



Universidad de Valladolid



ESCUELA DE INGENIERÍAS
INDUSTRIALES

UNIVERSIDAD DE VALLADOLID
ESCUELA DE INGENIERÍAS INDUSTRIALES

Máster en Ingeniería Industrial

Protesis de antebrazo con control mioeléctrico mediante Machine Learning

Autor:

Maroto García, Miguel Ángel

Tutor:

Javier Pérez Turiel

Ingeniería de Sistemas y Automática

Valladolid, Junio de 2020

Resumen

En este trabajo se describe el desarrollo de un prototipo de prótesis activa aplicable a amputados transradiales, usando técnicas y algoritmos de clasificación basados en Aprendizaje Automático (Machine Learning), a partir del análisis y preprocesamiento de datos obtenidas de señales mioeléctricas provenientes de sensores EMG, con potencia y destreza suficiente para la realización de tareas de vida diaria como agarre y manejo de objetos ligeros. Previamente a este desarrollo se realizará un estudio del contexto y Estado del Arte en cada uno de los ámbitos englobados por el actual trabajo, siendo estos sensorización para detección de intencionalidad, sistemas y algoritmos de Machine Learning, actuadores usados en biónica y biomimética y tipos de prótesis diferentes, describiendo los desarrollos más destacados en éste campo. Por último, se realizará un apartado de conclusiones posteriores al desarrollo, así como una sección dedicada a líneas futuras de desarrollo que puedan ramificar del trabajo actual.

Palabras clave

biónica, mioeléctrico, prótesis, machine, learning

Abstract

This work will describe the development of a active prosthesis prototype applicable to transradial amputees, using techniques and classification algorithms based on Machine Learning, from the analysis and preprocessing of data obtain from myoelectric signals acquired from EMG sensors, with enough power and dexterity as to perform daily life activities, such as grabbing and manipulating light objects. Previous to the development, a study of the context and State of the Art will on the aspects englobed by the current work will be performed, these being the sensor systems for intention detection, Machine Learning systems and algorithms, actuators used in bionics and biomimetics and different types of prosthesis, describing the most featured works in this field. Lastly, a chapter will be dedicated to the conclusions dranw from the current work, as well as a section dedicated to future development lines that could ramify from the current work.

Keywords

bionic, myoelectric, prosthesis, machine, learning

Índice

Índice	6
1 Introducción	7
1.1. Contexto	7
1.2. Objetivos	8
1.3. Hitos y resultados	9
2 Estado del arte	13
2.1. Diseños mecánicos	13
2.2. Sensórica	21
2.3. Sistemas de aprendizaje automático (Machine Learning)	26
3 Desarrollo	31
3.1. Interfaz mioeléctrica	33
3.2. Controlador Machine Learning (Clasificador de movimientos)	44
3.3. Prótesis electromecánica	56
3.4. Controlador del brazo	65
3.5. Desarrollo software “arm_prosthesis”	75
4 Conclusiones	79
5 Lineas futuras	81
Bibliografía	83
A Anexos	89
A.1. BOMs	89
A.2. Estudio económico	90
A.3. Esquemas electrónicos	93

Capítulo 1

Introducción

En éste primer capítulo se hará un breve resumen del trabajo, exponiendo el contexto del mismo, describiendo los objetivos a cumplir y los resultados esperados, divididos en hitos.

1.1. Contexto

En el marco de éste trabajo, siempre que se hable de prótesis fuera de un contexto de definición del tipo de diseño o sin especificaciones adicionales, nos referiremos a prótesis activas para amputados transradiales. Sólo en Estados Unidos, se realizan unas 50.000 amputaciones, de las cuales 1/4 de ellas son de miembros superiores, y las amputaciones transradiales representan el 60 % de los tipos de amputaciones de extremidades, sumando un número aproximado de 7500 personas al año en EEUU [11], y debido a la naturaleza del miembro amputado y la destreza del mismo, inhabilitan de forma considerable al afectado y le privan de la capacidad de realización de tareas de vida diaria.

