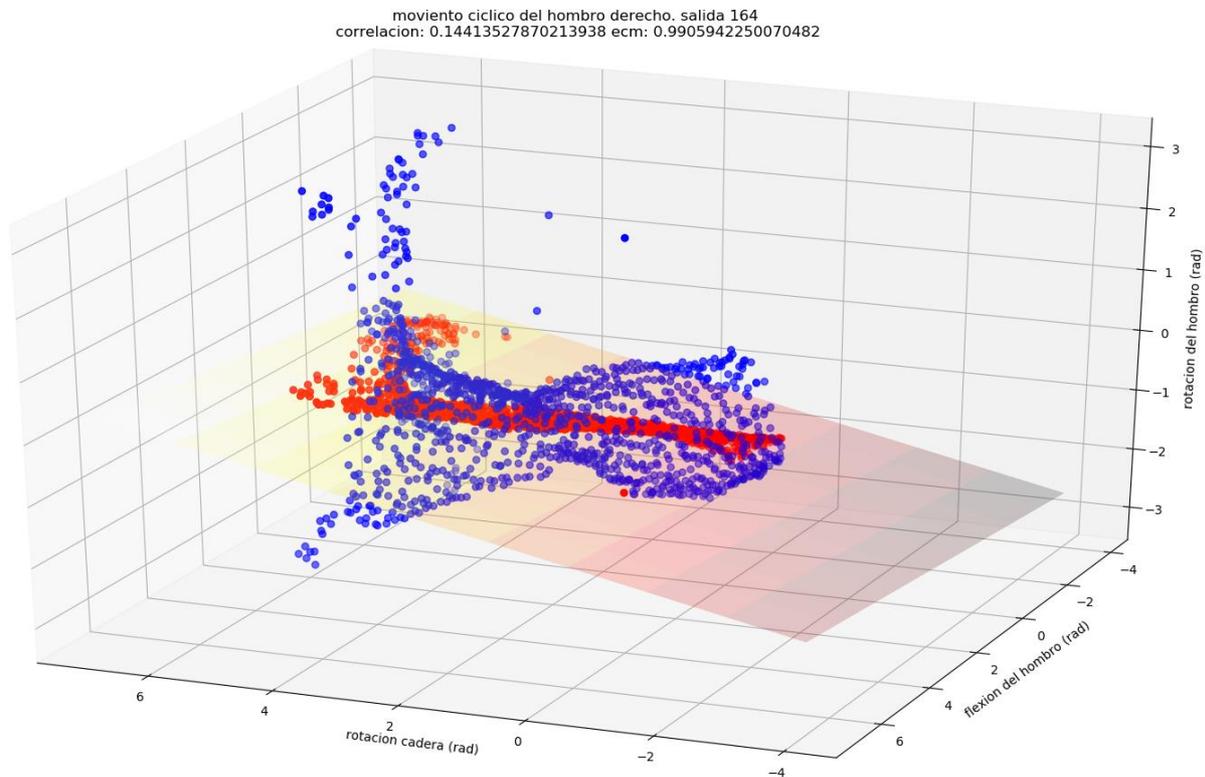


Gráfica 5.3-7. Evolución de los ángulos de los movimientos de las articulaciones a lo largo del tiempo de la salida 171. Aclaración. Las columnas son de derecha a izquierda la cadera, el hombro derecho e izquierdo, relativamente, y el codo derecho e izquierdo, relativamente. Y en el título está el nombre de cada articulación.



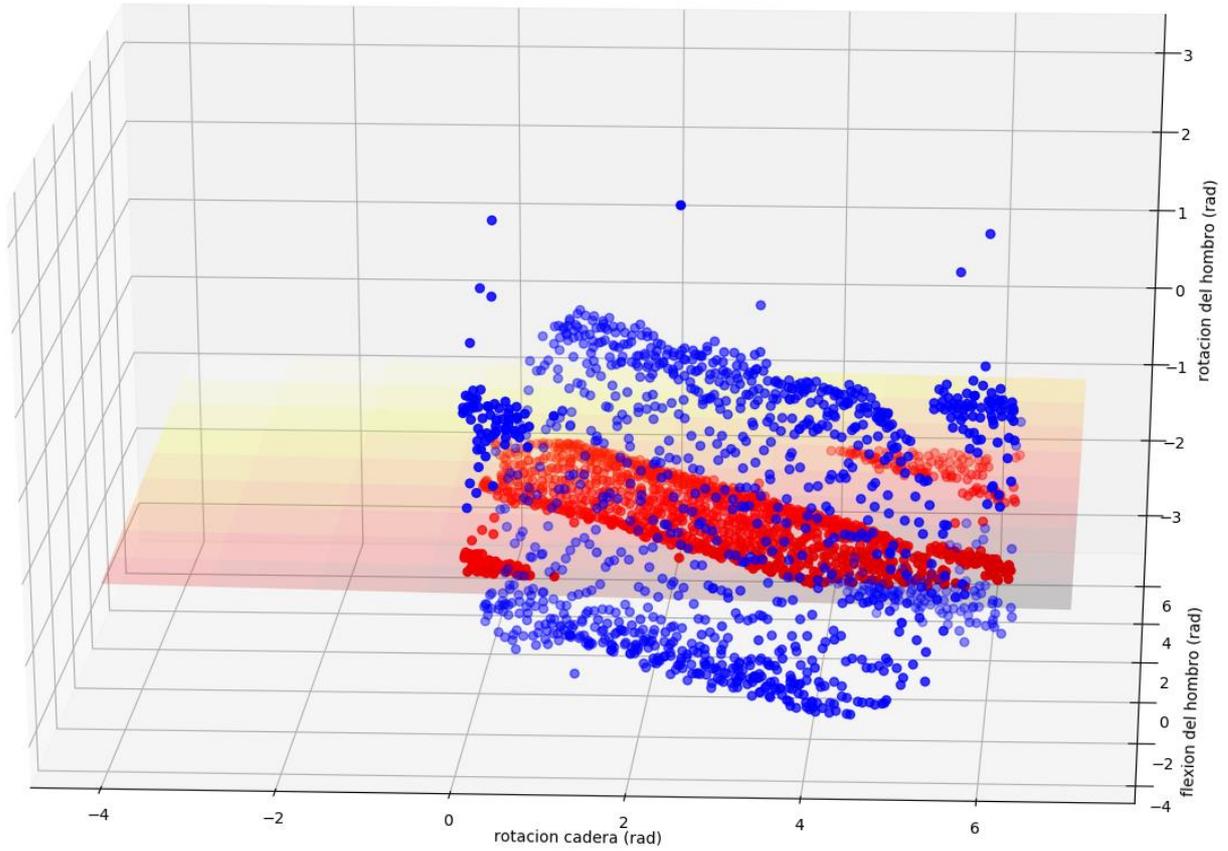
Gráfica 5.3-8. Evolución de los ángulos de los movimientos de las articulaciones a lo largo del tiempo de la salida 172. Aclaración. Las columnas son de derecha a izquierda la cadera, el hombro derecho e izquierdo, relativamente, y el codo derecho e izquierdo, relativamente. Y en el título está el nombre de cada articulación.

En la Gráfica 5.3-7 y la Gráfica 5.3-3 que muestran los ángulos de la salida 171 y 167, respectivamente. Se ve como la flexión del codo del brazo derecho debería ser solo positivo y el izquierdo negativo. Ya que este solo puede ser doblado en una única dirección. Sin embargo, se observa que, en el codo izquierdo del 167, no es así. Y la flexión del hombro del brazo derecho debería ser positivo y el izquierdo negativo. Sin embargo, se ve que esto no se cumple en el hombro derecho de la salida 171. Y tampoco, se cumple en el hombro izquierdo del brazo de la salida 167. A continuación, se verá los resultados de comparar los ángulos de las articulaciones de estas dos salidas. Estas comparaciones son tal y como se ha dicho en el apartado 4.4.1.3.

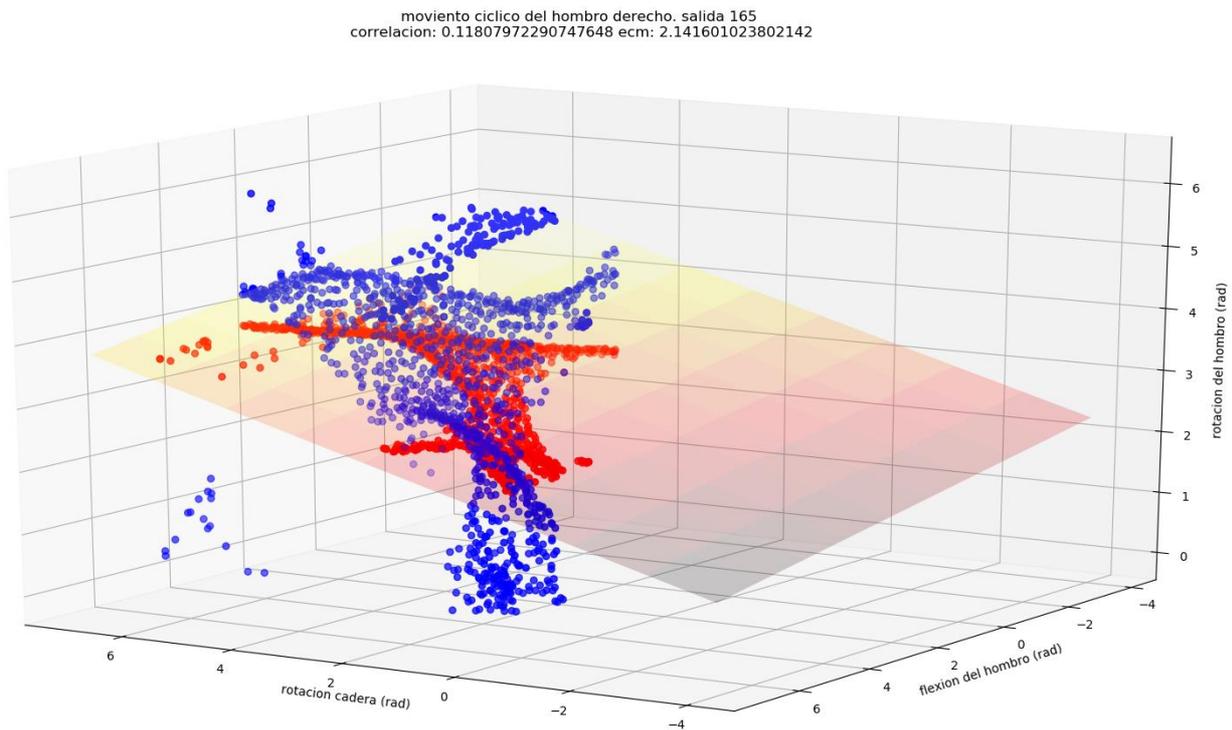


Gráfica 5.3-9. Comparación de las entradas de la rotación del hombro derecho con respecto a la flexión del hombro derecho y la rotación de la cadera de la salida 164. Y plano del modelo lineal de las entradas.

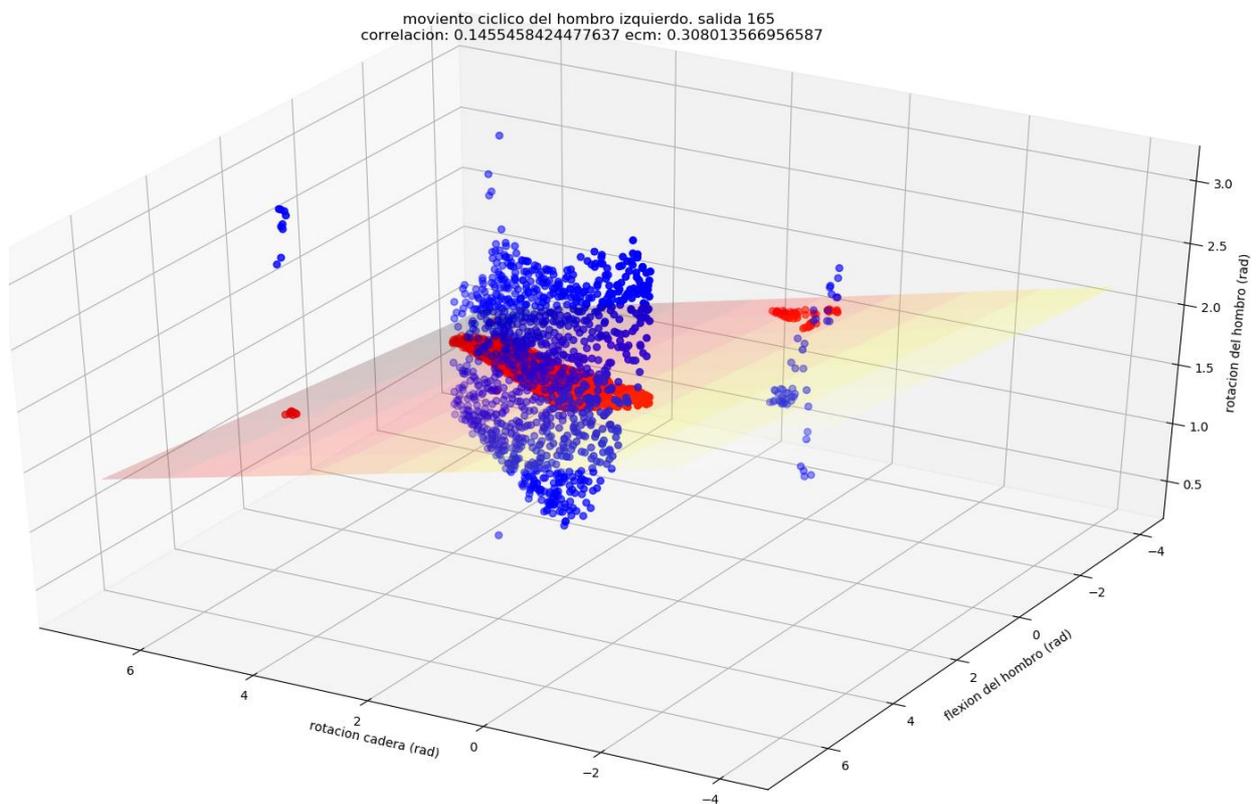
movimiento ciclico del hombro izquierdo. salida 164
correlacion: 0.0021669845504203034 ecm: 1.875309291787687



Gráfica 5.3-10. Comparación de las entradas de la rotación del hombro izquierdo con respecto a la flexión del hombro izquierdo y la rotación de la cadera de la salida 164. Y plano del modelo lineal de las entradas.

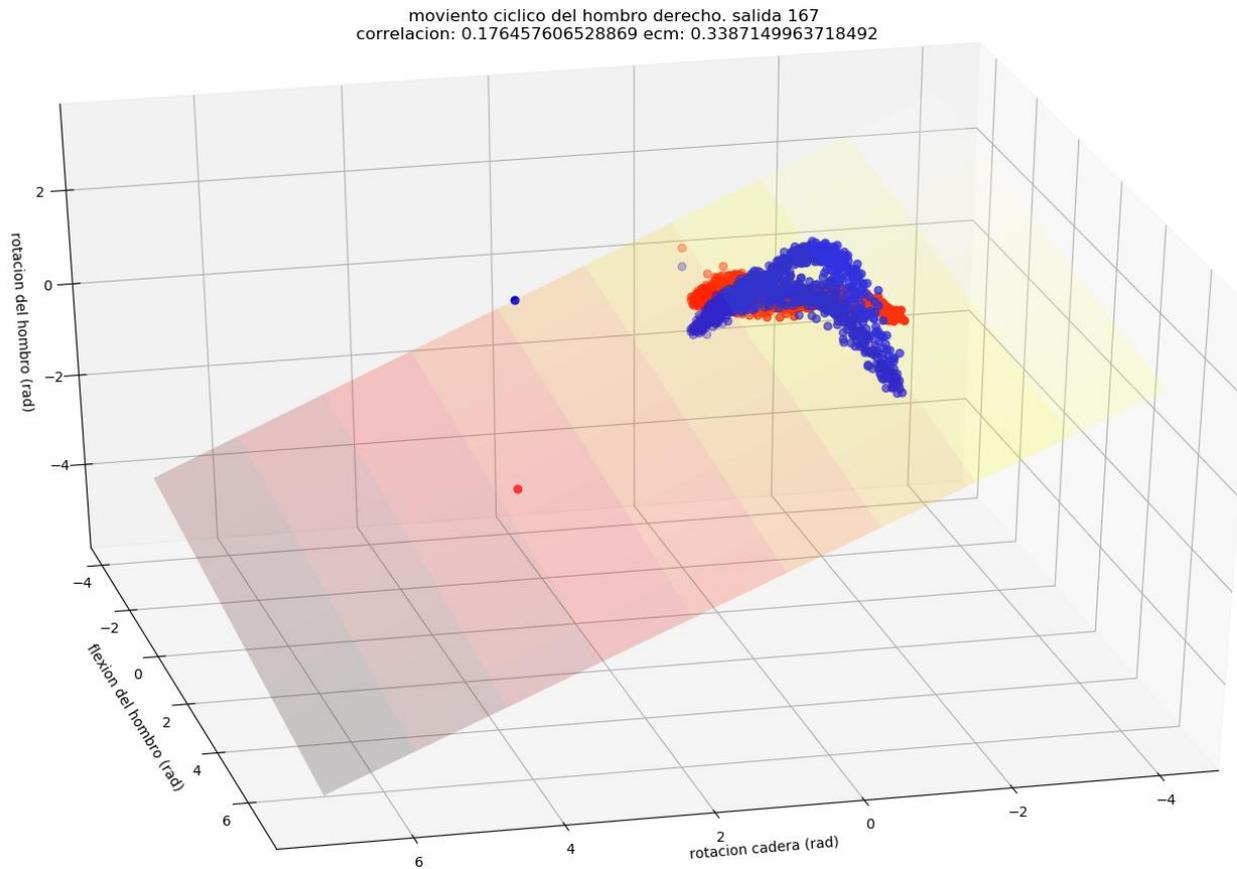


Gráfica 5.3-11. Comparación de las entradas de la rotación del hombro derecho con respecto a la flexión del hombro derecho y la rotación de la cadera de la salida 165. Y plano del modelo lineal de las entradas.



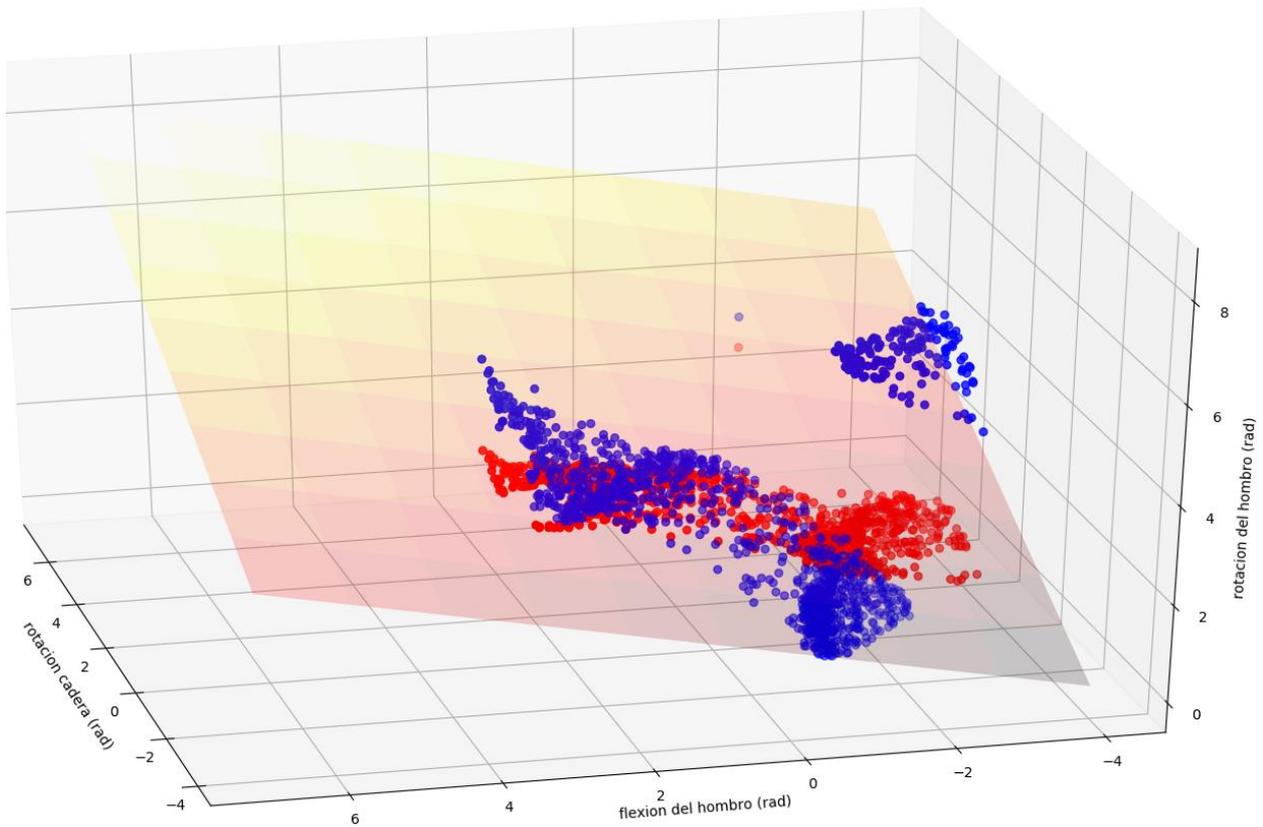
Gráfica 5.3-12. Comparación de las entradas de la rotación del hombro izquierdo con respecto a la flexión del

hombro izquierdo y la rotación de la cadera de la salida 165. Y plano del modelo lineal de las entradas.