En la actualidad existen diversos dispositivos para suplir las necesidades de personas en situación de discapacidad debido a amputaciones de extremidades, prácticamente para todos los niveles existentes. Dichas prótesis, según la incorporación o no de sensorización y motorización podrían clasificarse en activas o pasivas. Las prótesis pasivas han proliferado en los últimos años, con costes inferiores y alta personalización, lo que permite una gran accesibilidad a las mismas, en parte gracias a nuevas técnicas de visión artificial y fabricación aditiva.

Respecto a las prótesis activas, existe una gran cantidad de desarrollos, la mayo-

ría orientados a fabricación de prototipos con fines de investigación, pero existen productos establecidos, destacando por su impacto mediático el brazo “Hero” de la startup situada en Bristol, Open Bionics [3], el cual ofrece a amputados transradiales capacidad de agarre y movimiento de mano y dedos, a partir de señales mioeléctricas de la musculatura restante en el antebrazo. Otros desarrollos implementan diferentes tecnologías, en el ámbito mecánico, de control y de sensorización, con el objetivo de obtener movimientos más precisos, métodos menos invasivos, prótesis más accesibles, sensación de tacto e incluso destreza.

En este trabajo se desea establecer un punto de partida mediante el desarrollo de un prototipo con tecnologías y métodos acotados, con el objetivo de allanar el camino para futuras incursiones y profundización en los diferentes aspectos y campos que constituyen éste tipo de prótesis.

1.2. Objetivos

El principal objetivo es el desarrollo de un prototipo de prótesis de brazo activa para amputaciones de nivel transradial, mediante la obtención de señales mioeléctricas de los flexores y pronadores del antebrazo. Con el objetivo de desglosar el trabajo, se definen los siguientes objetivos específicos:

- Desarrollo de un brazo mecánico activo adaptable a amputados de nivel transradial: se busca la construcción de un dispositivo electromecánico suficientemente diestro y potente, que sustituya la mano de un amputado transradial, y que permita la realización de actividades diarias mediante posicionamientos específicos de la misma (mano en puño, apuntar, etc.). Se considerará como destreza suficiente la capacidad de mover cada dedo independientemente, y potencia suficiente la capacidad de agarre de objetos ligeros de formas variadas (lata, bolígrafo, vaso, cubiertos, teléfono móvil, etc)
- Desarrollo de un sistema de captación de señales mioeléctricas: se busca el desarrollo de un conjunto de dispositivo y software que permita la obtención de datos que permitan intuir la intencionalidad del usuario de la prótesis a partir de los músculos del antebrazo, mediante el uso de electromiografía y el software necesario para el procesamiento de las señales en bruto, de

tal manera que tengamos un conjunto de datos razonablemente pequeño y significativo para su uso en la predicción de movimientos de la prótesis

- Desarrollo de un sistema de control y clasificación de movimientos mediante el uso de técnicas de aprendizaje automático (Machine Learning): se busca el desarrollo de un sistema que, mediante algoritmos basados en técnicas de Machine Learning, y a partir del uso de unos datos obtenidos periódicamente del sistema de captación de señales, permita la clasificación de unos movimientos, previamente grabados y etiquetados correctamente, de tal manera que después de entrenar al algoritmo, este sea capaz de predecir la posición intencionada del usuario y devolver un valor representativo de la misma

Cumplidos todos los objetivos y usados los desarrollos mencionados en conjunto, se espera obtener un prototipo funcional de una prótesis de mano para amputados transradiales capaz de sustituir el miembro perdido y que permita la realización de tareas de vida diaria, de una forma suficientemente diestra, es decir, que el número de posiciones diferentes de la mano sea útil para muchos tipos de tareas.

1.3. Hitos y resultados

El desarrollo de este trabajo se dividirá en los siguientes hitos, detallando los resultados esperados y desglosando los objetivos mencionados anteriormente:

1. Estado del arte: revisión del contexto de los campos que engloba la prótesis y descripción de los últimos desarrollos en cada uno
 - a) Diseños mecánicos: se revisarán los diferentes actuadores disponibles, sus principios de funcionamiento, fuerza, biomimeticidad, etc. Una vez establecido el contexto se hablará de los desarrollos actuales más destacados en el ámbito de la biónica y las prótesis transradiales
 - b) Sensórica: se hará una breve contextualización de los sensores comúnmente usados en la predicción de intencionalidad de un sujeto, a la vez que se revisarán desarrollos actuales que involucren el tipo de sensor descrito
 - c) Sistemas de aprendizaje automático: se realizará una contextualización general sobre la definición de un sistema Machine Learning, los tipos de algoritmos que existen, para una posterior enumeración de los algoritmos más usados en clasificación de datos de intencionalidad de un sujeto para la obtención de predicciones útiles en prótesis

2. Desarrollo de interfaz mioeléctrica: descripción del desarrollo de una interfaz para la obtención de un conjunto de datos en tiempo real a partir de señales mioeléctricas
 - a) Sensor seleccionado: a partir de los dispositivos descritos en el estado del arte, se describirá la selección del sistema usado en el trabajo actual, argumentando la decisión y describiendo ventajas y desventajas del dispositivo
 - b) Obtención de señales: se procederá a realizar la descripción del desarrollo del método de obtención de señales electromiográficas en bruto
 - c) Obtención de parámetros de las señales: a partir de las señales en bruto, se desarrollarán los sistemas software basados en modelos matemáticos de obtención de características de gráficas, que permitirán la obtención de un conjunto de datos reducido y significativo
 - d) Implementación de la interfaz: se describirá la implementación del sistema software desarrollado, desglosando cada parte del módulo de sensorización completo
3. Controlador Machine Learning (Clasificador de movimientos): descripción del desarrollo de un módulo software de predicción de posiciones a partir de los datos de intencionalidad del sujeto
 - a) Algoritmo seleccionado: a partir de los algoritmos más usados descritos en el Estado del Arte, se seleccionará uno para su uso en el trabajo actual, argumentando la decisión y describiendo sus ventajas y desventajas
 - b) Plataforma seleccionada: entre todas las plataformas software en las que se pueda implementar el algoritmo Machine Learning seleccionados, se seleccionará una plataforma o librería que permita su implementación y entrenamiento de manera sencilla y eficiente
 - c) Implementación del controlador: se describirá la implementación del algoritmo, describiendo los datos que se alimentarán al sistema y las salidas esperadas, así como la estructura del mismo, los parámetros de entrenamiento usados y los resultados de precisión, pérdida, etc. obtenidos
4. Prótesis electromecánica: descripción del desarrollo de un diseño mecánico de mano protésica adaptable a amputados transradiales

- a) Selección del diseño mecánico: a partir del estudio del Estado del Arte realizado, se seleccionará un dispositivo existente, comercial o manufacturable, el cual cumpla con las condiciones de destreza y potencia necesarias
 - b) Fabricación, montaje y prueba del brazo: a partir del diseño elegido, se fabricará y montarán los sistemas necesarios, y se probarán en conjunto antes de acoplarse al sistema de prótesis completo
5. Controlador del brazo: descripción del desarrollo del sistema de control de la prótesis
- a) Algoritmo seleccionado: se seleccionará un algoritmo de control en consideración con el tipo de diseño mecánico seleccionado, así como los actuadores que lo componen
 - b) Plataforma seleccionada: se seleccionará un sistema hardware lo más sencillo posible que sea capaz de cumplir las funciones del algoritmo seleccionado
 - c) Implementación del controlador: se describirá la implementación del algoritmo seleccionado en la plataforma usada, usando para ello las herramientas que proporcione la misma (lenguaje, IDE, etc.)
6. Conclusiones y líneas futuras: posterior al completo desarrollo del trabajo, se hará una recopilación de las ideas principales del trabajo, ya sean conclusiones sacadas de los desarrollos o problemas surgidos, así como un capítulo adicional describiendo líneas futuras para desarrollos que puedan ramificar del trabajo actual



Figura 1.1 : desarrollo completo del prototipo propuesto en este trabajo

Capítulo 2

Estado del arte

En éste capítulo se procederá a descripción del actual estado del arte, haciendo un breve recorrido de las materias, dispositivos e investigaciones concernientes al marco de éste trabajo, clasificando los desarrollos expuestos como ejemplos dentro de clasificaciones propuestas en el trabajo, con el objetivo de organizar el Estado del Arte de una manera lógica al desarrollo de las partes del prototipo propuesto en este trabajo.