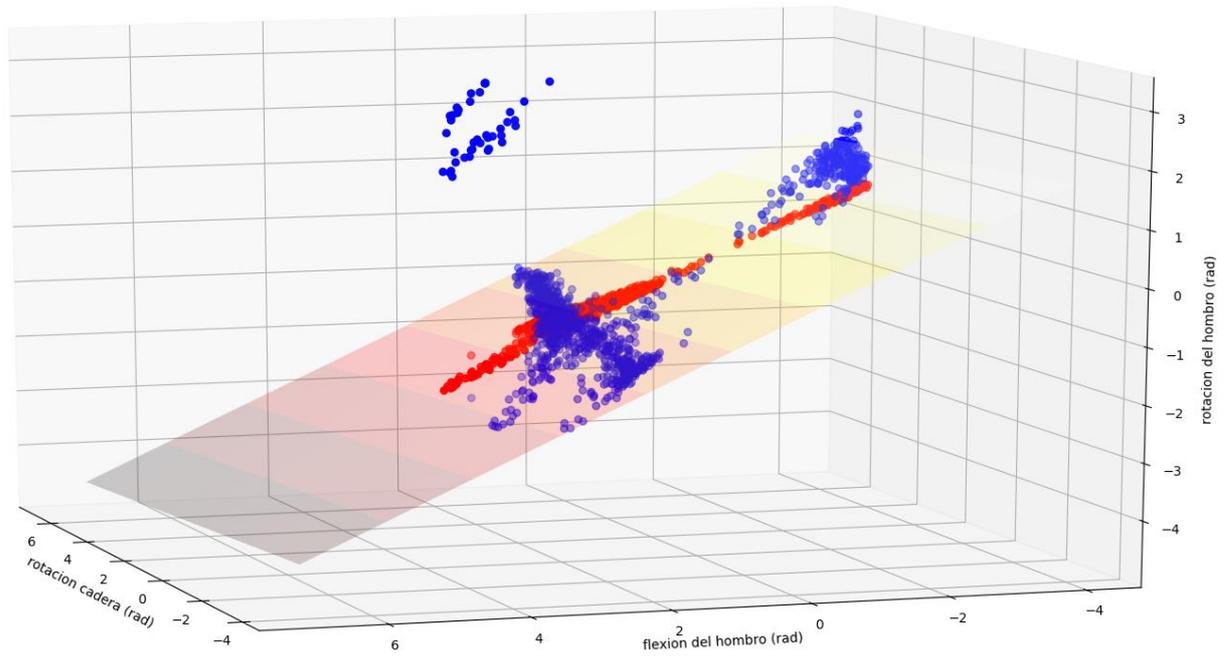
Gráfica 5.3-13. Comparación de las entradas de la rotación del hombro derecho con respecto a la flexión del hombro derecho y la rotación de la cadera de la salida 167. Y plano del modelo lineal de las entradas.

movimiento cíclico del hombro izquierdo, salida 167
correlacion: 0.10688000478680437 ecm: 3.0462026271313176

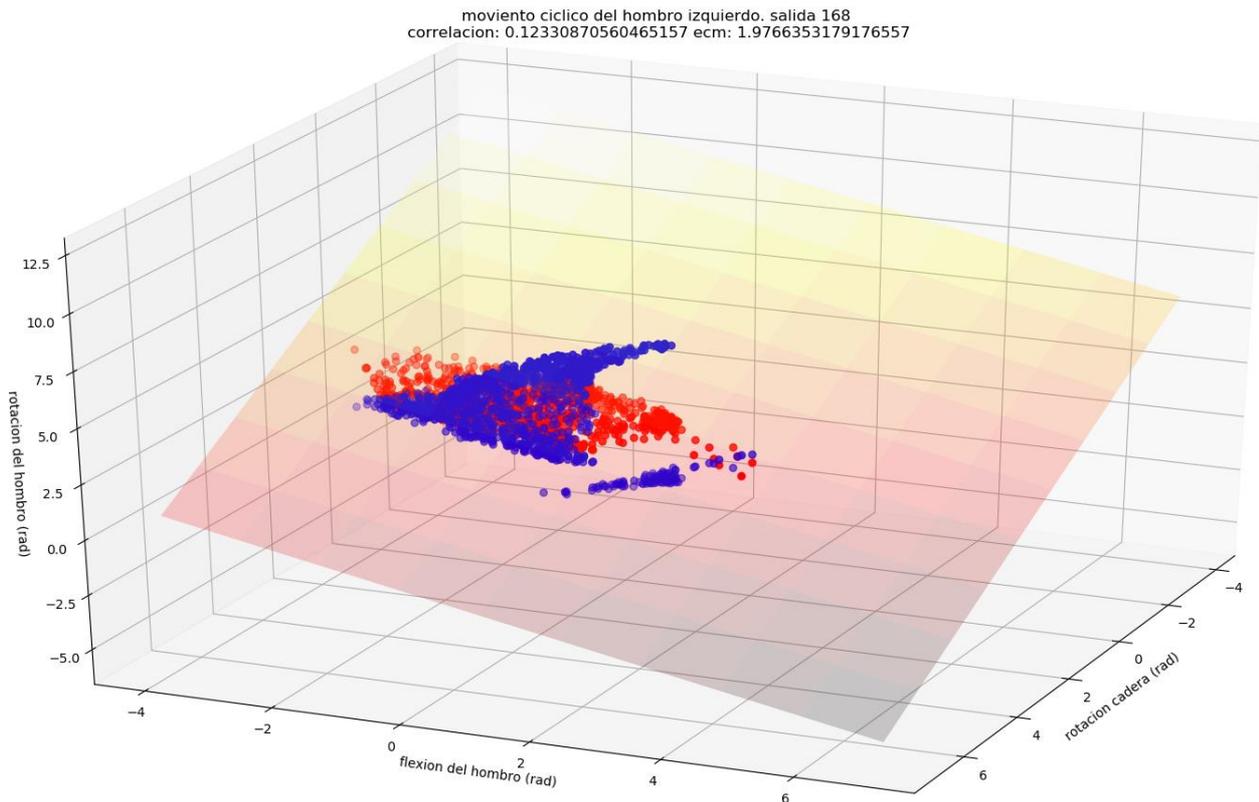


Gráfica 5.3-14. Comparación de las entradas de la rotación del hombro izquierdo con respecto a la flexión del hombro izquierdo y la rotación de la cadera de la salida 167. Y plano del modelo lineal de las entradas.

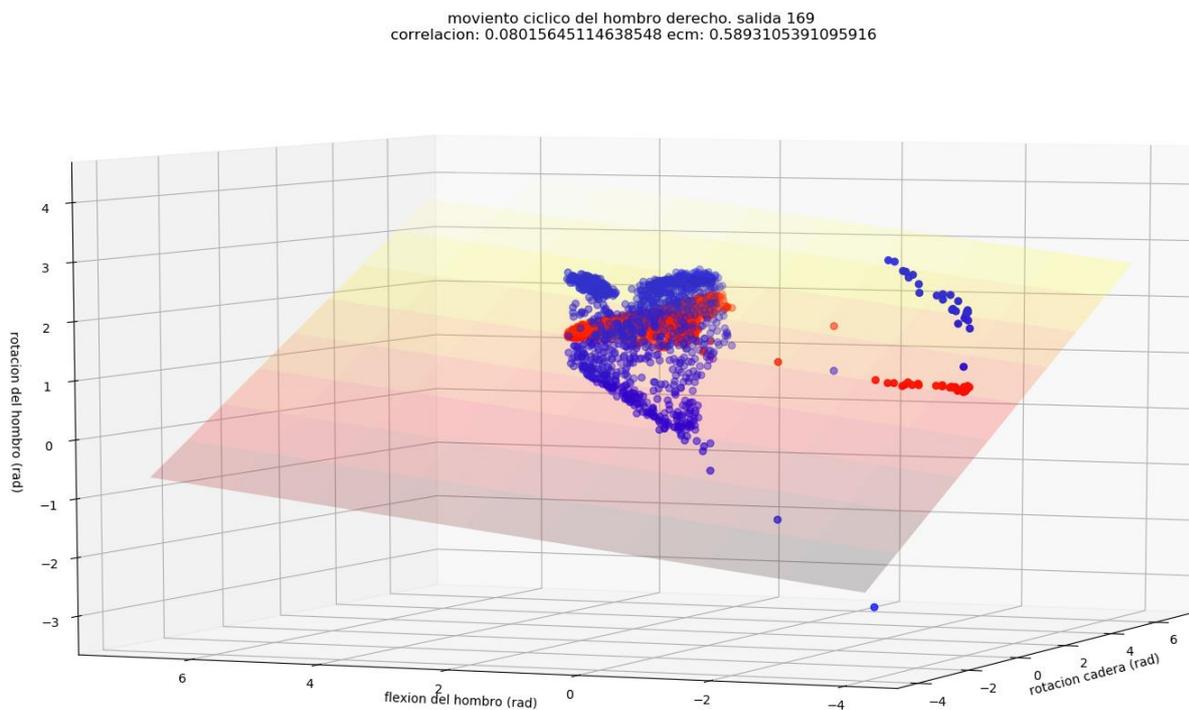
movimiento ciclico del hombro derecho. salida 168
correlacion: 0.3877949433982465 ecm: 1.12888029139316



Gráfica 5.3-15. Comparación de las entradas de la rotación del hombro derecho con respecto a la flexión del hombro derecho y la rotación de la cadera de la salida 168. Y plano del modelo lineal de las entradas.

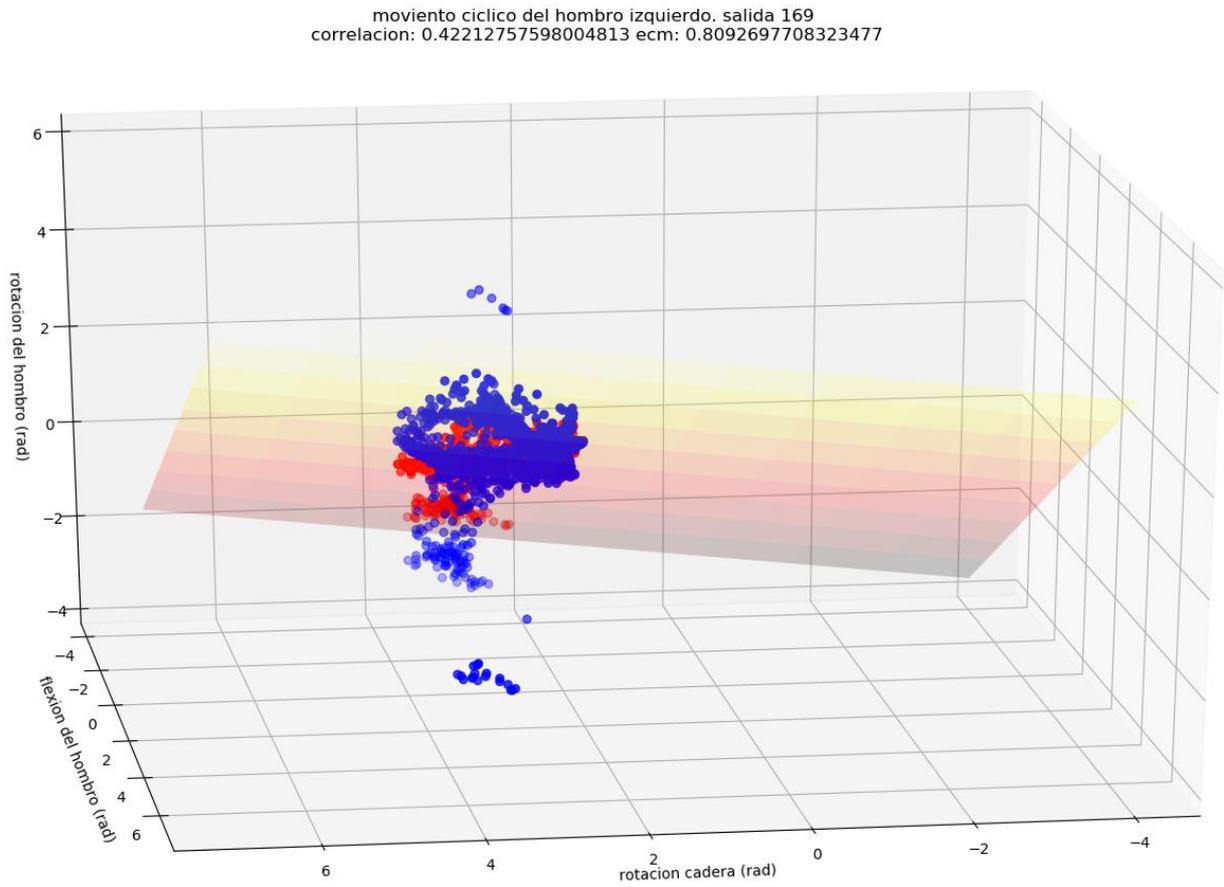


Gráfica 5.3-16. Comparación de las entradas de la rotación del hombro izquierdo con respecto a la flexión del hombro izquierdo y la rotación de la cadera de la salida 168. Y plano del modelo lineal de las entradas.

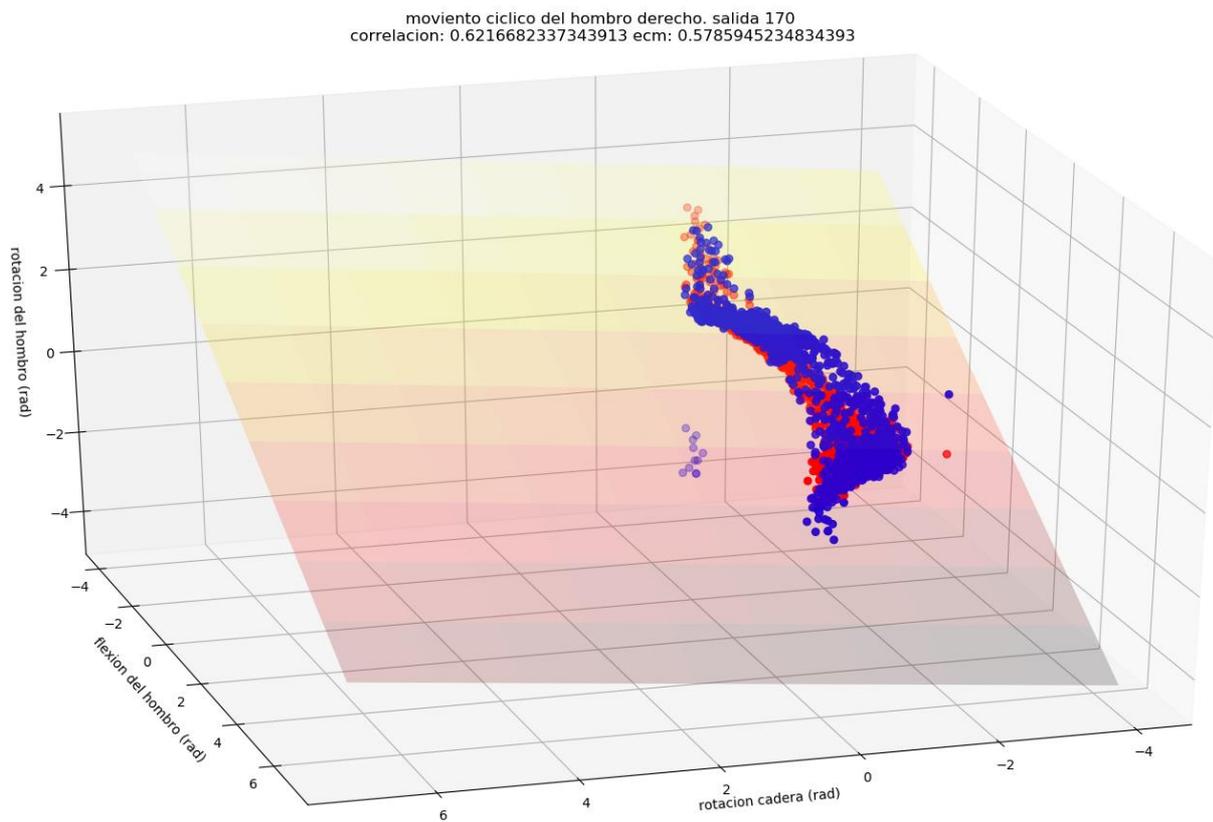


Gráfica 5.3-17. Comparación de las entradas de la rotación del hombro derecho con respecto a la flexión del

hombro derecho y la rotación de la cadera de la salida 169. Y plano del modelo lineal de las entradas.

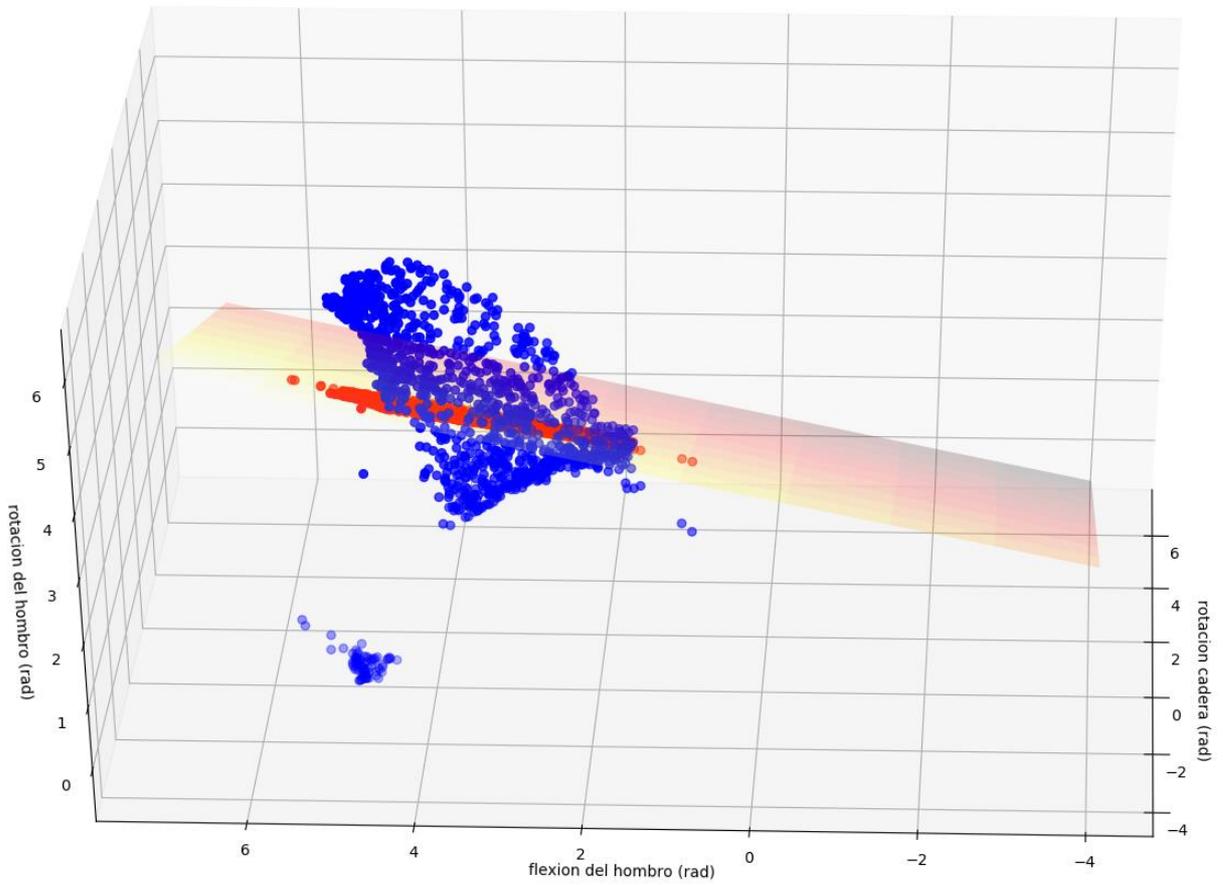


Gráfica 5.3-18. Comparación de las entradas de la rotación del hombro izquierdo con respecto a la flexión del hombro izquierdo y la rotación de la cadera de la salida 169. Y plano representa el modelo lineal de las entradas.



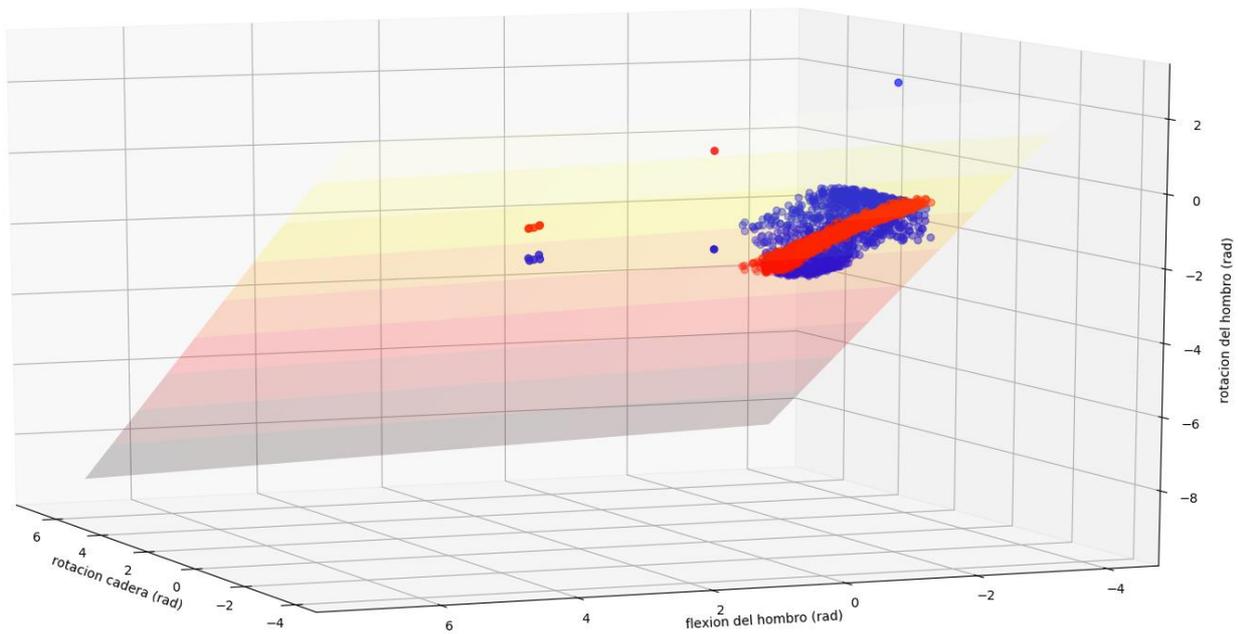
Gráfica 5.3-19. Comparación de las entradas de la rotación del hombro derecho con respecto a la flexión del hombro derecho y la rotación de la cadera de la salida 170. Y plano representa el modelo lineal de las entradas.

movimiento ciclico del hombro izquierdo. salida 170
correlacion: 0.06442671663881938 ecm: 1.9693431319101242



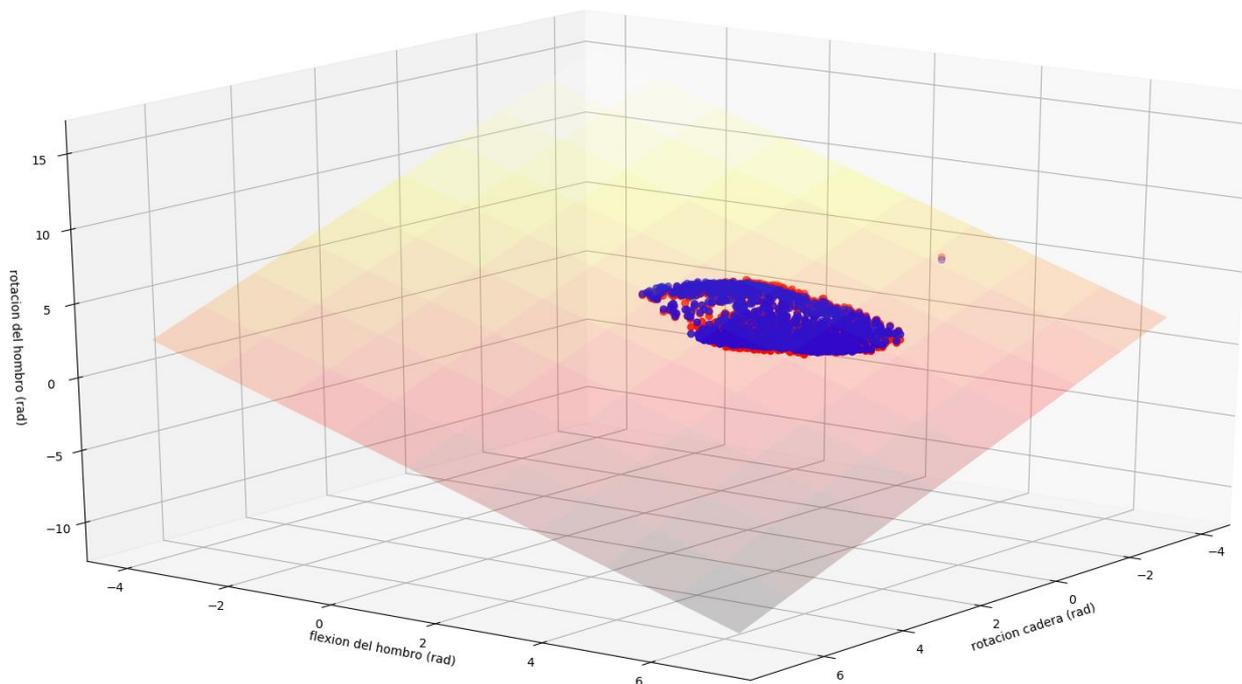
Gráfica 5.3-20. Comparación de las entradas de la rotación del hombro izquierdo con respecto a la flexión del hombro izquierdo y la rotación de la cadera de la salida 170. Y plano representa el modelo lineal de las entradas.

movimiento ciclico del hombro derecho. salida 171
 correlacion: 0.4858756902019552 ecm: 0.32588711058618897



Gráfica 5.3-21. Comparación de las entradas de la rotación del hombro derecho con respecto a la flexión del hombro derecho y la rotación de la cadera de la salida 171. Y plano representa el modelo lineal de las entradas.

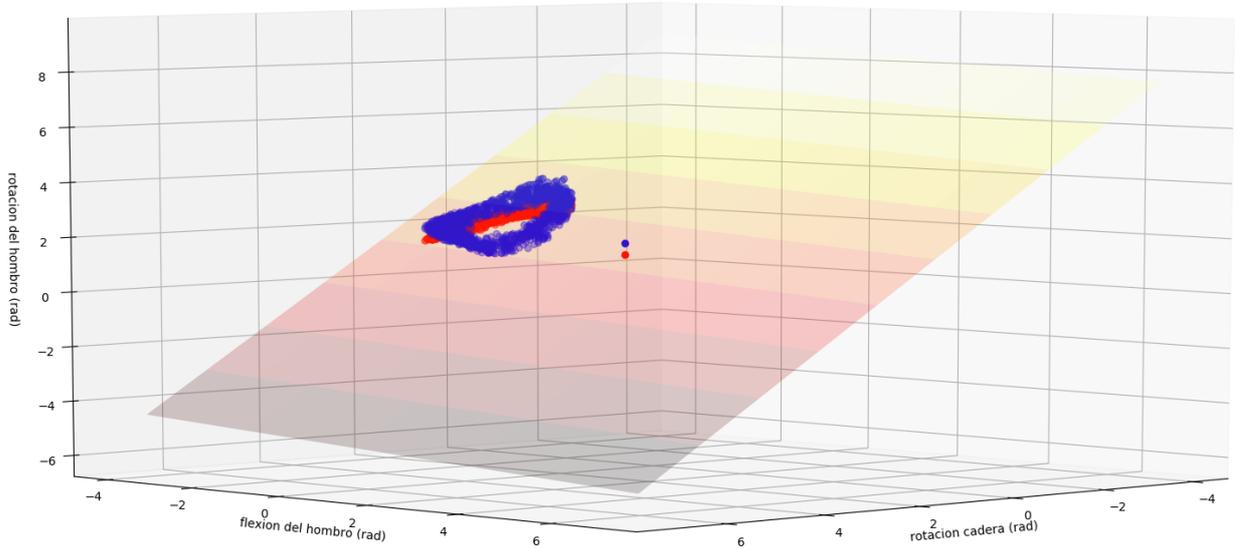
movimiento ciclico del hombro izquierdo. salida 171
 correlacion: 0.9687038015096667 ecm: 0.03723064942517965



Gráfica 5.3-22. Comparación de las entradas de la rotación del hombro izquierdo con respecto a la flexión del hombro izquierdo y la rotación de la cadera de la salida 171. Y plano representa el modelo lineal de las

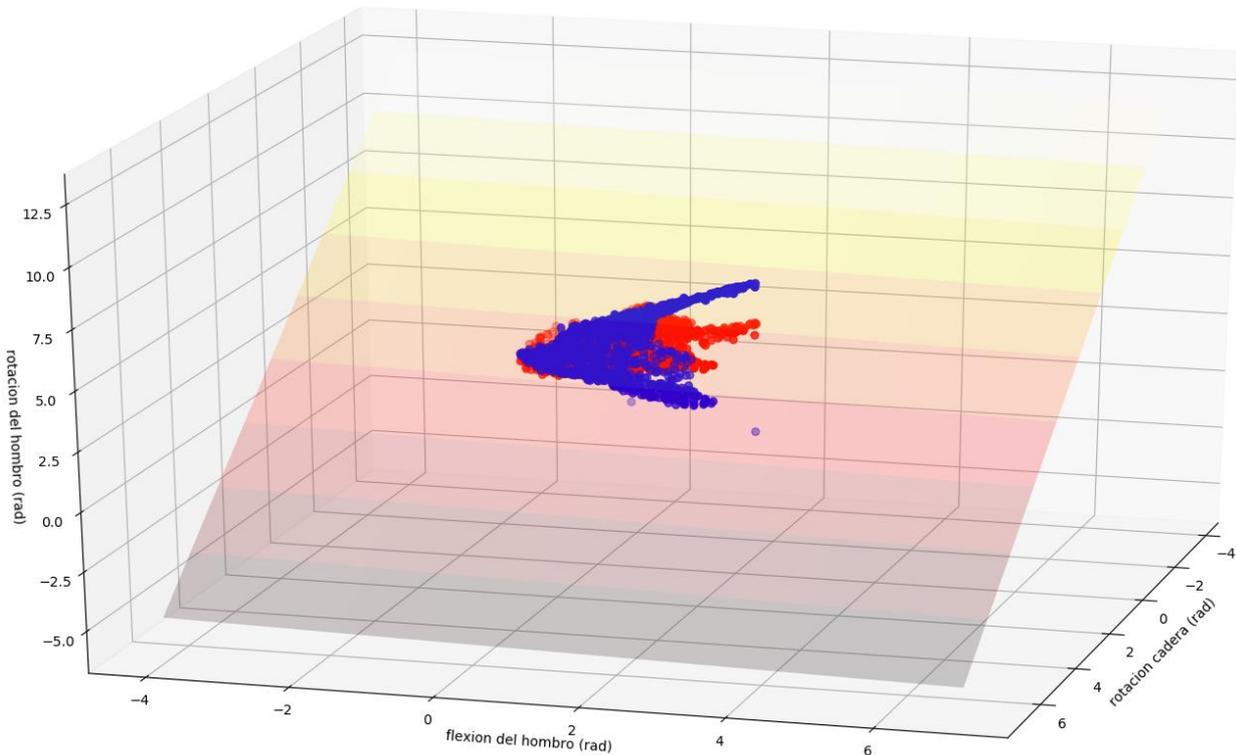
entradas.

moviento ciclico del hombro derecho. salida 172
correlacion: 0.3711007505856365 ecm: 0.30267985773740474



Gráfica 5.3-23. Comparación de las entradas de la rotación del hombro derecho con respecto a la flexión del hombro derecho y la rotación de la cadera de la salida 172. Y plano representa el modelo lineal de las entradas.

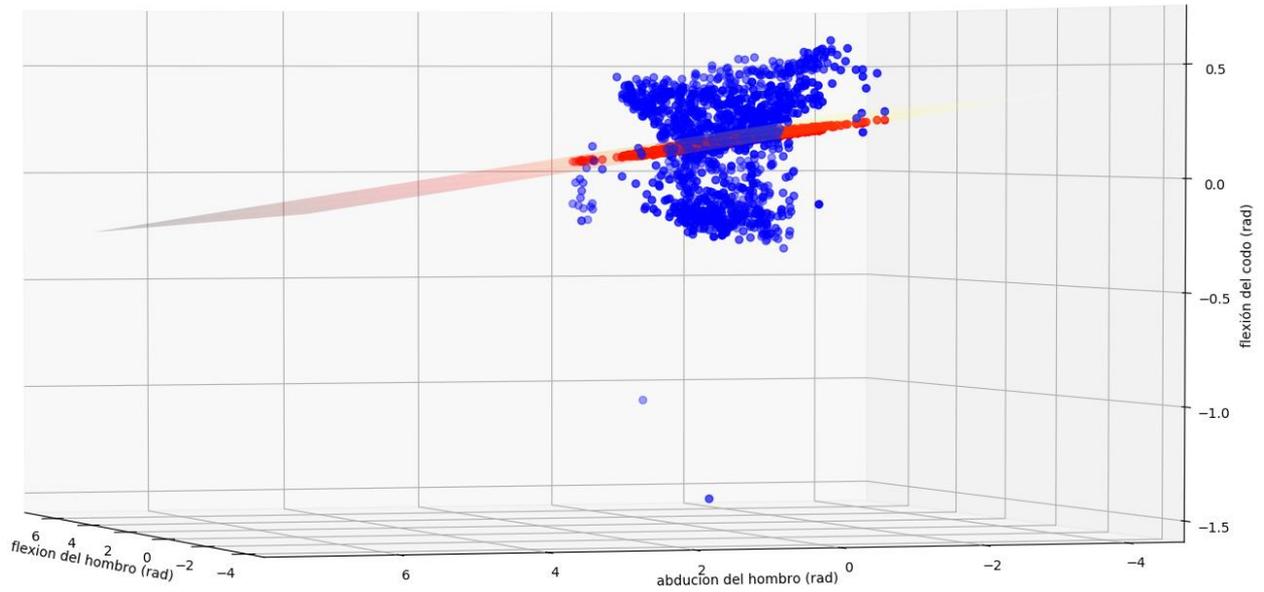
movimiento cíclico del hombro izquierdo. salida 172
 correlacion: 0.29318274905461383 ecm: 0.8872963737708316



Gráfica 5.3-24. Comparación de las entradas de la rotación del hombro izquierdo con respecto a la flexión del hombro izquierdo y la rotación de la cadera de la salida 172. Y plano representa el modelo lineal de las entradas.

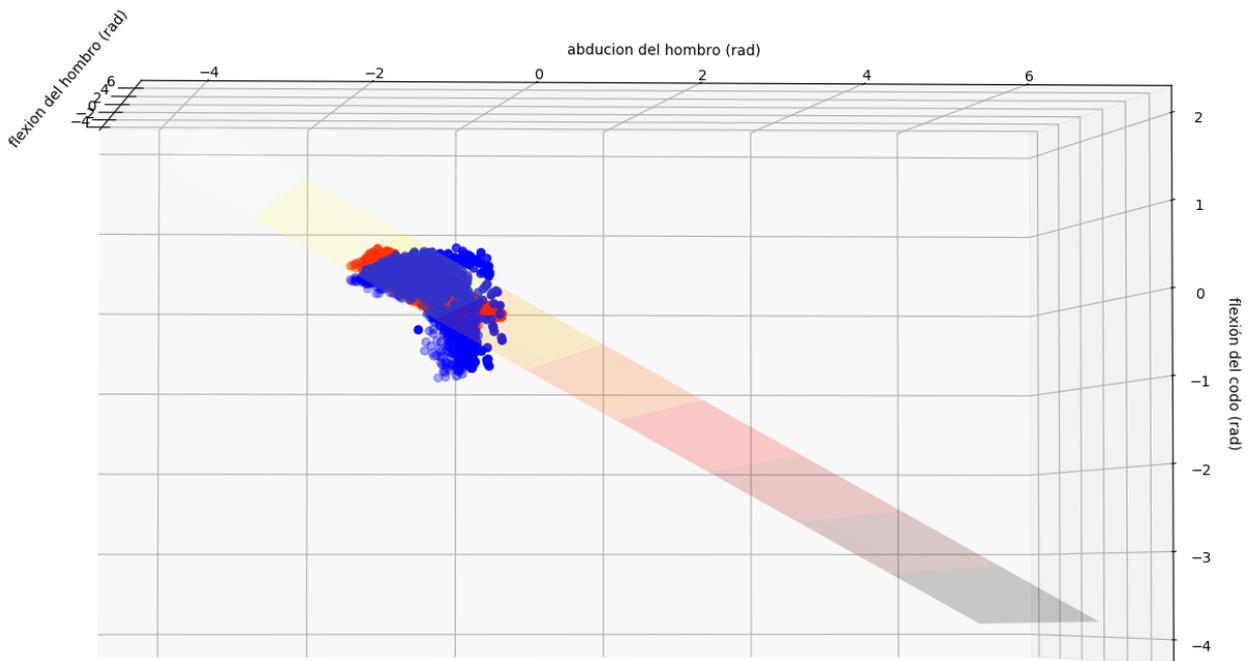
En la Gráfica 5.3-13 y en la Gráfica 5.3-22, se corresponde a los ángulos de los hombros de distinto brazos. Y cada gráfica está representado a un piragüista diferente. Sin embargo, en ambos casos se puede comprobar como la figura realizada por los puntos es parecida. Tiene una especie de figura parabólica cerrada. Se puede interpretar que una rama es el ataque y otra rama sería la rotación. La rama que cierra la parábola tiene menos puntos debido a que es más corta. Y no se habrá podido muestrear correctamente en todas las ocasiones. Esta figura tiene diferente inclinación en la rotación como se puede ver en ambas variables. Estas inclinaciones muestran lo rápido que hace la rotación del hombro en la fase de rotación. La amplitud de las dos ramas depende de la diferencia en el rango de rotación del hombro. Si vemos en la Gráfica 5.3-13 y Gráfica 5.3-22 se observa esta diferencia de rango. Si esta más paralelo a un eje que de otro da una pista de cómo se realiza la fase de rotación. E indica si este se realiza con la cadera o si hace más esfuerzo con el hombro. Esto se verá con más detalle en las gráficas de rendimiento de la rotación. Se puede ver como la Gráfica 5.3-22 encaja perfectamente con una regresión lineal. Sin embargo, en la Gráfica 5.3-13 la inteligencia artificial no ajusta bien los parámetros. Esto puede deberse a la inclinación de la figura resultante.

movimiento del codo derecho. salida 164
correlacion: 0.021720615336323412 ecm: 0.05351321529427134



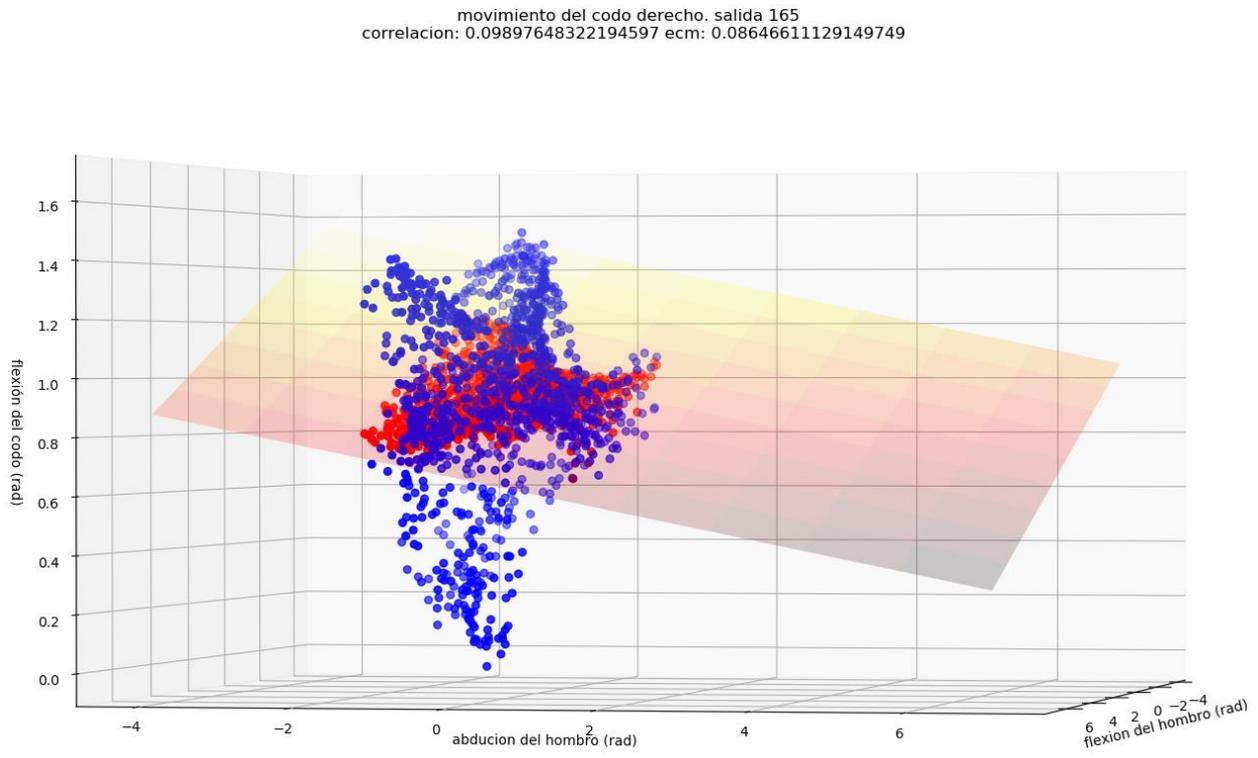
Gráfica 5.3-25. Comparación de las entradas de la flexión del codo derecho con respecto a la flexión y abducción del hombro derecho de la salida 164. Y plano representa el modelo lineal de las entradas.

movimiento del codo izquierdo. salida 164
correlacion: 0.3493092346235097 ecm: 0.07343617862546495



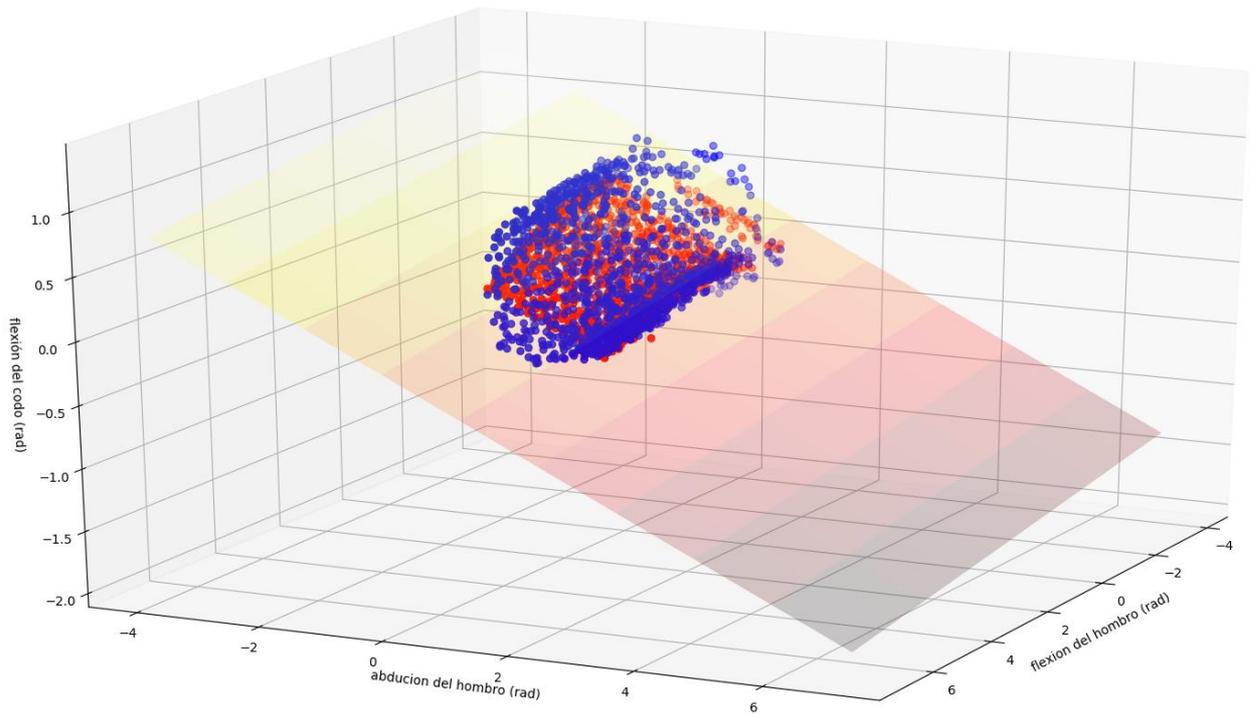
Gráfica 5.3-26. Comparación de las entradas de la flexión del codo izquierdo con respecto a la flexión y

abducción del hombro izquierdo de la salida 164. Y plano representa el modelo lineal de las entradas.