2.1. Diseños mecánicos

La parte más importante de una prótesis, en cuanto a su funcionalidad, se define en el diseño mecánico, ya que el resto de elementos adicionales proveen al diseño de potencial adicional para la adición de habilidades al usuario de la misma. Hay una enorme variedad de diseños y enfoques en prótesis, así como una infinidad de implementaciones de los mismos dependiendo del tipo de amputación con la que se esté trabajando, ya sea del tren superior o inferior.

Prótesis pasivas

Constituyen un tipo de prótesis caracterizadas por la carencia de actuadores que asistan al movimiento de la misma, de tal manera que su funcionalidad viene provista por mecanismos que aprovechan la movilidad disponible del usuario, y la transforman en otro tipo de acción que, de otra manera, el sujeto no podría realizar debido a su condición física. Éste tipo de diseños forman alrededor de 1/3 de las manos protésicas en uso y que se aplican en pacientes, pero son a su vez las más descuidadas en cuanto a nuevos desarrollos [29].

Considerando los diseños más comunes y basándonos en su apariencia, complejidad y funcionalidades, haremos una somera distinción de los tipos de prótesis pasivas existentes:

- **Prótesis sin funcionalidad mecánica:** nos referimos a los diseños puramente estéticos, sin finalidades de suplir habilidad al sujeto. La mayoría consisten en una simple funda con estética más o menos parecida a un brazo humano, amoldada a la forma de la amputación. Un ejemplo de este tipo de prótesis son las manos hechas a medida o los cubre-brazo de silicona fabricados por la compañía Ottobock [40], entre otros tantos ejemplos de tiendas de ortopedia.



Figura 2.1 : mano personalizada y cubre-brazo de silicona, extraído de Ottobock

Cabe destacar que, además de éstas prótesis completamente fijas, se encuentran algunos diseños que, aunque mantienen la posición, ésta es ajustable, brindando un ligero grado de funcionalidad por encima del enfoque cosmético [21].

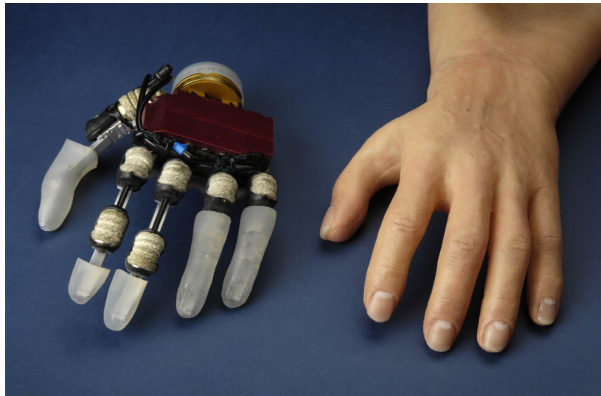
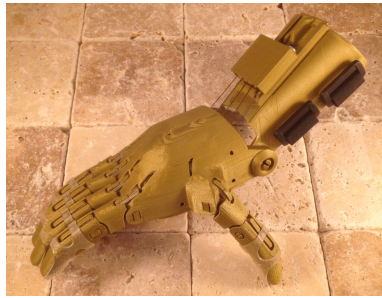
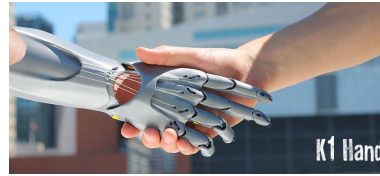


Figura 2.2 : mano protésica cosmética multifuncional, extraído de IEEE

- Prótesis con funcionalidad mecánica: estos diseños tienen como objetivo suplir al usuario con habilidades adicionales perdidas debido a la amputación transradial. Según criterios de complejidad y fabricación, podemos diferenciar dos tipos:
 - Prótesis convencionales: en este grupo se encontrarían prótesis ortopédicas comunes, como ganchos, ya sean actuados mediante el movimiento del codo o mediante la extremidad superior funcional, o prótesis con herramientas configurables, como apuntadores, herramientas de sujeción, etc.
 - Prótesis avanzadas: la diferencia principal con el grupo anterior se encuentra en las tecnologías de fabricación utilizadas; uso de visión artificial para la adaptación de la prótesis, y fabricación aditiva para la consecución de la misma, permitiendo una muy alta personalización y flexibilidad reduciendo los costes de la ortopedia más convencional. Un ejemplo destacado por las ventajas mencionadas son las prótesis pasivas eNable [13], proyecto que ofrece una gran variedad de diseños, imprimibles en 3D, baratos de fabricar y escalables, por lo que pueden ser aplicados a sujetos de cualquier edad.