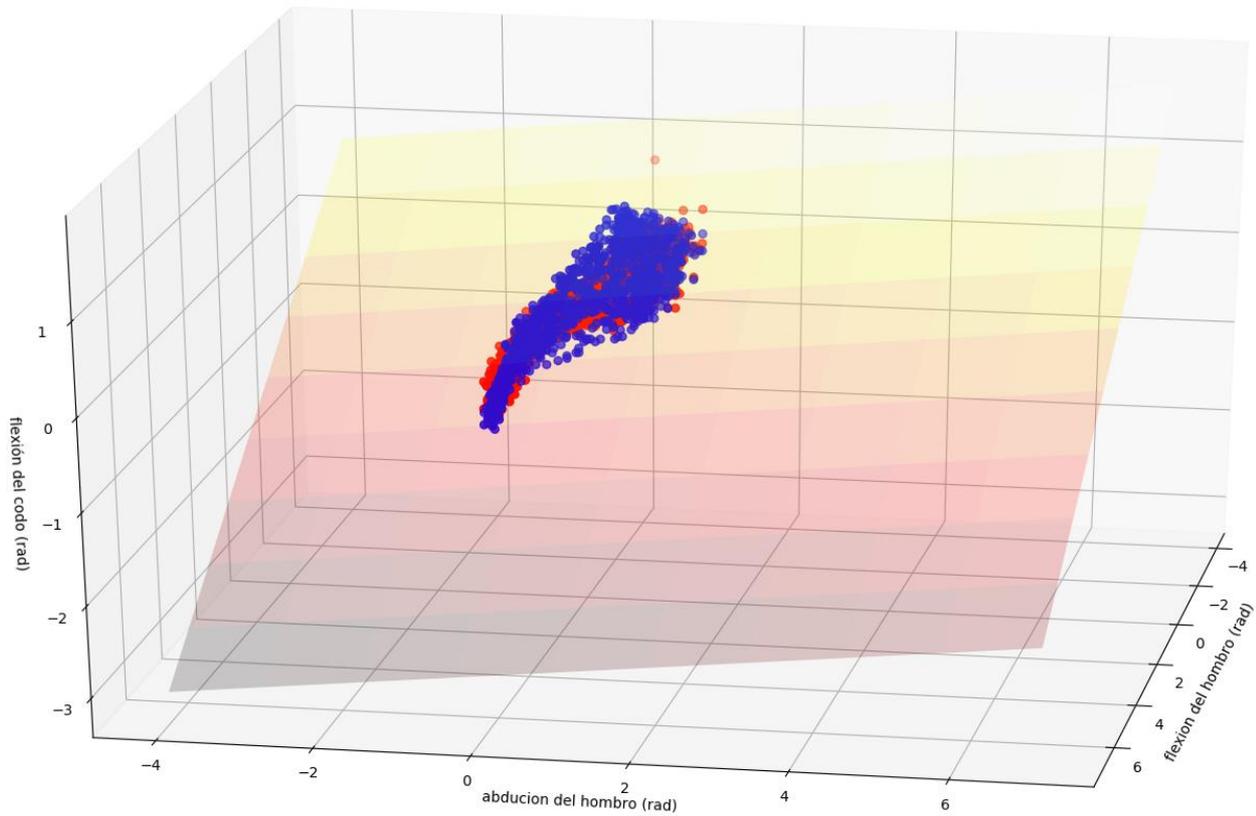
Gráfica 5.3-27. Comparación de las entradas de la flexión del codo derecho con respecto a la flexión y abducción del hombro derecho de la salida 165. Y plano representa el modelo lineal de las entradas.

movimiento del codo izquierdo. salida 165
correlacion: 0.35214588004375247 ecm: 0.07530515454935292



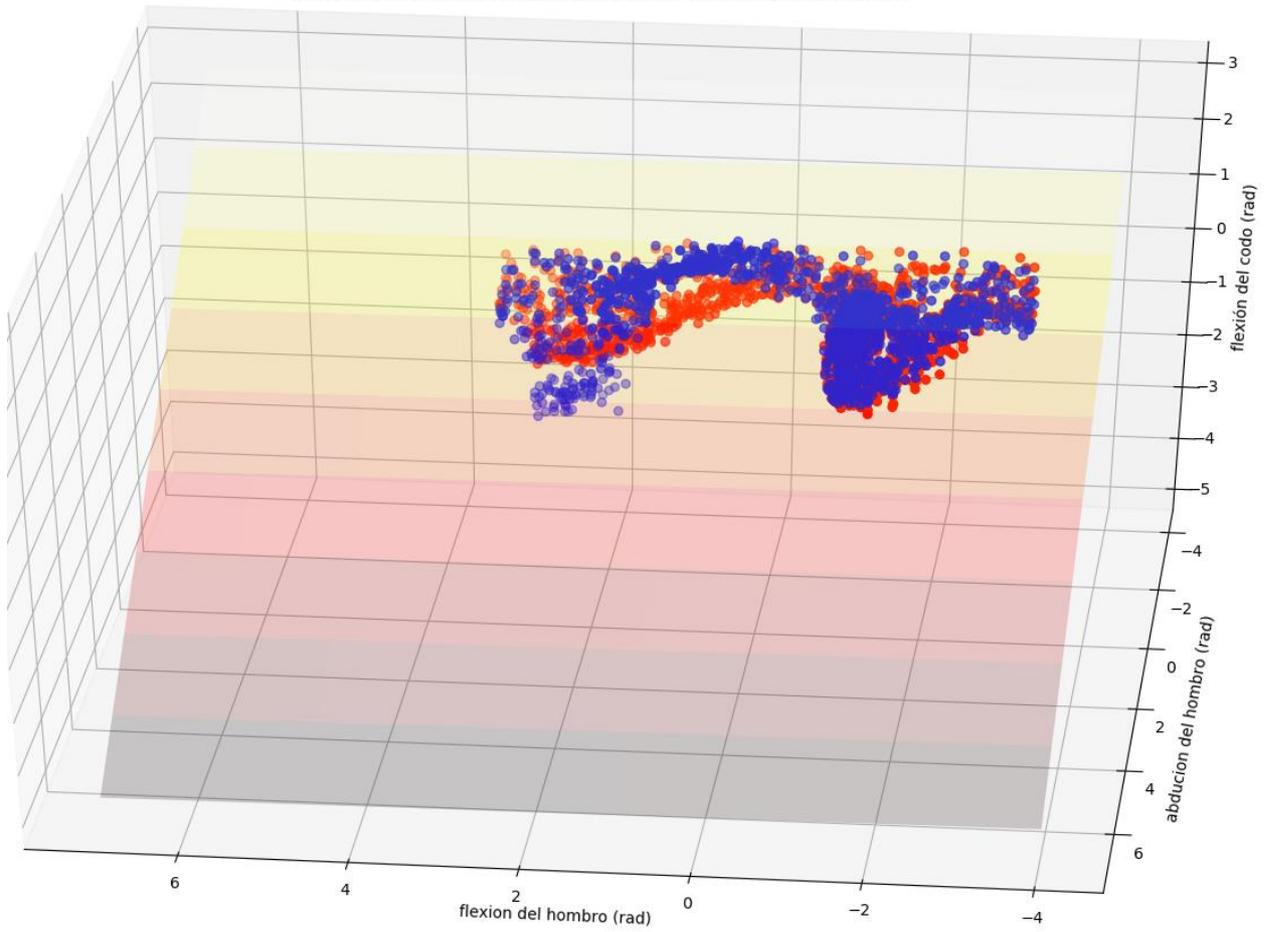
Gráfica 5.3-28. Comparación de las entradas de la flexión del codo izquierdo con respecto a la flexión y abducción del hombro izquierdo de la salida 165. Y plano representa el modelo lineal de las entradas.

movimiento del codo derecho. salida 167
correlacion: 0.7213714669867874 ecm: 0.0482993899967726



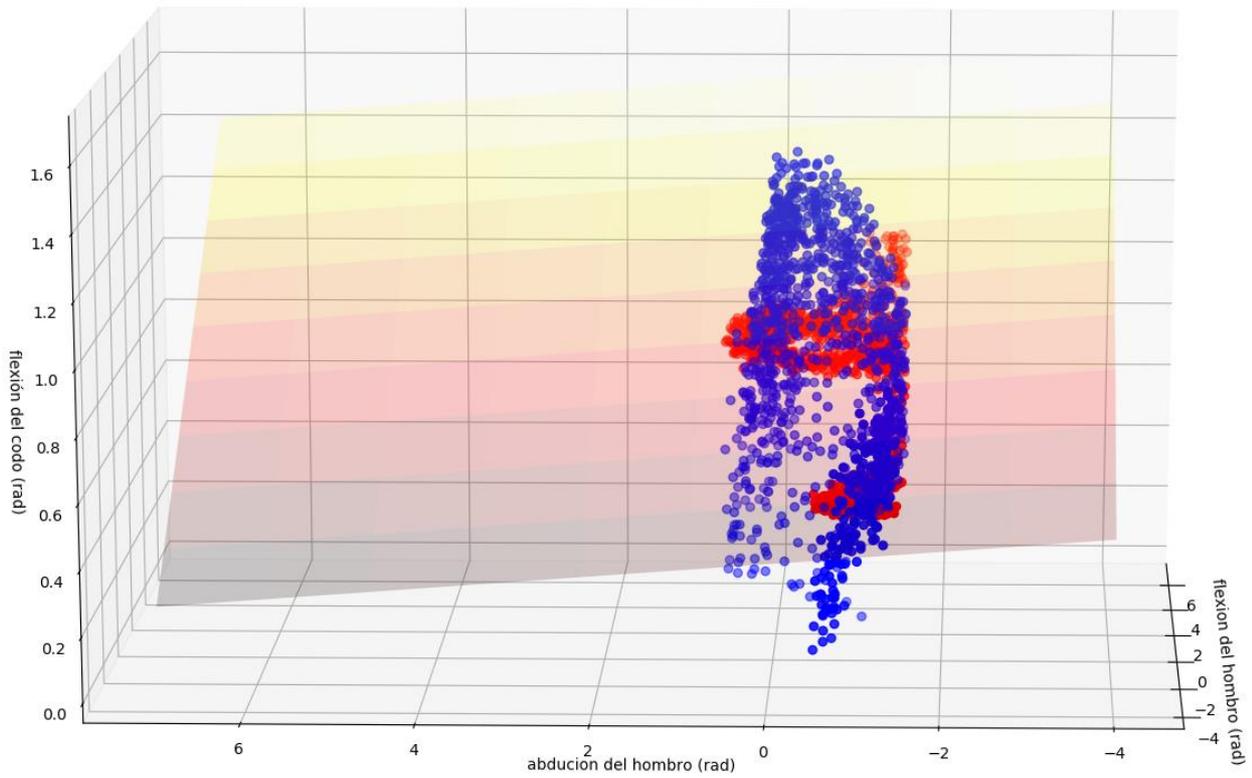
Gráfica 5.3-29. Comparación de las entradas de la flexión del codo derecho con respecto a la flexión y abducción del hombro derecho de la salida 167. Y plano representa el modelo lineal de las entradas.

movimiento del codo izquierdo. salida 167
correlacion: 0.43249097896620414 ecm: 0.1693727749842314



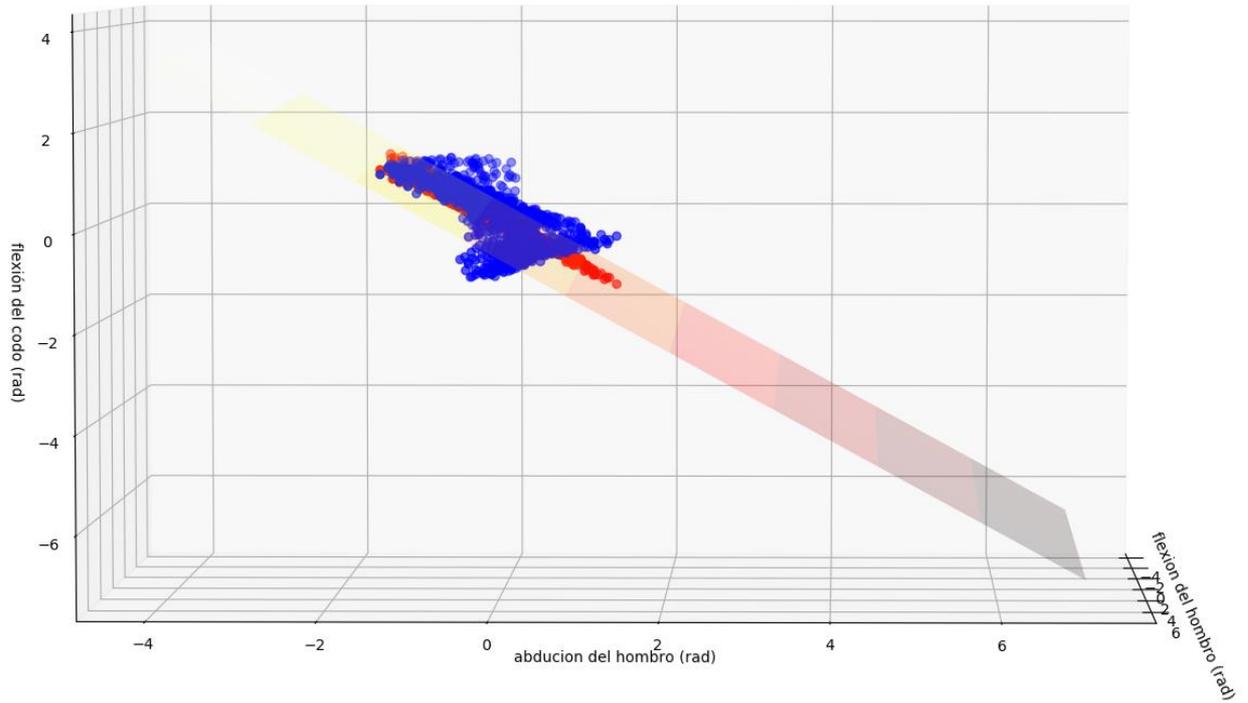
Gráfica 5.3-30. Comparación de las entradas de la flexión del codo izquierdo con respecto a la flexión y abducción del hombro izquierdo de la salida 167. Y plano representa el modelo lineal de las entradas.

movimiento del codo derecho. salida 168
correlacion: 0.23541525043783973 ecm: 0.07983354619471932



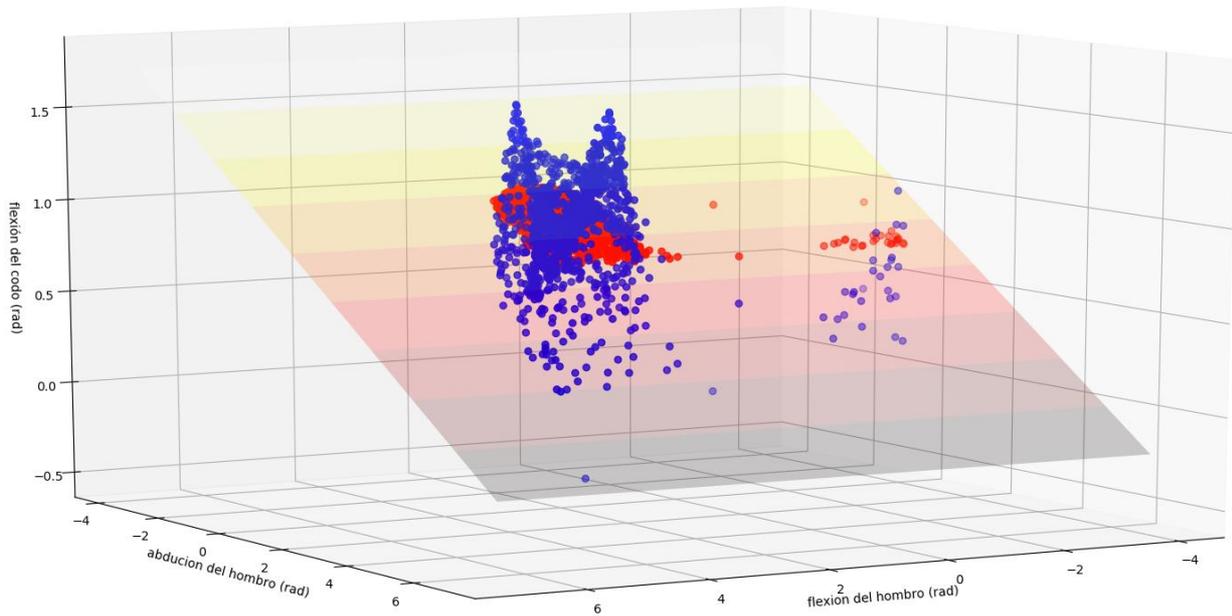
Gráfica 5.3-31. Comparación de las entradas de la flexión del codo derecho con respecto a la flexión y abducción del hombro derecho de la salida 168. Y plano representa el modelo lineal de las entradas.

movimiento del codo izquierdo. salida 168
correlacion: 0.5142262666790711 ecm: 0.20305492065206934



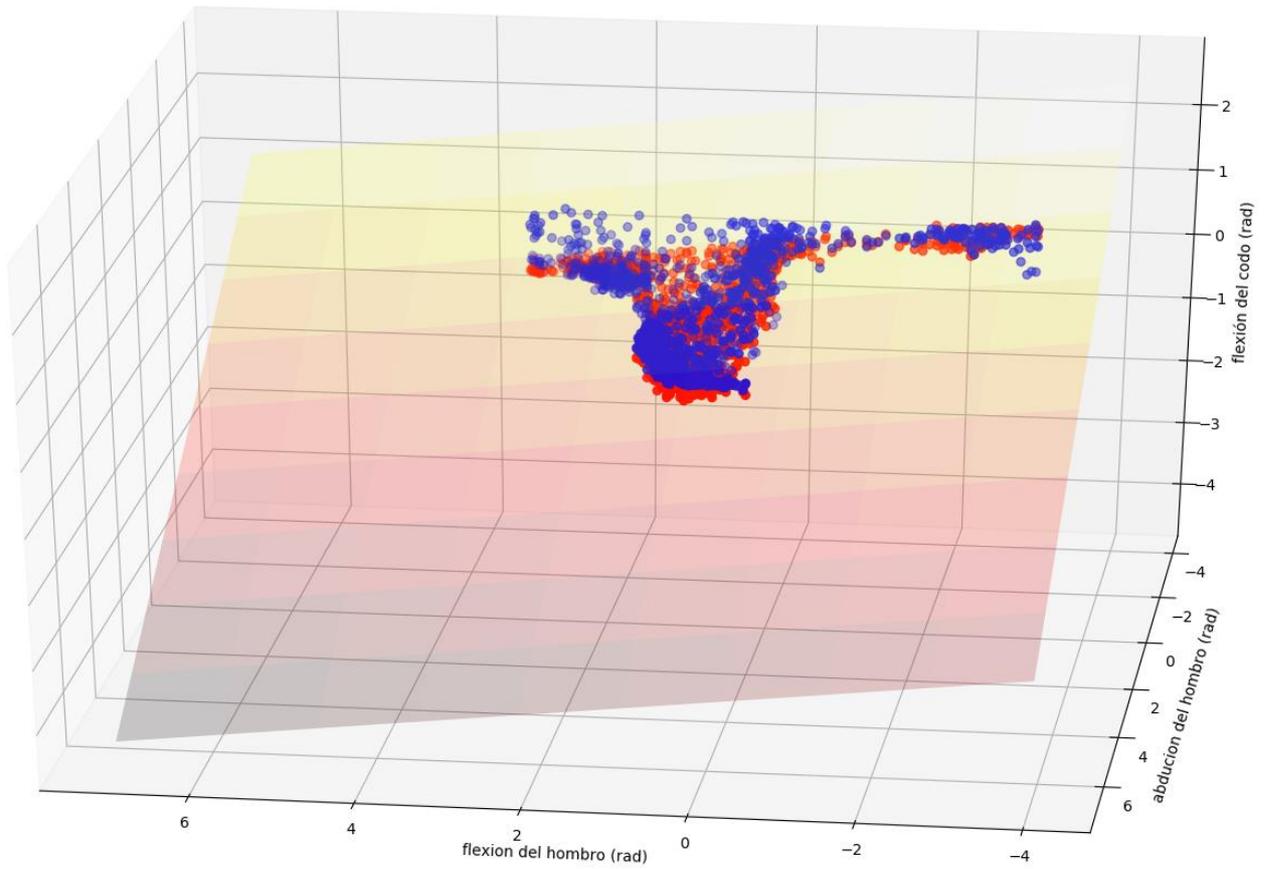
Gráfica 5.3-32. Comparación de las entradas de la flexión del codo izquierdo con respecto a la flexión y abducción del hombro izquierdo de la salida 168. Y plano representa el modelo lineal de las entradas.

movimiento del codo derecho, salida 169
correlacion: 0.11189234869187725 ecm: 0.061614513985309025

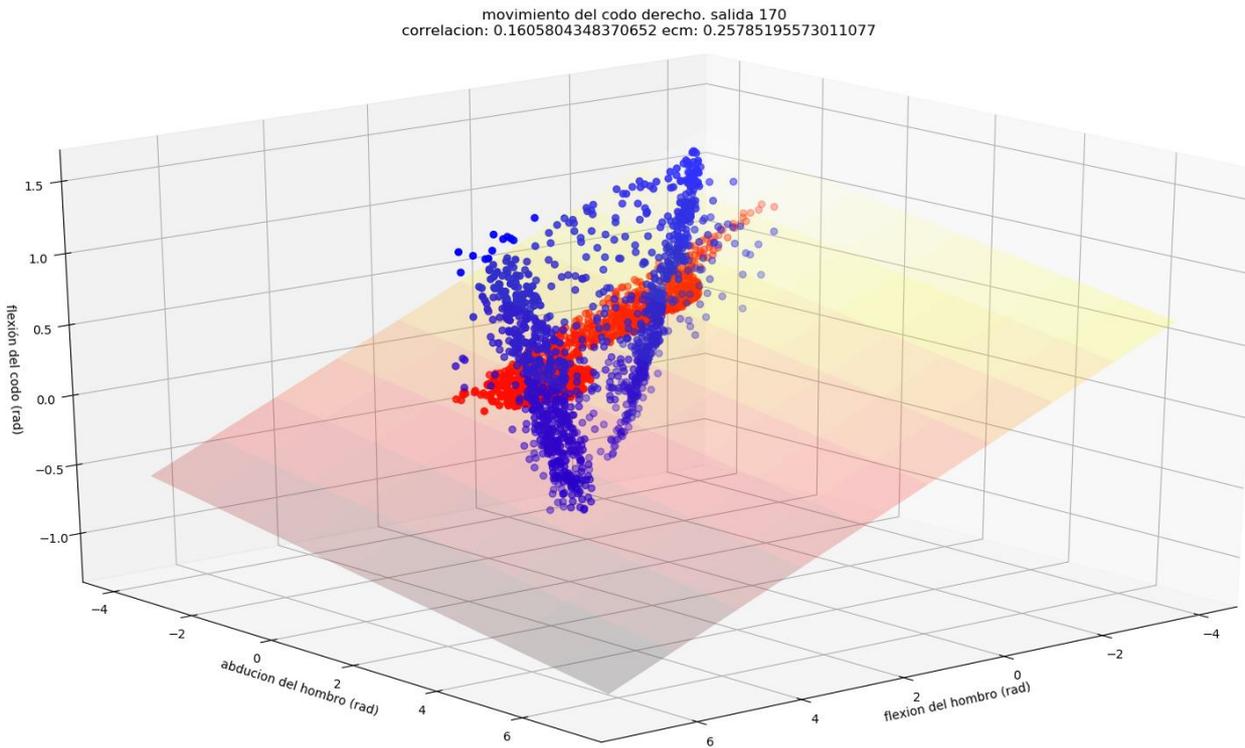


Gráfica 5.3-33. Comparación de las entradas de la flexión del codo derecho con respecto a la flexión y abducción del hombro derecho de la salida 169. Y plano representa el modelo lineal de las entradas.

movimiento del codo izquierdo, salida 169
correlacion: 0.74277053505727 ecm: 0.07382518046495896

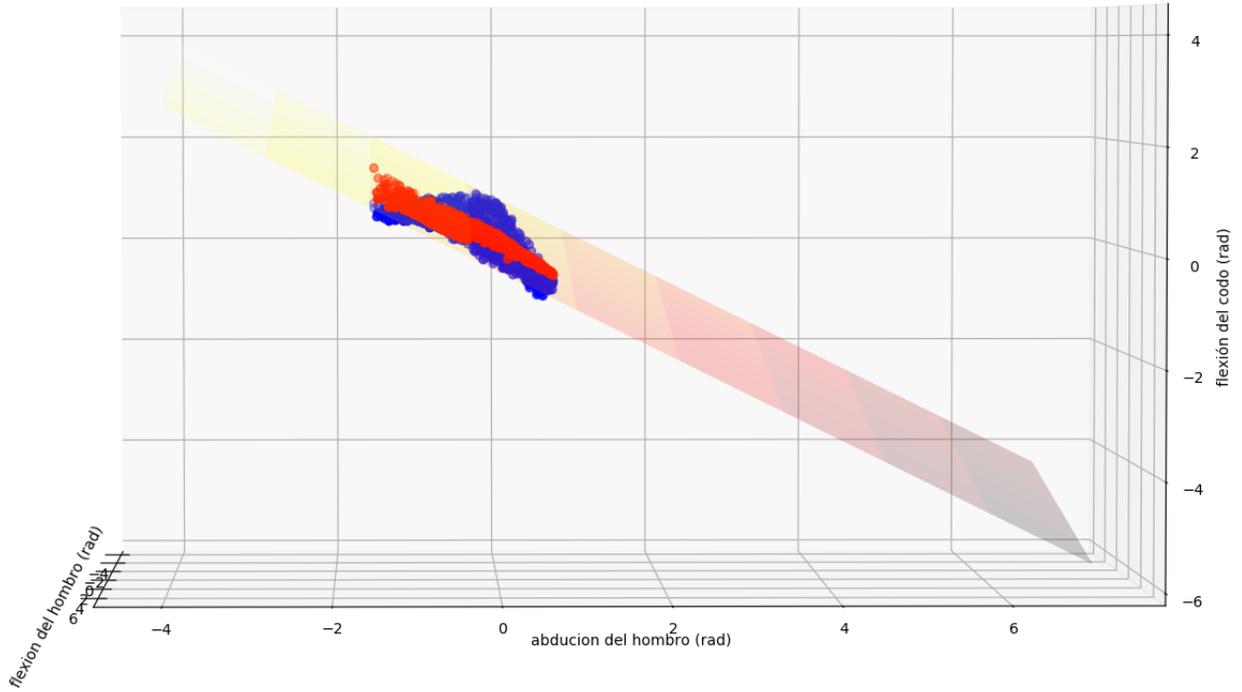


Gráfica 5.3-34. Comparación de las entradas de la flexión del codo izquierdo con respecto a la flexión y abducción del hombro izquierdo de la salida 169. Y plano representa el modelo lineal de las entradas.



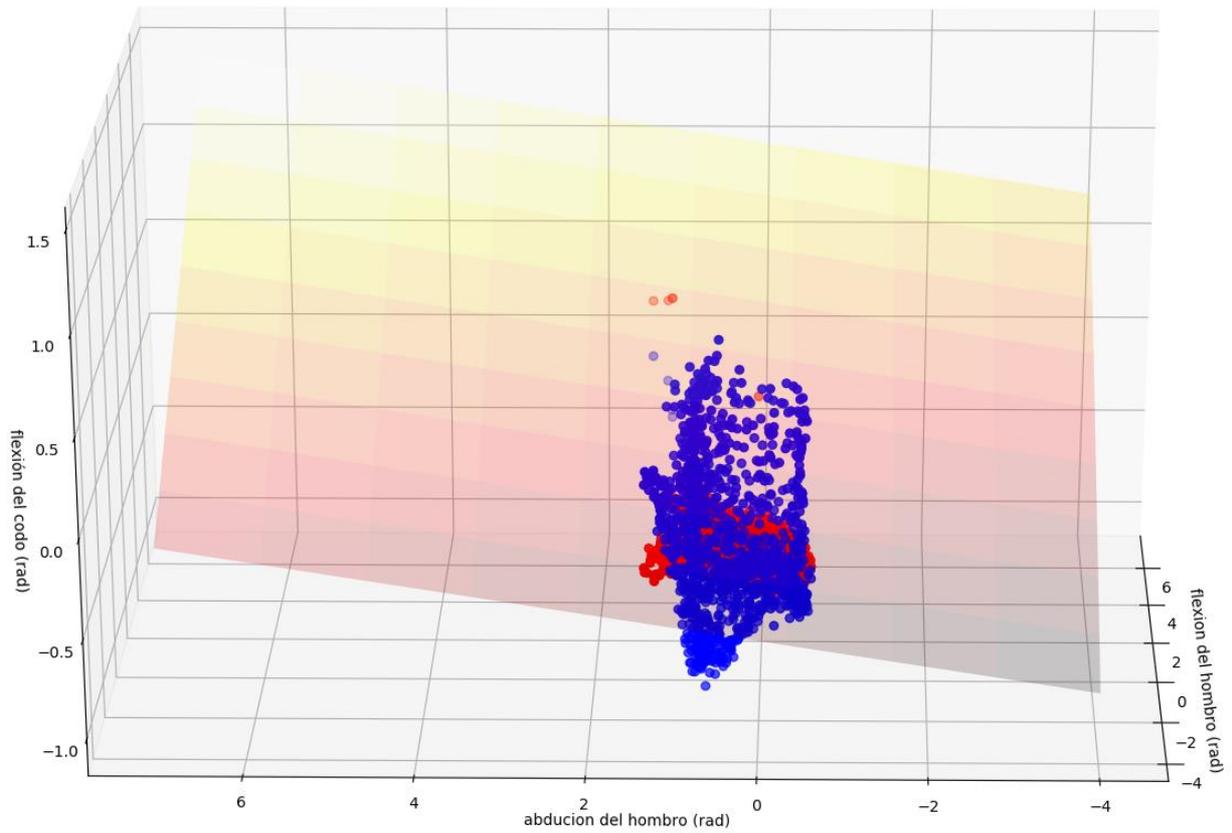
Gráfica 5.3-35. Comparación de las entradas de la flexión del codo derecho con respecto a la flexión y abducción del hombro derecho de la salida 170. Y plano representa el modelo lineal de las entradas.

movimiento del codo izquierdo. salida 170
correlacion: 0.6654862456430627 ecm: 0.05196993427724108



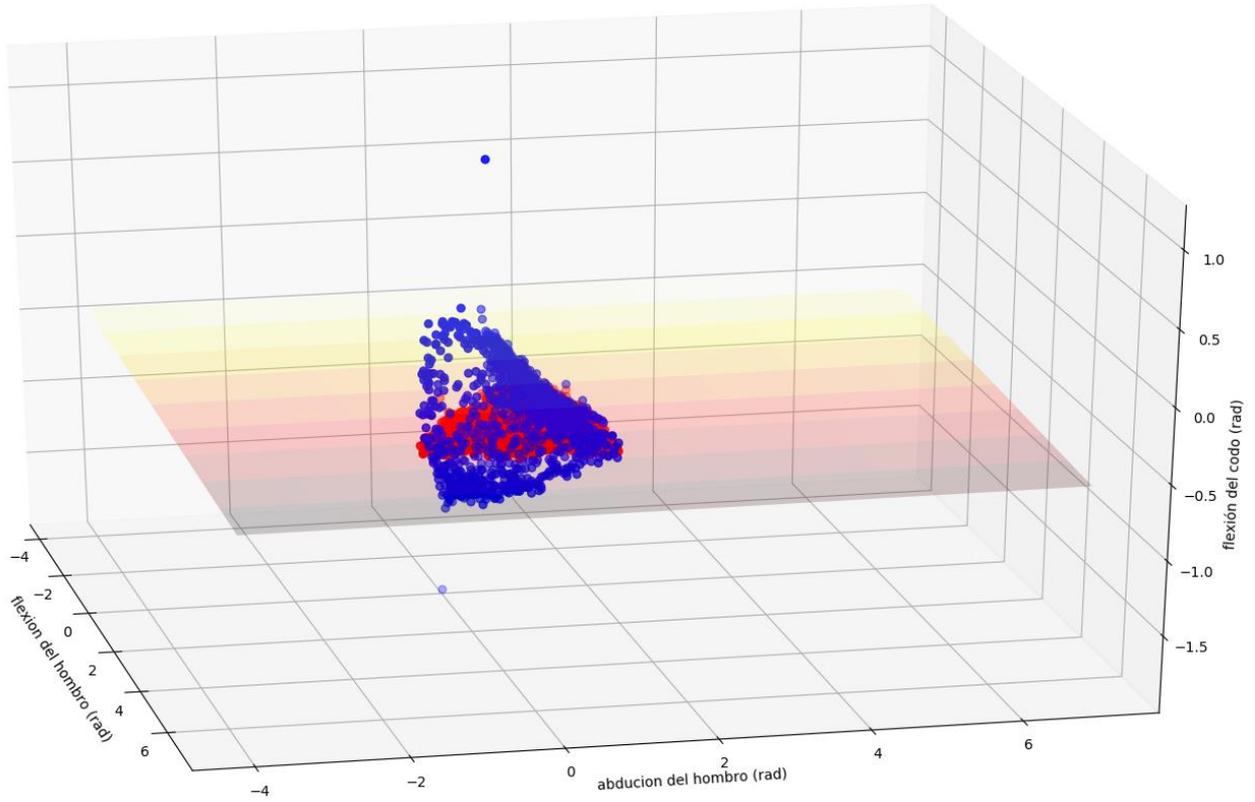
Gráfica 5.3-36. Comparación de las entradas de la flexión del codo izquierdo con respecto a la flexión y abducción del hombro izquierdo de la salida 170. Y plano representa el modelo lineal de las entradas.

movimiento del codo derecho. salida 171
correlacion: 0.04937942913852067 ecm: 0.12053855945881968



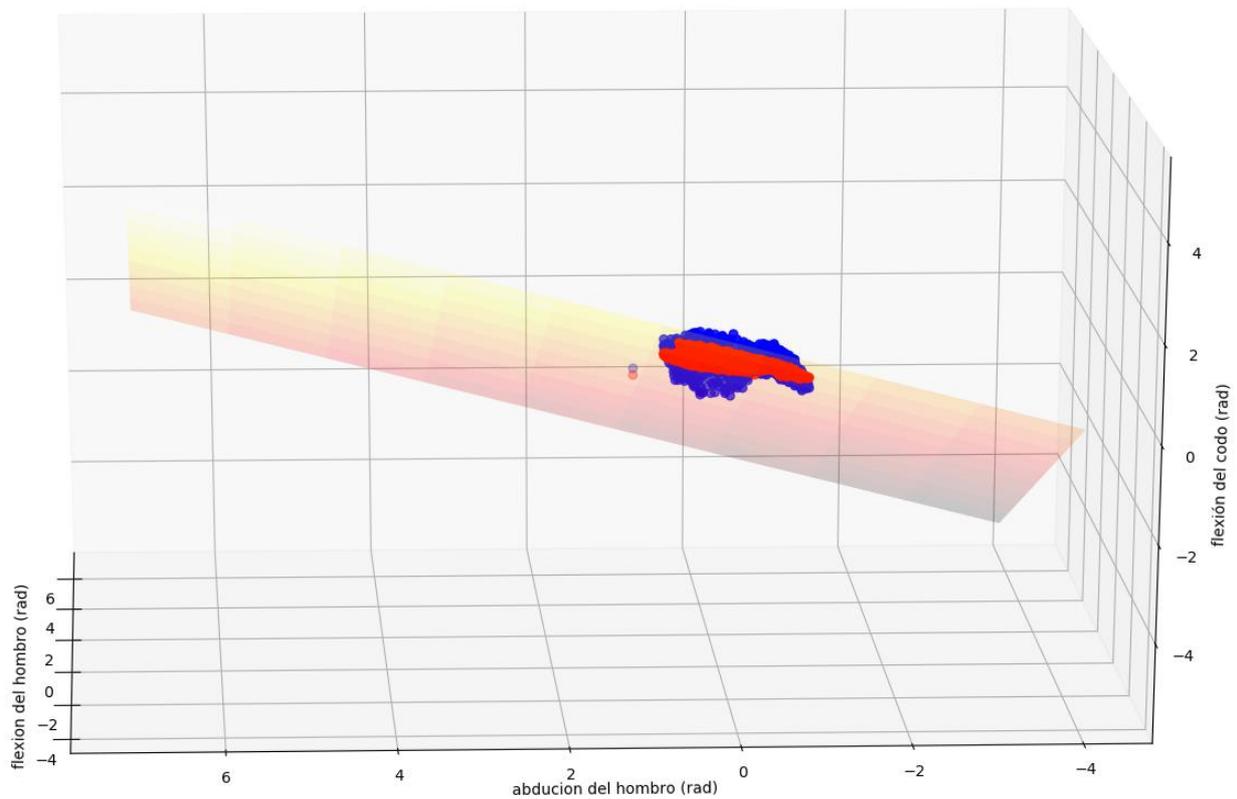
Gráfica 5.3-37. Comparación de las entradas de la flexión del codo derecho con respecto a la flexión y abducción del hombro derecho de la salida 171. Y plano representa el modelo lineal de las entradas.

movimiento del codo izquierdo. salida 171
correlacion: 0.0023949043354886124 ecm: 0.08250307371809851



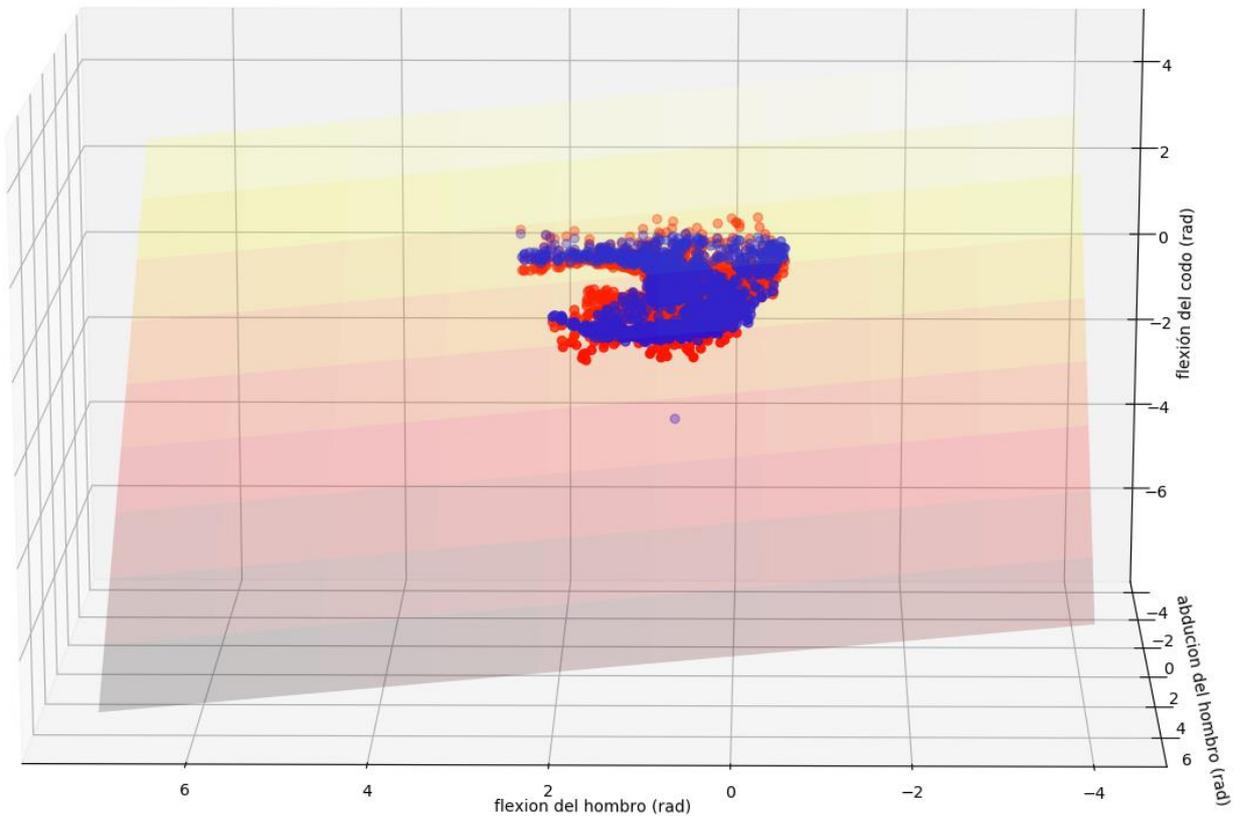
Gráfica 5.3-38. Comparación de las entradas de la flexión del codo izquierdo con respecto a la flexión y abducción del hombro izquierdo de la salida 171. Y plano representa el modelo lineal de las entradas.

movimiento del codo derecho. salida 172
correlacion: 0.565302024257621 ecm: 0.03805013740119623



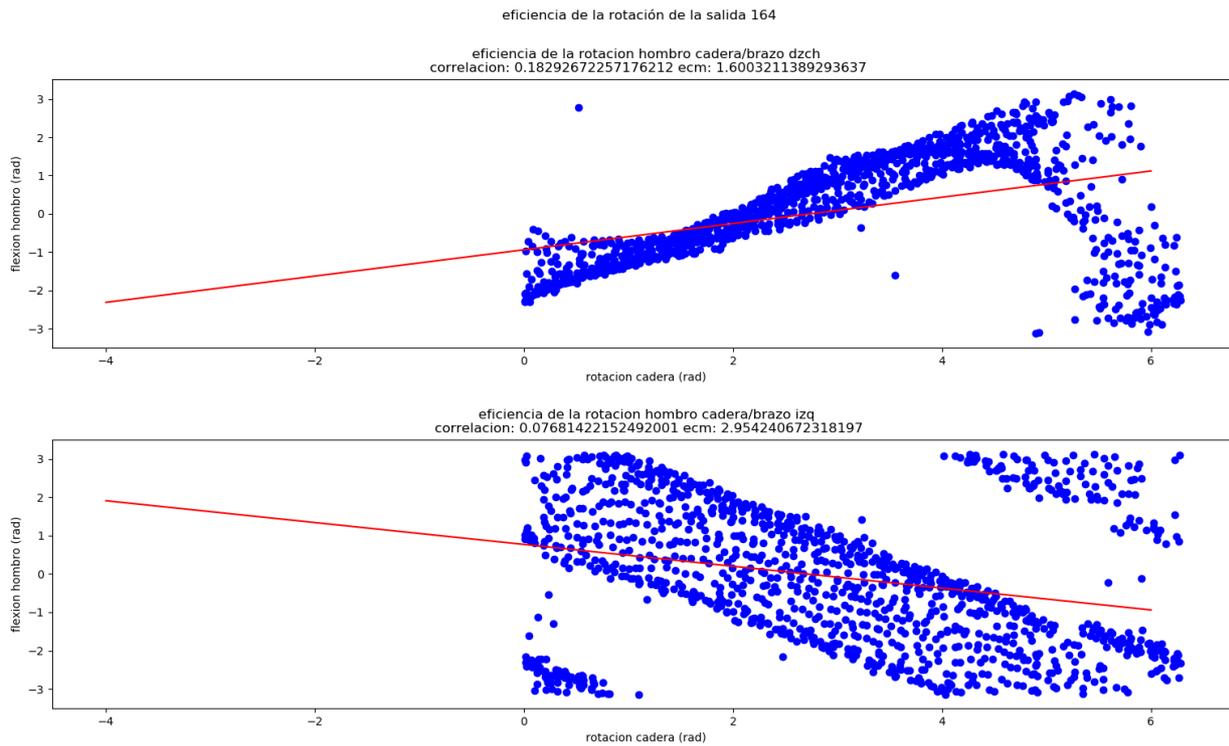
Gráfica 5.3-39. Comparación de las entradas de la flexión del codo derecho con respecto a la flexión y abducción del hombro derecho de la salida 172. Y plano representa el modelo lineal de las entradas.