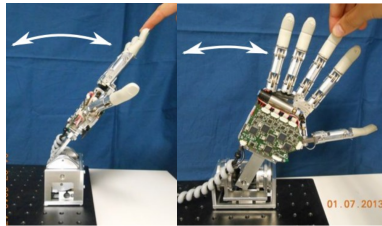
(a)



(b)

Figura 2.3 : mano Phoenix v2 (a) y K1 (b), dos de la más de decena de diseños publicados por el proyecto e-Nable, extraído de e-Nable

Algunos de estos ejemplos muestran una gran ajustabilidad, y una alta fidelidad en la fisionomía interna de la mano humana [33] [34].



(a)



(b)

Figura 2.4 : desarrollos de 2013 y 2017 respectivamente sobre un diseño de mano pasiva con rigidez ajustable, extraído de IEEE

Prótesis activas

Prótesis activas son aquellas que, mediante actuadores y predicción de la intencionalidad del sujeto, son capaces de asistir al movimiento del usuario, brindando capacidades más cercanas a la destreza de un brazo humano.

Para proveer de dicha asistencia al movimiento, deben de estar provistos de actuadores mecánicos que sean capaces de mover la prótesis. Atendiendo al tipo de movimiento que realizan y el principio físico del que se aprovechan, distinguiremos los siguientes dispositivos:

- **Motores convencionales:** en este caso nos referiremos a motores DC, ya sean con escobillas o sin escobillas, generalmente acompañados de reductoras y sistemas de tensores y cabrestantes para transmitir el movimiento a la prótesis.

- Motores lineales: dispositivos electromagnéticos en forma tubular, capaces de mover un eje cilíndrico a lo largo de su recorrido [28]. Pueden usarse tanto como para accionar directamente la mano o usar sistemas de transmisión con tensores o con enlaces mecánicos rígidos [10].



Figura 2.5 : motor lineal corto, extraído de LinMot

- Neumática:
 - Neumática convencional: cilindros actuados mediante aire comprimido. Fáciles de controlar y con cierta suavidad dependiendo de la presión y velocidad de actuación usadas, pero necesitan grandes y pesados sistemas adicionales para generar el aire comprimido y almacenarlo para su uso.
 - Neumática biomimética: actuadores biomiméticos accionados por aire comprimido. En este tipo de actuadores destacan los músculos neumáticos de Festo, un producto biónico comercial.



(a)



(b)

Figura 2.6 : músculos neumáticos biomiméticos Festo, extraído de Festo

- Hidráulica: cilindros actuados mediante fluidos incompresibles. Suelen tener una fuerza muy superior al resto de los actuadores mencionados en esta sección, pero son grandes y pesados, y necesitan de sistemas auxiliares que provean de presión al sistema.
- Electropolímeros: polímeros los cuales, al aplicarle una corriente eléctrica, son capaces de producir movimiento [2].

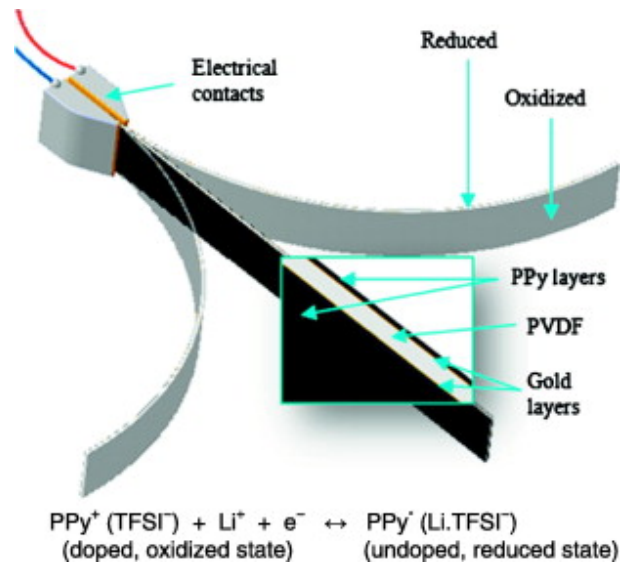


Figura 2.7 : actuador Electropolimérico, extraído de Elsevier [35]