movimiento del codo izquierdo. salida 172
correlacion: 0.6402923010938555 ecm: 0.11843535162965087

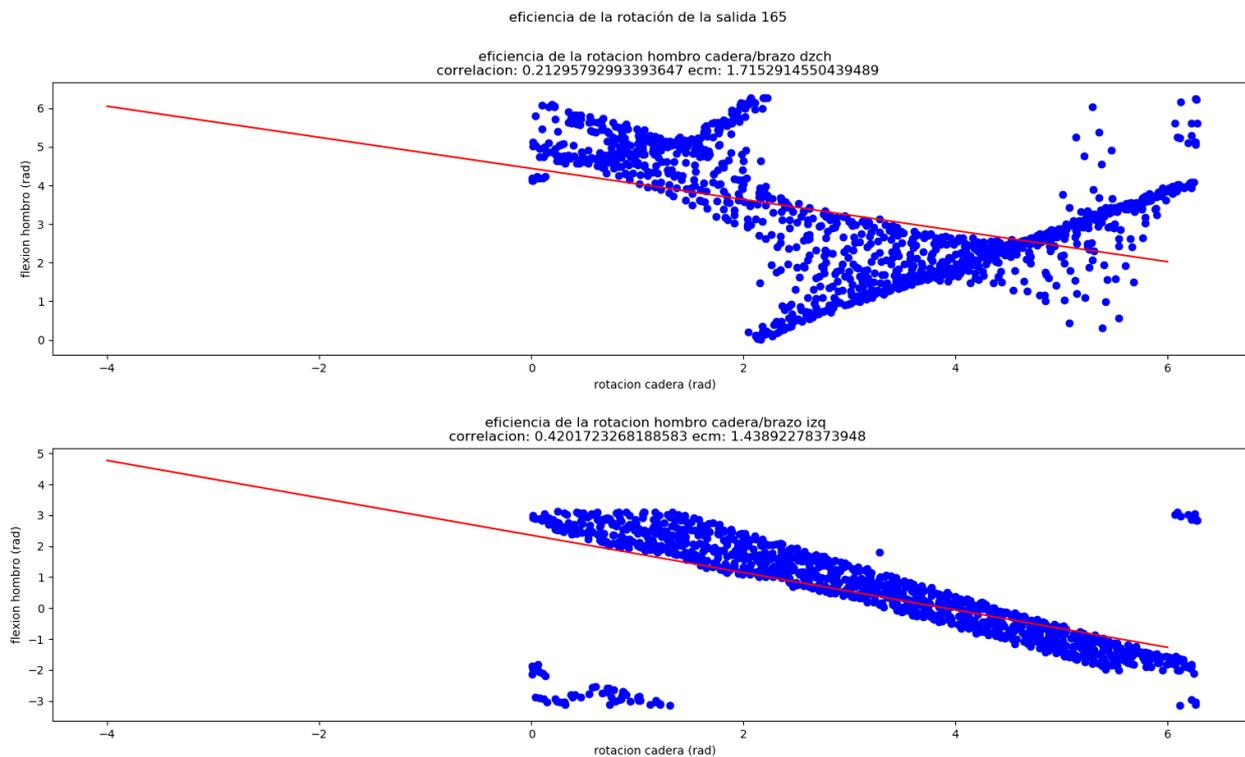


Gráfica 5.3-40. Comparación de las entradas de la flexión del codo izquierdo con respecto a la flexión y abducción del hombro izquierdo de la salida 172. Y plano representa el modelo lineal de las entradas.

Se puede ver tanto en la Gráfica 5.3-29 y la Gráfica 5.3-38 una cierta inclinación en los puntos. Para estudiar estos casos, sería mejor separar las diferentes variables independientes. Se puede ver en la Gráfica 5.3-38, que la dependencia entre la abducción y la flexión del codo forma una figura cerrada. Esta normalmente tiene una sección donde la abducción es aproximadamente vertical con respecto a la flexión del codo. Esto se relaciona con la fase de ataque. El siguiente tramo sería la de rotación en el que la abducción baja mientras que la flexión del codo sube. Para finalizar, estarían las fases de subida y salida. Que va ganando tanto abducción como flexión del codo para colocarte en posición de ataque. Aquí se puede ver patrones importantes como la altura del brazo en el ataque y como recoge el brazo en la salida. Por otra parte, las variables de la flexión del codo con respecto a la flexión del hombro. La bajada de la curva sería la fase de ataque y la subida sería la fase de la rotación. Las fases de salida y subidas hacen que la flexión se haga negativa. En este caso, se puede ver más la evolución de la flexión del codo en el ataque y en la rotación. Además, se puede ver la extensión del hombro en las fases de salida y subida. Como ejemplo de estas figuras se puede ver claramente en la Gráfica 5.3-34. Cómo se puede observar esta figura en la mayoría de las ocasiones no se ajusta adecuadamente a una regresión lineal. A lo mejor con regresiones más avanzada se puede sacar mejores parámetros de la palada.

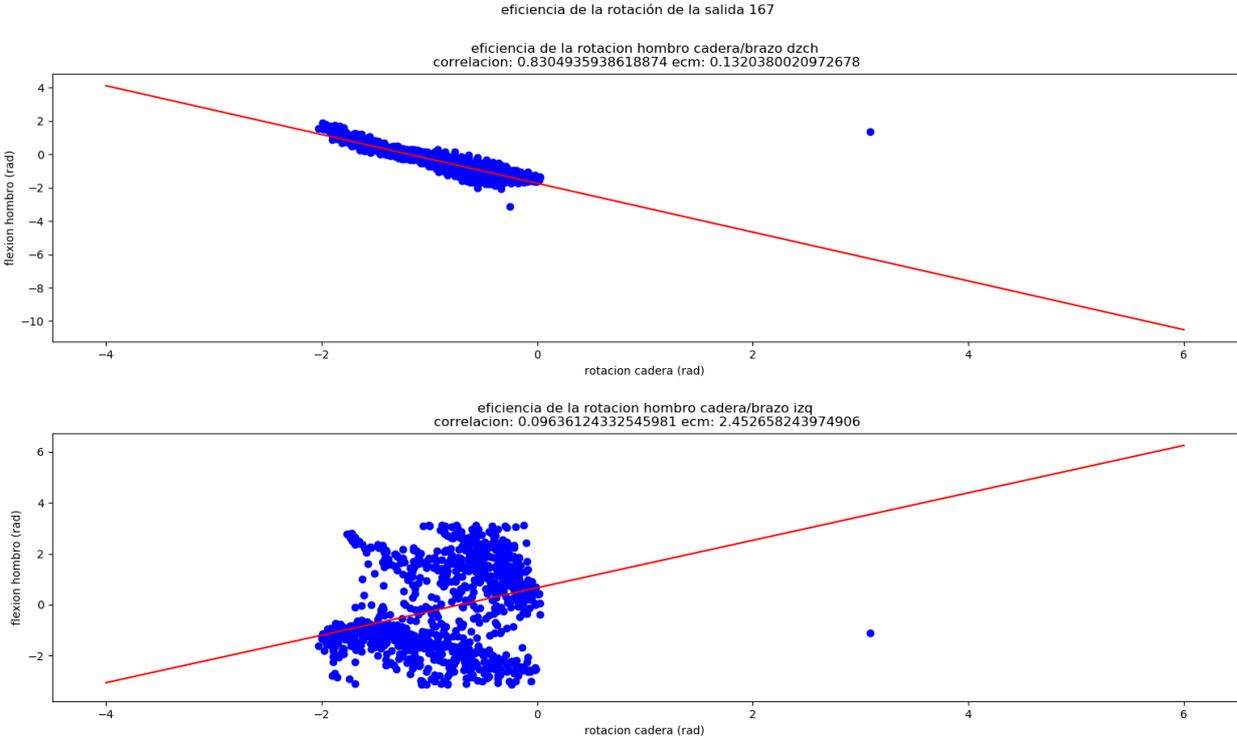


Gráfica 5.3-41. Comparación de las entradas de la flexión del hombro derecho (arriba) e izquierdo (abajo) con respecto la rotación de la cadera de la salida 164. Y la línea recta representa el modelo lineal de las entradas.

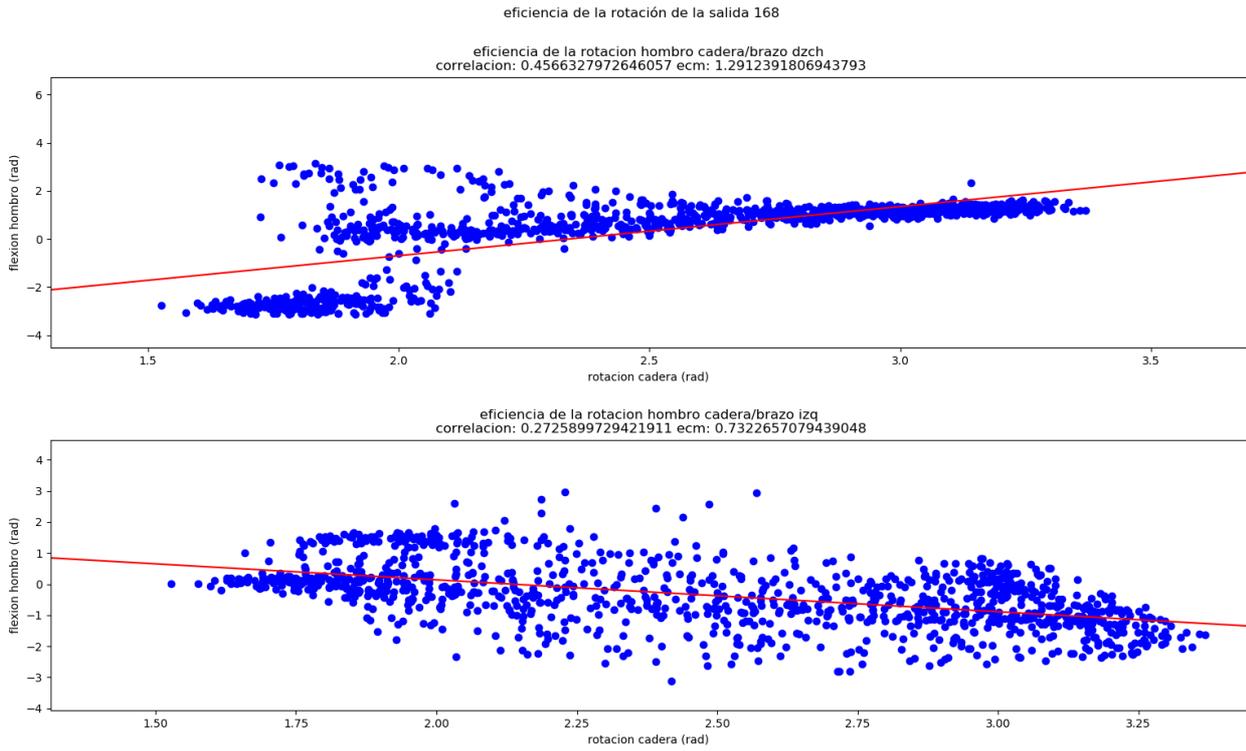


Gráfica 5.3-42. Comparación de las entradas de la flexión del hombro derecho (arriba) e izquierdo (abajo) con respecto la rotación de la cadera de la salida 165. Y la línea recta representa el modelo lineal de las entradas.

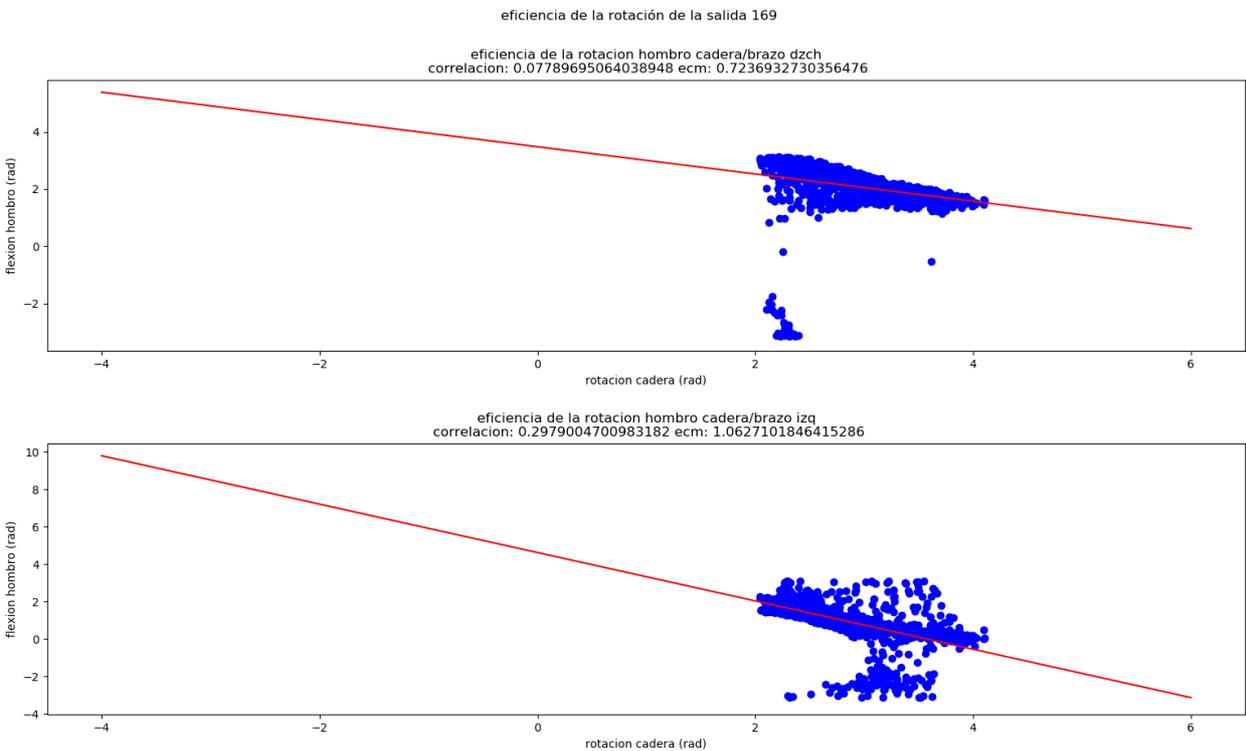
respecto la rotación de la cadera de la salida 165. Y la línea recta representa el modelo lineal de las entradas.



Gráfica 5.3-43. Comparación de las entradas de la flexión del hombro derecho (arriba) e izquierdo (abajo) con respecto la rotación de la cadera de la salida 167. Y la línea recta representa el modelo lineal de las entradas.

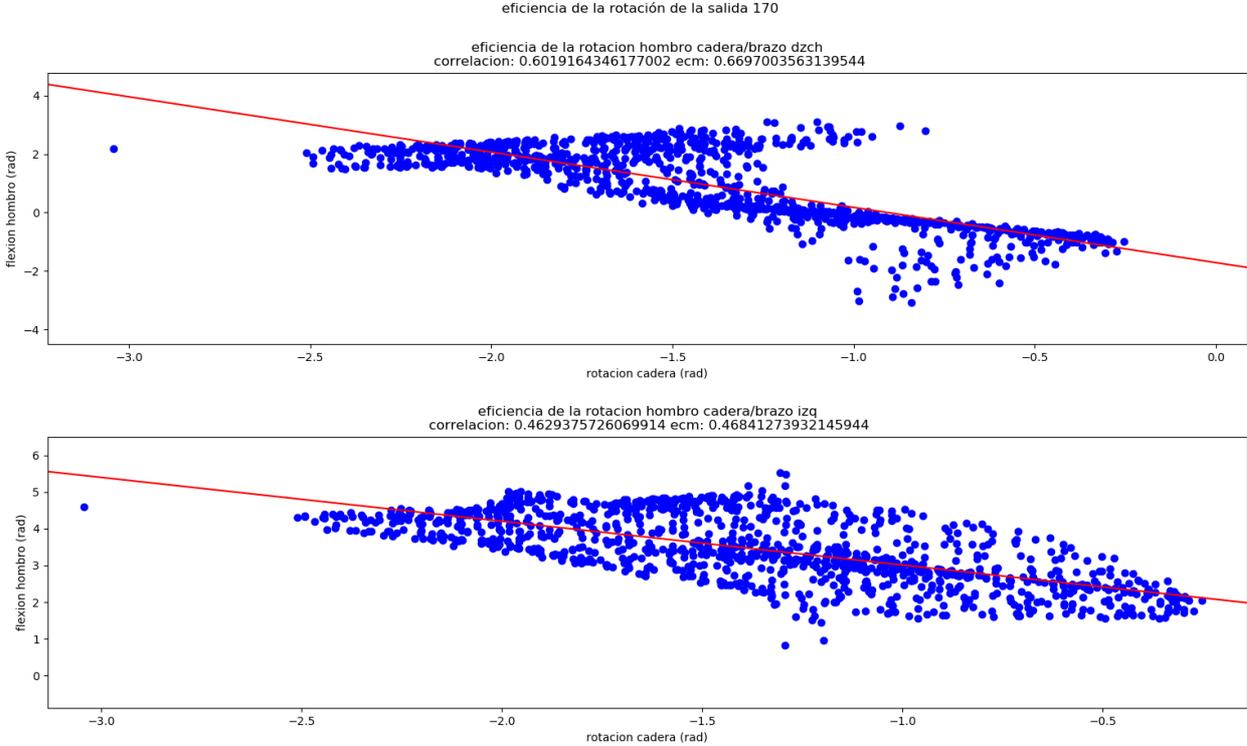


Gráfica 5.3-44. Comparación de las entradas de la flexión del hombro derecho (arriba) e izquierdo (abajo) con respecto la rotación de la cadera de la salida 168. Y la línea recta representa el modelo lineal de las entradas.

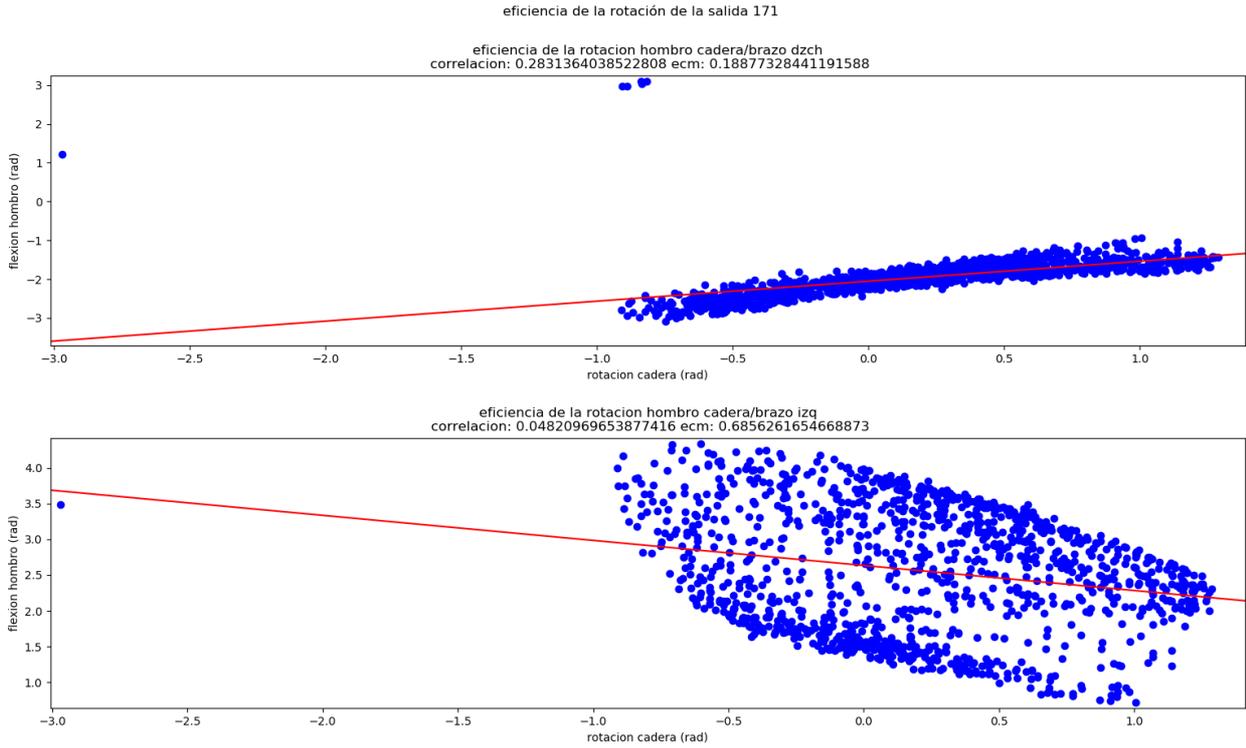


Gráfica 5.3-45. Comparación de las entradas de la flexión del hombro derecho (arriba) e izquierdo (abajo) con respecto la rotación de la cadera de la salida 169. Y la línea recta representa el modelo lineal de las entradas.

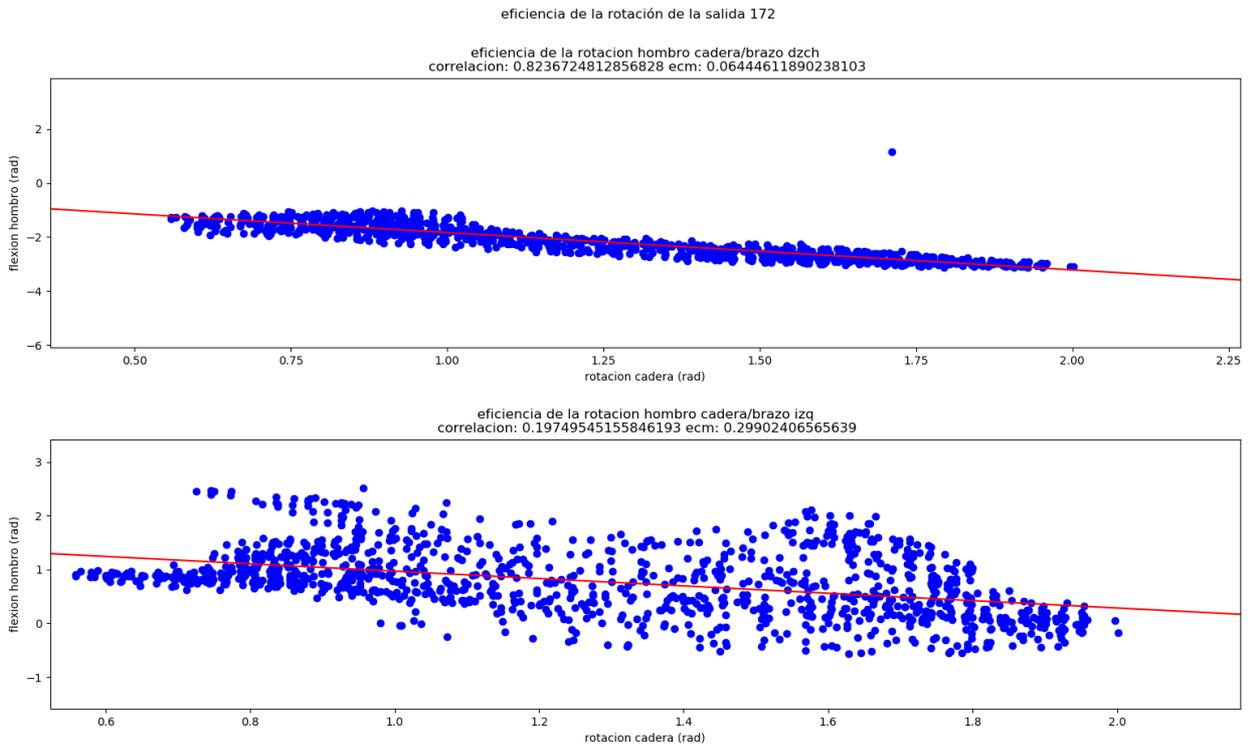
respecto la rotación de la cadera de la salida 169. Y la línea recta representa el modelo lineal de las entradas.



Gráfica 5.3-46. Comparación de las ekkntradas de la flexión del hombro derecho (arriba) e izquierdo (abajo) con respecto la rotación de la cadera de la salida 170. Y la línea recta representa el modelo lineal de las entradas.



Gráfica 5.3-47. Comparación de las entradas de la flexión del hombro derecho (arriba) e izquierdo (abajo) con respecto la rotación de la cadera de la salida 171. Y la línea recta representa el modelo lineal de las entradas.



Gráfica 5.3-48. Comparación de las entradas de la flexión del hombro derecho (arriba) e izquierdo (abajo) con respecto la rotación de la cadera de la salida 172. Y la línea recta representa el modelo lineal de las entradas.

respecto la rotación de la cadera de la salida 172. Y la línea recta representa el modelo lineal de las entradas.

Bueno de la Gráfica 5.3-47 y de la Gráfica 5.3-43, como ya se ha visto, dos gráficas en ellas. Como ya se ha visto en la salida 171, se analizará la del brazo izquierdo. En el caso de la salida 167 se analizará la gráfica del brazo derecho. En estos datos, otra vez se observa que existe una cierta inclinación en el conjunto de datos. Y se ve como estas dos variables se podrían representar mediante esta regresión lineal. La inclinación de estas gráficas refleja si utiliza más el brazo o la cadera en la rotación. Cuanto menor sea la inclinación, más utiliza la cadera.

Se ha podido observar que en las gráficas de regresiones de la salida 171, hay puntos con valor mayor que π . Esto es debido a que como se puede ver en la Gráfica 5.3-7, los valores de rotación y flexión del hombro pasan de π a menos π . Esto hace que las gráficas salgan discontinuas y la inteligencia artificial falle. Para evitar esto, se ha hecho que estos movimientos se representen en el rango de cero a dos π . Esto se ha aplicado en los casos en los que los valores estaban en torno a π y menos π . Y también se cumplía que no había valores alrededor de 0. De esta forma la mayoría de los datos no tendrán discontinuidades.

El modelo lineal permite sacar algunas conclusiones de las paladas de los piragüistas. Sin embargo, como se puede ver existen regresiones que se pueden adaptar más al modelo real. Sin embargo, se añadiría una mayor complejidad a estos modelos. Por otro lado, se puede ver como se ha realizado el análisis de la salida completa y no de la palada. De forma que se estudia el caso general de la acción de todas las paladas y no las separa independientemente. En los casos, en los que ha salido más dispersos los puntos debería comprobar cómo ha salido cada palada. No se ha realizado por falta de tecnología para medir cada una de las fases del piragüismo. Sin embargo, esto también nos permite observar que en la mayoría de los casos hay poca dispersión relativamente. Por lo tanto, todas las paladas hechas con ese brazo siguen un patrón en esos movimientos. La inclinación en cada una de las variables nos da información acerca de cómo se han realizado las paladas.

El sistema y el proceso que se ha montado ha dado en estos casos buenos resultado. Sin embargo, este sistema y el proceso de medida se puede mejorar. A su vez la sincronización de los datos de todos los microcontroladores no es perfecta. Aunque se ha elegido un periodo de muestreo lo suficientemente grande para que sea despreciable. Y se tomó las diferencias de las cinco comunicaciones despreciables. También se podría mejorar las medidas mediante un magnetómetro que arreglará el error de *drift*. Y se podría utilizar medir la postura a través de un sistema auxiliar más exacto. Para comprobar lo bien que se ajusta nuestro sistema con la realidad.

5.4 Conclusión

Bueno se ha analizado ya los dos casos que han salido mejor dentro de las pruebas. Los errores cometidos podrían ser arreglados en el futuro teniendo más tecnología a nuestra disposición. Pero como se ha comprobado el sistema permite describir algunos aspectos sobre la salida. Esto podría permitir comparar las paladas de los distintos piragüistas. De forma que pueda ayudarlo a saber qué aspectos mejorar de la misma

6 CONCLUSIÓN

Para finalizar el trabajo, se va a hablar sobre si con este se han cumplido los objetivos planteados con este. Para empezar sí que se consiguieron sacar datos acerca de la postura de los piragüistas a través de sensores. Para ello se ha usado unos **IMU** (*Inertial Measurement Unit*), que se ha colocado en partes del tren superior. De tal forma que, con un análisis posterior, consiguió obtener la postura del piragüista en todo momento. Sí que es cierto que se ha necesitado de la posición inicial y que no es la postura completa. Esto es debido a que nos falta articulaciones por medir. Además, al hombro todavía le queda tres grados de libertad que medir para medir toda su movilidad. Sin embargo, con estos datos de la medida son suficientes para poder realizar un buen modelo de la técnica. Y la velocidad de muestreo es de 100 ms que considero suficiente para medir salidas a R1 o R2. Por otro lado, solo el uso de giroscopio y acelerómetro permite medir el ángulo. Aunque si es cierto que hay que tener cuidado cómo evoluciona el *drift* y muchas salidas no son válidas.

Por otro lado, se tiene la Raspberry pi, como módulo central. De tal forma, que puede comunicarse y almacenar estos. Este tiene una memoria **RAM** (*Random Access Memory*) de 1GB que es más que suficiente para medir medidas cortas relativamente. Y una memoria SD (*Secure Digital*) de 32GB, donde puede almacenar todos datos de la **BBDD** (base de datos). Y también es capaz de llevar la comunicación con los microcontroladores simultáneamente. La solución que se ha dado a la comunicación es Bluetooth. Puede que esta solución sea un poco incomoda a la hora de realizar medidas. Ya que se tiene que encontrar la Raspberry pi en un barco y esto dificulta la accesibilidad a ella. Y ya que, los sensores no tienen un procesador para controlarlo ni modulo Bluetooth. Se ha tenido que añadir el microcontrolador ESP 32, para que sea intermediario entre ambos lados. Además, al estar en un medio acuático, se tiene que encargarse de la protección de esta al agua. Pero esto asegura que los sensores siempre van a estar a rango y disminuye la distancia con ello. Lo que permita que la diferencia de tiempos en la comunicación con ellos no sea excesiva. Además, como tiene capacidad de sobra esta misma puede servir como centro de análisis, para desarrollar la inteligencia artificial. Este ha creado una regresión lineal que saca las características principales del modelo que describe la técnica. De tal forma, que facilita sacar sus características de la técnica.

Para realizar las medidas en los casos reales, se ha creado un traje donde podemos equipar los sensores. Estos a su vez están almacenado junto a su microcontrolador en un recipiente que impide la entrada de agua. Se puede comunicar con estos a través de una sesión de Shell remota creada en el dispositivo móvil. Los programas están diseñados para esperar a que se le dé el comienzo de la prueba. Además de estar diseñada para saber cuándo pararla. Para terminar, se van a mencionar las continuaciones de este trabajo que se han podido sacar.

- Realizar el chip integrado con todas las características que necesita el módulo sensor. De esta forma se conseguirá que sea más cómodo para los deportistas. Y que este se pueda ajustar mejor a la parte del cuerpo correspondiente.
- Se podría diferenciar las diferentes paladas y hacer el modelo de cada palada y no de cada salida. Esto aportaría más datos y más información de las diferencias entre cada una. Ya que se puede ver como varía este con el tiempo.
- Mediante muchos datos, se podría tener una inteligencia artificial que clasificará como reman. De tal forma, que te indicará que aspectos tienes que mejorar y a quien te tienes que parecer. Para ello se debe tener muchos datos sobre las salidas de diferentes personas. Que estarían clasificado y los buenos piragüistas formarían el grupo de entrenamiento. De tal forma, que los demás deportistas se asociarán a algunas de las buenas paladas. Que ha sido con las que se ha entrenado a la inteligencia. O a una mezcla de diferentes características de esta. Dependiendo de las características físicas del deportista y como rema.
- Poner más sensores que midan más articulaciones. Por ejemplo, el estudio de la muñeca o de las piernas sería interesante. Ya que, aunque no lo parezca el movimiento de piernas es fundamental en la técnica del piragüismo. Además, se puede incluir otros sensores, que midan características de la salida. Como la velocidad que ha alcanzado. Esto permitiría ver la relación entre las características del barco y la palada.
- Estos mismos modelos se podrían aplicar de diferentes formas a otros deportes similares. Como sería

el remo o la canoa.

- Crear una regresión más exacta para representar un modelo más parecido a la realidad.

Lo anterior serían algunas ideas para incluir en para la mejora del trabajo. Además, de algunas ideas para saber qué dirección tomará los futuros trabajos.

7.1 Interfaces

```
pi@raspberrypi:~/TFG_ale $ python3 confBBDD.py
Bienvenido a la configuracion de la BBDD, que deseas hacer:
1-Introducir o cambiar datos de un piraguista
2-Introducir o cambiar datos de un barco
3-Introducir o cambiar datos de una pala
4-Cambiar datos de una salida
5-Añadir los angulos a la base de datos a partir de los cuaterniones
1
Quieres actualizar o insertar algún piraguista:[A/C]
C
Si quieres dejar algun dato en blanco solo escribe - en su apartado
Introduce el nombre del piraguista:

Introduce los apellidos del piraguista:

Introduce la altura del piraguista:

Introduce el peso del piraguista:

Introduce la longitud del torso del piraguista:

Introduce la longitud del antebrazo del piraguista:

Introduce la longitud del hombro al codo del piraguista:

Introduce la longitud del cuello al hombro del piraguista:

Introduce la categoria del piraguista:

Introduce el club del piraguista:

Introduce el nivel del piraguista:
```

Ilustración 7.1-1. Ejemplo para mostrar la interfaz para introducir los datos de un nuevo piragüista.

```
pi@raspberrypi:~/TFG_ale $ python3 confBBDD.py
Bienvenido a la configuracion de la BBDD, que deseas hacer:
1-Introducir o cambiar datos de un piraguista
2-Introducir o cambiar datos de un barco
3-Introducir o cambiar datos de una pala
4-Cambiar datos de una salida
5-Añadir los angulos a la base de datos a partir de los quaterniones
1
Quieres actualizar o insertar algún piraguista:[A/C]
A
Si quieres dejar algun dato en blanco solo escribe - en su apartado
cual es su nombre
Eloy
cuales son sus apellidos

cuales es su club

Selecciona aquel que estas buscando
1- nombre: Eloy apellidos: Parra Boyero
   altura: 1.66 peso: 68.0
   categoria: veterano club: club kayak Torneo nivel: medio
1
Introduce los nuevos datos del piraguista
Si quieres dejar algun dato en blanco solo escribe - en su apartado
Introduce el nombre del piraguista:

Introduce los apellidos del piraguista:

Introduce la altura del piraguista:

Introduce el peso del piraguista:

Introduce la longitud del torso del piraguista:

Introduce la longitud del antebrazo del piraguista:

Introduce la longitud del hombro al codo del piraguista:

Introduce la longitud del cuello al hombro del piraguista:

Introduce la categoria del piraguista:

Introduce el club del piraguista:

Introduce el nivel del piraguista:
```

Ilustración 7.1-2. Ejemplo para mostrar la interfaz para actualizar los datos de un piragüista.

```
pi@raspberrypi:~/TFG_ale $ python3 confBBDD.py
Bienvenido a la configuracion de la BBDD, que deseas hacer:
1-Introducir o cambiar datos de un piraguista
2-Introducir o cambiar datos de un barco
3-Introducir o cambiar datos de una pala
4-Cambiar datos de una salida
5-Añadir los angulos a la base de datos a partir de los quaterniones
2
Quieres actualizar o insertar algún barco:[A/C]
C
Si quieres dejar algun dato en blanco solo escribe - en su apartado
Introduce el modelo del barco:

Introduce el fabricante del barco:

Introduce el tamaño del barco:

Introduce los datos del propietario
Si quieres dejar algun dato en blanco solo escribe - en su apartado
cual es su nombre

cuales son sus apellidos

cuales es su club
```

Ilustración 7.1-3. Ejemplo para mostrar la interfaz para introducir los datos de un nuevo barco.

```
pi@raspberrypi:~/TFG_ale $ python3 confBBDD.py
Bienvenido a la configuracion de la BBDD, que deseas hacer:
1-Introducir o cambiar datos de un piraguista
2-Introducir o cambiar datos de un barco
3-Introducir o cambiar datos de una pala
4-Cambiar datos de una salida
5-Añadir los angulos a la base de datos a partir de los quaterniones
2
Quieres actualizar o insertar algún barco:[A/C]
A
cual es el modelo

cual es su fabricante

A continuacion vamos a buscar el propietario del barco
Si quieres dejar algun dato en blanco solo escribe - en su apartado
cual es su nombre
David
cuales son sus apellidos

cuales es su club

Selecciona aquel que estas buscando
1- nombre: David apellidos: Rodriguez
   altura: 1.76 peso: 80.0
   categoria: senior club: club kayak Torneo nivel: medio
1
propietario = %s AND
[3, 'Vanquish 3', 'Nelo', None, 'David', 'Rodriguez', 'club kayak Torneo']
Selecciona cual es el barco
1- modelo: Vanquish 3 fabricante: Nelo tamaño: -
   datos del propietario
   nombre: David apellidos: Rodriguez club: club kayak Torneo
1
Introduce los nuevos datos del barco
Si quieres dejar algun dato en blanco solo escribe - en su apartado
Introduce el modelo del barco:

Introduce el fabricante del barco:

Introduce el tamaño del barco:

Introduce los datos del propietario
Si quieres dejar algun dato en blanco solo escribe - en su apartado
cual es su nombre

cuales son sus apellidos

cuales es su club
```

Ilustración 7.1-4. Ejemplo para mostrar la interfaz para actualizar los datos de un barco.

```
pi@raspberrypi:~/TFG_ale $ python3 confBBDD.py
Bienvenido a la configuracion de la BBDD, que deseas hacer:
1-Introducir o cambiar datos de un piraguita
2-Introducir o cambiar datos de un barco
3-Introducir o cambiar datos de una pala
4-Cambiar datos de una salida
5-Añadir los angulos a la base de datos a partir de los quaterniones
3
Quieres actualizar o insertar alguna pala:[A/C]
C
Si quieres dejar algun dato en blanco solo escribe - en su apartado
Introduce el modelo de la pala:

Introduce la longitud de la pala:

Introduce los grado de la pala:

Introduce el fabricante de la pala:

Introduce los datos del propietario
Si quieres dejar algun dato en blanco solo escribe - en su apartado
cual es su nombre

cuales son sus apellidos

cuales es su club
```

Ilustración 7.1-5. Ejemplo para mostrar la interfaz para introducir los datos de una nueva pala.

```

pi@raspberrypi:~/TFG_ale $ python3 confBBDD.py
Bienvenido a la configuracion de la BBDD, que deseas hacer:
1-Introducir o cambiar datos de un piraguita
2-Introducir o cambiar datos de un barco
3-Introducir o cambiar datos de una pala
4-Cambiar datos de una salida
5-Añadir los angulos a la base de datos a partir de los quaterniones
3
Quieres actualizar o insertar alguna pala:[A/C]
A
cual es el modelo

cual es su fabricante

A continuacion vamos a buscar el propietario del barco
Si quieres dejar algun dato en blanco solo escribe - en su apartado
cual es su nombre
Juan Antonio
cuales son sus apellidos

cuales es su club

Selecciona aquel que estas buscando
1- nombre: Juan Antonio apellidos: Écija Fernández
   altura: 1.64 peso: 71.0
   categoria: veterano club: club kayak Torneo nivel: bajo
1
[2, 'IV', 'Braca-sport', None, None, 'Juan Antonio', 'Écija Fernández', 'club kayak Torneo']
Selecciona cual es la pala
1- modelo: IV fabricante: Braca-sport
   longitud: - angulo: -
   datos del propietario
   nombre: Juan Antonio apellidos: Écija Fernández club: Écija Fernández
1
Introduce los nuevos datos de la pala
Si quieres dejar algun dato en blanco solo escribe - en su apartado
Introduce el modelo de la pala:

Introduce la longitud de la pala:

Introduce los grado de la pala:

Introduce el fabricante de la pala:

Introduce los datos del propietario
Si quieres dejar algun dato en blanco solo escribe - en su apartado
cual es su nombre

cuales son sus apellidos

cuales es su club

```

Ilustración 7.1-6. Ejemplo para mostrar la interfaz para actualizar los datos de una pala.

```

pi@raspberrypi:~/TFG_ale $ python3 confBBDD.py
Bienvenido a la configuracion de la BBDD, que deseas hacer:
1-Introducir o cambiar datos de un piraguista
2-Introducir o cambiar datos de un barco
3-Introducir o cambiar datos de una pala
4-Cambiar datos de una salida
5-Añadir los angulos a la base de datos a partir de los quaterniones
4
Indica el piraguista que participo en la salida
Si quieres dejar algun dato en blanco solo escribe - en su apartado
cual es su nombre
Eloy
cuales son sus apellidos

cuales es su club

Selecciona aquel que estas buscando
1- nombre: Eloy apellidos: Parra Boyero
   altura: 1.66 peso: 68.0
   categoria: veterano club: club kayak Torneo nivel: medio
1
Indica la pala que participo en la salida
cual es el modelo

cual es su fabricante

A continuacion vamos a buscar el propietario del barco
Si quieres dejar algun dato en blanco solo escribe - en su apartado
cual es su nombre

cuales son sus apellidos

cuales es su club

No se han pasado ningun argumento.
No se han pasado ningun argumento.
Indica el barco que participo en la salida
cual es el modelo

cual es su fabricante

A continuacion vamos a buscar el propietario del barco
Si quieres dejar algun dato en blanco solo escribe - en su apartado
cual es su nombre

cuales son sus apellidos

cuales es su club

No se han pasado ningun argumento.
No se han pasado ningun argumento.
Indica la fecha de la salida(yyyy-m-d)

Selecciona aquel que estas buscando
1- velocidad: - tiempo: 120000 distancia:-
   ritmo: - fecha: 2021-07-16 descripcion: eloy 1
   Datos del piraguista
     nombre: Eloy apellidos: Parra Boyero club: club kayak Torneo
   Datos de la pala
     modelo: ExtremeXL fabricante: Roman
   Datos del barco
     modelo: 4 fabricante: Nelo
2- velocidad: - tiempo: 120000 distancia:-
   ritmo: - fecha: 2021-07-16 descripcion: eloy 2
   Datos del piraguista
     nombre: Eloy apellidos: Parra Boyero club: club kayak Torneo
   Datos de la pala
     modelo: ExtremeXL fabricante: Roman
   Datos del barco
     modelo: 4 fabricante: Nelo
3- velocidad: - tiempo: 120000 distancia:-
   ritmo: - fecha: 2021-07-16 descripcion: eloy 3
   Datos del piraguista
     nombre: Eloy apellidos: Parra Boyero club: club kayak Torneo
   Datos de la pala
     modelo: ExtremeXL fabricante: Roman
   Datos del barco
     modelo: 4 fabricante: Nelo
4- velocidad: - tiempo: 120000 distancia:-
   ritmo: - fecha: 2021-07-16 descripcion: eloy 4
   Datos del piraguista
     nombre: Eloy apellidos: Parra Boyero club: club kayak Torneo
   Datos de la pala
     modelo: ExtremeXL fabricante: Roman
   Datos del barco
     modelo: 4 fabricante: Nelo
1
Introduce los nuevos datos de la salida
Vamos a tomar los datos de la salida.
Si quieres dejar algun dato en blanco solo escribe - en su apartado
el deportista que va a competir esta en la base de datos[S/N]

la pala con la que va a competir esta en la base de datos[S/N]

el barco con el que va a competir esta en la base de datos[S/N]

cual es el ritmo que hay que llevar:

cual es la distancia que recorre:

cual es la longitud de la pedalina:

Indica la fecha de la salida(yyyy-m-d)

```

Ilustración 7.1-7. Ejemplo para mostrar la interfaz para actualizar los datos de una salida.

REFERENCIAS

1. *Modern Technologies in Sport, with Reference to Video Technologies*. **Viduka, dejan, Ilić, Luka and Dimitrijević, Vanja**. Belgrade : Singidunum University, 2021. in Sinteza 2021 - International Scientific Conference on Information Technology and Data Related Research. pp. 277-281.
2. **VCR**. El primer uso de Video Assistant Referee (VAR) para waterpolo en los juegos olímpicos. *VCR*. [Online] <https://vcr.com.co/el-primer-uso-de-video-assistant-referee-var-para-waterpolo-en-los-juegos-olimpicos/>.
3. **Armenteros, Manuel, Benitez, Anto J. and Betancor, Miguel Angel, [ed.]**. *The Use of Video Technologies in Refereeing Football and Other Sports*. s.l. : Routledge, 2019. 9781138312043.
4. *Evaluation of MRI-US Fusion Technology in Sports-Related Musculoskeletal Injuries*. **Wong-on, Manuel, Til-Pérez, Lluís and Balius, Ramón**. 32, 2015, *Advances in Therapy*, pp. 580-594.
5. *Eye Tracking technology in sport-related concussion: a systematic review and meta-analysis*. **Snegireva, N, et al.** 12, 20 December 2018, *Physiological measurement*, Vol. 39.
6. **Intertek**. Neuromarketing-Técnica Eye-Tracking. *intertek*. [Online] <https://www.intertek.es/analisis-sensorial/eye-tracking/>.
7. *Exploring the Role of Wearable Technology in Sport Kinematics and Kinetics: A Systematic Review*. **Adesida, Yewande, Papi, Enrica and McGregor, Alison H.** 7, 2 April 2019, *Sensors* [online], Vol. 19.
8. *Overview of progress in Smart-Clothing project for health monitoring and sport applications*. **Borges, Luis M., et al.** Aalborg : s.n., 2008. International Symposium on Applied Sciences on Biomedical and Communication Technologies, ISABEL. 9781424426478.
9. **MacDonald, Kerry, et al.** Validation of an inertial measurement unit for the measurement of jump count and height. 7 December 2016, Vol. 25, pp. 15-19.
10. *CanoeSense: Monitoring canoe sprint motion using wearable sensors*. **Wang, Zhelong, et al.** Budapest : IEEE, 2016. IEEE International Conference on Systems, Man and Cybernetics. 16656826.
11. **Camomilla, Valentina, et al.** Trends Supporting the In-Field Use of Wearable Inertial Sensors for Sport Performance Evaluation: A Systematic Review. *Sensor* [online]. 15 March 2018, Vol. 18, 3.
12. *Golfer Body Posture Monitoring Using e-Textile Technology*. **Caya, Meo Vicent C., et al.** Nagoya : s.n., 2018. 2018 3rd International Conference on Computer and Communication Systems (ICCCS). 9781538663516.
13. *A pilot study on the e-kayak System: A wireless DAQ Suited for performance analysis in flatwater Sprint kayaks*. **Bonaiuto, Vincenzo, et al.** 2, s.l. : MDPI, *Sensors*, Vol. 20.
14. *Development of Kayak Ergometer using Electromagnetic Brake for Indoor Training*. **Hashimoto, Takuya, Kondo, Kazuki and Kikuchi, Naoki**. Tsu : IEEE, 2018. 12th France-Japan and 10th Europe-Asia Congress on Mechatronics . 18167455.
15. **Braca-sport**. Ergómetro Dansprint PRO. *Braca-sport*. [Online] <https://nauticalia.es/producto/ergometro-dansprint-pro/>.
16. *Tridimensional kinematic analysis on a kayaking simulation: key factors to success performance*. **Limonta, Eloisa, et al.** 27-34, s.l. : Springer Link, 2010, *Sport Sci Health*, Vol. 6.
17. **elsuperhinch**. Piragüismo para novatos. Historia, Tipos, Reglas básicas... *el superhinch*. [Online] <https://elsuperhinch.com/piraguismo-para-novatos-historia-tipos-reglas-basicas/#:~:text=Como%20deporte%2C%20empez%3%B3%20a%20surgir,los%20n%C3%B3rdicos%20y%20centroeuropeos%20despu%C3%A9s..>

18. **Tomás Mora, Iván.** *Dispositivo electrónico para el entrenamiento en piragüismo.* Zaragoza : Escuela de Ingeniería y Arquitectura Universidad de Zaragoza, 2020.
19. **Cardona Lenis, Andrés Felipe.** Técnica de paleo en el piragüismo. *Unycos.* [Online] 27 1 2021. <https://unycos.com/blog/tecnica-paleo-piraguismo/>.
20. *AI-IMU Dead-Reckoning.* **Brossard, Martin, Barrau, Axel and Bonnabel, Silvère.** 13 March 2020, IEEE Transactions on Intelligent Vehicles, Vol. 5, pp. 585-595.
21. *RIDI: Robust IMU Double Integration.* **Yan, Hang, Shan, Qi and Furukawa, Yasutaka.** 2018. European Conference on Computer Vision (ECCV). pp. 621-636.
22. *A Kalman Filter-Based Algorithm for IMU-Camera Calibration: Observability Analysis and Performance Evaluation.* **Mirzaei, Faraz M. and Roumeliotis, Stergios I.** 3 October 2008, IEEE Transactions on Robotics, Vol. 24, pp. 1143-1156.
23. **Ahmad, Norhafizan, et al.** Reviews on Various Inertial Measurement Unit. *International Journal of Signal Processing Systems.* 2 December 2013, Vol. 1, 2, pp. 256-262.
24. **Farnell.** Farnell. *Acelerómetro.* [Online] <https://es.farnell.com/sensor-accelerometer-motion-technology>.
25. **Sheldon.** La física tras el acelerómetro. *Átomos y bits.* [Online] 13 5 2014. <https://atomosybits.com/la-fisica-tras-el-acelerometro/>.
26. **Omega.** Omega. *Accelerometers.* [Online] 28 August 2018. <https://www.omega.com/en-us/resources/accelerometers>.
27. **Madrigal Sastré, Heberto.** *Implementación de un sistema AHRS basado en un observador no lineal.* Puebla : Benemérita Universidad Autónoma de Puebla. Facultad de Ciencias electrónicas, 2011.
28. **Farnell.** Farnell. *Giroscopio.* [Online] <https://es.farnell.com/sensor-gyroscope-technology>.
29. <https://www.britannica.com>. <https://www.britannica.com/technology/gyroscope>. [Online] Encyclopedia Britannica, 12 November 2020.
30. **Apostolyuk, Vladislav.** *Theory and Design of Micromechanical Vibratory Gyroscopes.* Physic, Artrise Corporation. 2007.
31. **Persson, Anders.** How Do We Understand the Coriolis Force? *Bulletin of the American Meteorological Society.* 1 July 1998, Vol. 79, pp. 1373-1386.
32. **Luis Llamas.** Cómo usar un giroscopio en nuestros proyectos en Arduino. *Luis Llamas.* [Online] 4 9 2016. <https://www.luisllamas.es/como-usar-un-giroscopio-arduino/>.
33. **Wikipedia.** Giróscopio láser de anillo. *Wikipedia The Free Encyclopedia.* [Online] https://es.wikipedia.org/wiki/Gir%C3%B3scopo_l%C3%A1ser_de_anillo.
34. laplace.us.es/wiki/. http://laplace.us.es/wiki/index.php/%C3%81ngulos_de_Euler_y_de_navegaci%C3%B3n. [Online] 26 November 2016.
35. **Pérez López, Paula.** *Un método de calibración de sensores.* 2015.
36. **Ben-Ari, Moti.** A tutorial on Euler Angle. *weizman.* [Online] 2.0.1. <https://www.weizmann.ac.il/sci-tea/benari/sites/sci-tea.benari/files/uploads/softwareAndLearningMaterials/quaternion-tutorial-2-0-1.pdf>.
37. **Eberly, David.** <https://www.geometrictools.com/>. <https://www.geometrictools.com/Documentation/EulerAngles.pdf>. [Online] 28 April 2020.
38. **Wyss-Gallifent, Justin.** <http://www.math.umd.edu>. http://www.math.umd.edu/~immortal/MATH431/lecturenotes/ch_gimballock.pdf. [Online] 26 August 2021.
39. **Wikipedia.** Quaternions and spatial rotation. *Wikipedia The Free Encyclopedia.* [Online] https://en.wikipedia.org/wiki/Quaternions_and_spatial_rotation.
40. **Jia, Yan-Bin.** Quaternions and Rotations. [Online] 10 9 2013.

<https://graphics.stanford.edu/courses/cs348a-17-winter/Papers/quaternion.pdf>.

41. *The Case of the Misguided Gyro*. Beavers, Ian. 139, March 2017, Analog Dialogue.
42. Kok, Manon and Schön, Thomas B. Magnetometer Calibration Using Inertial Sensors. *IEEE Sensors Journal*. 15 July 2016, Vol. 16, pp. 5679-5689.
43. Luís Llamas. Determinar la orientación con Arduino y el IMU MPU-6050. *Luís Llamas*. [Online] 22 9 2016. <https://www.luisllamas.es/arduino-orientacion-imu-mpu-6050/>.
44. Naylamp Mechatronics. Tutorial MPU6050, acelerómetro y giroscopio. *Naylamp Mechatronics*. [Online] https://naylampmechatronics.com/blog/45_tutorial-mpu6050-acelerometro-y-giroscopio.html.
45. Tecnófilo. GY-521 6DOF MPU-6050 MÓDULO SENSOR DE 3 EJES ACELERÓMETRO Y GIRÓSCOPO. *Tecnófilo*. [Online] <https://www.tecnofilo.es/sensores/71-gy-521-6dof-mpu-6050-modulo-sensor-de-3-ejes-acelerometro-y-giroscopo.html>.
46. InvenSense Inc. MPU-6000 and MPU-6050 Product Specification Revision 3.4. Sunnyvale : s.n., 2013. PS-MPU-6000A-00.
47. Luís Llamas. El bus I2C en Arduino. *Luis Llamas*. [Online] 2016 5 18. [Cited: 22 7 2022.] <https://www.luisllamas.es/arduino-i2c/>.
48. Robots Didácticos. Descripción y funcionamiento del Bus I2C. *Robot didáctico*. [Online] 24 6 2019. [Cited: 21 7 2022.] <https://robots-argentina.com.ar/didactica/descripcion-y-funcionamiento-del-bus-i2c/>.
49. Boot & Work Corp. I2C Bus en el PLC Arduino para automatización industrial. *Industrial shields*. [Online] 11 2 2020. https://www.industrialshields.com/es_ES/blog/blog-industrial-open-source-1/post/i2c-bus-en-el-plc-arduino-para-automatizacion-industrial-192.
50. Lutkevich, Ben. microcontroller (MCU). *TechTarget*. 11 2019.
51. Cadence PCB solutions. What is an MCU and How do Microcontroller Units work. *cadence*. [Online] 2020. <https://resources.pcb.cadence.com/blog/2020-what-is-an-mcu-and-how-do-microcontroller-units-work>.
52. Solé, Roberto. RISC: La arquitectura de procesadores usada por ARM para cambiar el mercado. *professional review*. [Online] 17 7 2021. <https://www.profesionalreview.com/2021/07/17/que-es-risc/>.
53. Espressif Systems. ESP32 Series Datasheet. *ESPRESSIF*. [Online] 2022. https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.
54. Espressif. ESP32-WROOM-32D & ESP32-WROOM-32U. *ESPRESSIF*. [Online] 2021. https://www.mouser.es/datasheet/2/891/esp32_wroom_32d_esp32_wroom_32u_datasheet_en-1365844.pdf.
55. HiLetgo. HiLetgo ESP-WROOM-32 ESP32 ESP-32D Development Board 2.4GHz Dual-Mode WiFi + Bluetooth Dual Cores Microcontroller Processor Integrated with Antenna RF AMP Filter AP STA for Arduino IDE. *Amazon*. [Online] <https://www.amazon.es/HiLetgo-ESP-wroom-32-Desarrollo-procesador-microcontrolador/dp/B0718T232Z>.
56. Kurniawan, Agus. *Internet of Things Projects with ESP 32*. Birmingham : Packt, 2019. 978-1-78995-687-0.
57. *Literature Survey on Physiotherapy Assessment Using ESP32 and Memes Technology Based IMU Sensor as a Wearable Sleeve*. Vachanamruth, et al. 1, s.l. : Journal of VLSI Design and Signal Processing, 2022, Vol. 8. 2581-8449.
58. *Performance evaluation of ESP32 Camera Face Recognition for various projects*. Prasetyo Adi, Puput Dani. s.l. : Ascee, 2022. 2774-4353.
59. Bluetooth. *Vision and Mission*. [Online] <https://www.bluetooth.com/about-us/vision/>.
60. Sánchez Daze, Jose Manuel. Bluetooth: la historia de un sistema basado en un rey danés. *ABC*. 11 November 2014.
61. Bluetooth. *Understanding Bluetooth Range*. [Online] <https://www.bluetooth.com/learn-about->

bluetooth/key-attributes/range/.

62. Alonso, Rodrigo. *hard zone. componentes.* [Online] 27 08 2020. <https://hardzone.es/tutoriales/componentes/perfiles-bluetooth/>.
63. García, Rocío. *Adsl Zone. Tecnología.* [Online] 23 03 2021. <https://www.adslzone.net/reportajes/tecnologia/bluetooth/>.
64. Bluetooth SIG proprietary. Specification of the Bluetooth System, v5.0. *Bluetooth.* [Online] 6 12 2016. <https://www.mouser.it/pdfdocs/bluetooth-Core-v50.pdf>.
65. Fernández , Yúbal. *Xataka. Xataka Basics.* [Online] 21 04 2020. <https://www.xataka.com/basics/bluetooth-diferencias-caracteristicas-sus-clases-versiones>.
66. INCIBE. *Analizando Bluetooth. incibe-cert_.* [Online] 26 7 2016. <https://www.incibe-cert.es/blog/analizando-bluetooth>.
67. Departamento de Ingeniería Telemática. *Practica 5: Introduccion Raspberry pi.* Sevilla : Escuela Técnica Superior de Ingeniería. Universidad de Sevilla , 2017.
68. Velasco, Rubén . *Análisis: Raspberry pi 3 modelo B+.* *hard zone.* [Online] 24 8 2021. <https://hardzone.es/reviews/perifericos/analisis-raspberry-pi-3-modelo-b/>.
69. Dow, Colin. *Internet of Things Programming Projects.* Birmingham : Packt, 2018. 978-1-78913-480-3.
70. Tang, Jeff. *Developing TensorFlow Apps on Raspberyy Pi Setting up Raspberry Pi and making it move. Intelligent Mobile Projects with Tensor flow.* Birmingham : Packt, 2018.
71. ARM. *Introduction to the ARM Architecture. ARM Architecture Reference Manual.* s.l. : Addison-Wesley , 2001.
72. Alonso, Rodrigo. *Todo lo que necesitas saber sobre los procesadores ARM.* *hard zone.* [Online] 10 6 2021. <https://hardzone.es/tutoriales/componentes/procesador-arm/>.
73. *Raspberry para novatos. Raspberry Pi 4 Model B 4GB Memoria RAM.* *Raspberry para novatos.* [Online] <https://raspberryparanovatos.com/tienda/raspberry-pi/raspberry-pi-4-model-b-1gb-memoria-ram-2/>.
74. Oracle. *What is a Database? OCI.* [Online] [Cited: 10 8 2022.] <https://www.oracle.com/database/what-is-database/>.
75. Moore, Lindsay. *distributed database. TechTarget.* [Online] 9 2018. [Cited: 10 8 2022.] <https://www.techtarget.com/searchoracle/definition/distributed-database#:~:text=A%20distributed%20database%20is%20a,distributed%20among%20multiple%20atabase%20nodes..>
76. Ionos. *Graph database: bases de datos para una interconexión eficiente. Digital Guide Ionos.* [Online] 30 10 2019. [Cited: 13 8 2022.] <https://www.ionos.es/digitalguide/hosting/cuestiones-tecnicas/graph-database/>.
77. *Historia de la empresa. Tipos de lenguajes de base de datos y sus usos (con ejemplos).* *Historiadelaempresa.com.* [Online] [Cited: 10 8 2022.] <https://historiadelaempresa.com/lenguajes-de-bases-de-datos>.
78. PostgreSQL. *About. PostgreSQL.* [Online] [Cited: 10 8 2022.] <https://www.postgresql.org/about/>.
79. Angel. *¿Qué es y para que sirve Microsoft Access?* [Online] 30 12 2012. [Cited: 10 8 2022.] <https://www.accessyexcel.com/que-es-y-para-que-sirve-microsoft-access/>.
80. Borges, Esteban. *Servidor Base de Datos. infranetworking.* [Online] 17 3 2019. [Cited: 10 8 2022.] <https://blog.infranetworking.com/servidor-base-de-datos/>.
81. MongoDB. *JSON and BSON. MongoDB.* [Online] [Cited: 11 8 2022.] <https://www.mongodb.com/json-and-bson>.
82. Oracle. *¿Qué es una inteligencia artificial? Obtenga más información sobre inteligencia artificial. OCI.*

[Online] <https://www.oracle.com/es/artificial-intelligence/what-is-ai/>.

83. Iberdrola. ¿Somos conscientes de los retos y principales aplicaciones de la Inteligencia artificial? *Iberdrola*. [Online] [Cited: 11 8 2022.] <https://www.iberdrola.com/innovacion/que-es-inteligencia-artificial>.

84. *Mining Surgery Phase-Related Sequential Rules from Vertebroplasty Simulations Traces*. Toussaint, Ben-Manson and Luengo, Vanda. Pavia : s.n., 2015.

85. *Artificial Intelligence applications for thoracic imaging*. Chassagnon, Guillaume, et al. s.l. : Elsevier, 2020, Vol. 123.

86. *An Overview of Artificial Intelligence Applications for Power Electronics*. Zhao, Shuai, Blaabjerg, Frede and Wang, Huai. 4, 2020, Vol. 36.

87. Iberdrola. 'Big data': principales usos y aplicaciones. *Iberdrola*. [Online] [Cited: 11 8 2022.] <https://www.iberdrola.com/innovacion/que-es-para-que-sirve-big-data>.

88. Vignia Lau. How healthbots can assist patients and HCPs. *Vignia Lau*. [Online] 8 1 2018. <https://virginialau.wordpress.com/2018/01/08/how-healthbots-can-assist-patients-and-hcps/>.

89. Pablo. Hay más de una inteligencia artificial. *Orange*. [Online] 13 12 2021. [Cited: 11 8 2022.] <https://blog.orange.es/consejos-y-trucos/tipos-inteligencia-artificial/#>.

90. *Machine Learning Algorithms - A Review*. Mahesh, Batta. 1, 1 2020, International Journal of Science and Research, Vol. 9. 2319-7064.

91. Sanabria Castro, Alexis Andrés. Una introducción a los Árboles de decisión. *Dabia*. [Online] 19 5 2020. <https://www.grupodabia.com/post/2020-05-19-arbol-de-decision/>.

92. Merino, Marcos. Conceptos de inteligencia artificial: qué es el aprendizaje por refuerzo. *Xataka*. [Online] 27 1 2019. <https://www.xataka.com/inteligencia-artificial/conceptos-inteligencia-artificial-que-aprendizaje-refuerzo>.

93. Pascual Estapé , Juan Antonio. Inteligencia artificial: qué es, como funciona y para que se utiliza en la actualidad. *Computer hoy*. [Online] 3 9 2021. [Cited: 12 8 2022.] <https://computerhoy.com/reportajes/tecnologia/inteligencia-artificial-469917>.

94. TIBCO. ¿Qué es una red neuronal? *TIBCO*. [Online] <https://www.tibco.com/es/reference-center/what-is-a-neural-network>.

95. scikit learn. scikit learn. [Online] [Cited: 12 8 2022.] <https://scikit-learn.org/stable/index.html>.

96. —. Linear Models. *scikit learn*. [Online] [Cited: 12 8 2022.] https://scikit-learn.org/stable/modules/linear_model.html.

97. Suárez, Héctor R. ¿Qué es una correlación?... y herramientas de análisis de datos. *incibe-cert*. [Online] 9 7 2015. <https://www.incibe-cert.es/blog/correlacion-herramientas-analisis-datos#:~:text=Es%20una%20t%C3%A9cnica%20de%20an%C3%A1lisis,el%20sentido%20de%20la%20relaci%C3%B3n..>

98. Lobato Ríos, Victor. *Modelo Flexible de Movimiento de Torso, Brazo, Antebrazo y Muñeca*. Coordinación de Ciencias Computacionales, Instituto Nacional de Astrofísica, Óptica y Electrónica. Puebla : INAOE, 2016. p. 153, Tesis.