Su fuerza y usabilidad todavía están cuestionadas, y en fase de desarrollo, por lo que existen muy pocos diseños que se nutran de éste tipo de tecnología.

Éstas prótesis suelen tener un diseño más complejo que las pasivas, incorporando más grados de libertad, sistemas mecánicos que transmitan el movimiento de los actuadores y espacios en la misma para la inserción de la amputación, fuentes de autonomía, sistemas de control, etc. Atendiendo a la complejidad de los movimientos que puedan realizar éstas prótesis, diferenciaremos dos tipos principales de diseños:

- **Diseños funcionales:** nos referiremos como diseños funcionales a aquellos cuyo objetivo tiene la realización de una o varias tareas de vida diaria concretas (agarrar, mano abierta para saludar, dedos individuales extendidos para teclear, etc). Actualmente, el uso de estos diseños es bastante extendido, con varias soluciones comerciales disponibles. Estos diseños se caracterizan por:
 - Sencillez en el diseño mecánico: con pocos grados de libertad, a menudo pueden tomar formas diferentes a una mano humana, ej. un simple gancho con un motor y un cabrestante para agarre de objetos.
 - Movimientos básicos: el objetivo de estos diseños es proveer al usuario la capacidad de realizar actividades sencillas de vida diaria, como agarrar y apuntar.

- Robustez: con el objetivo de proveer una fiabilidad suficiente para actividades de vida diaria, estos diseños suelen estar provistos de estructuras resistentes y actuadores suficientemente potentes para actividades de agarre y manipulación de objetos.
- Autonomía extendida: debido a que la aplicación de estos diseños se centra en actividades de vida diaria, éstas prótesis viene provistas de baterías con una capacidad suficiente para proveer energía durante periodos extendidos de uso, con duraciones empezando por varias horas, pasando por un día entero, extendiéndose hasta varios días en algunos diseños.

Entre este tipo de prótesis destaca, por su certificación para uso médico e impacto mediático, las manos protésicas de Open Bionics [3], con su Hero Arm, un brazo protésico con varias posiciones, detección de intencionalidad y tamaño y aspecto personalizable.



Figura 2.8 : prótesis de antebrazo Hero Arm, extraído de Open Bionics

- Diseños biomiméticos: en ésta categoría abarcaremos los diseños más enfocados a la consecución de prótesis con mayor destreza, las cuales se centren en obtener una movilidad similar a la humana. Por lo general, estos diseños están relegados a ámbitos de investigación, con muy pocas unidades existentes o prototipos únicos para el diseño, y suelen ser muy complejos de llevar a un mercado amplio debido al coste, dificultad de implementación o especificidad de la prótesis respecto al sujeto. Estos diseños se caracterizan por:

- Complejidad: amplio número de grados de libertad (a menudo mayor de 10), permitiendo el posicionamiento individual de dedos y falanges, movimientos de flexo-extensión de dedos, etc.
- Destreza: la complejidad de estos diseños permite movimientos más diestros, agarres de objetos con formas extrañas, posicionamiento preciso, etc, para actividades diestras como escribir, realizar gestos complejos, etc.
- Menor robustez: aunque esto depende en gran medida de los materiales, éstas prótesis no suelen estar enfocadas a un uso continuo que pudiera someterlas a desgaste.
- Menor autonomía: debido al número de actuadores que requiere la mayor complejidad y el límite de espacio en la prótesis impuesto por los elementos mecánicos de la prótesis, la autonomía se ve relegada a un uso conectado a una fuente conmutada de potencia, o baterías externas, las cuales no proveen de suficiente energía para un uso continuado.

Algunos ejemplos pueden ser la mano ACT [51], con 15 publicaciones relacionadas y la mano biomimética con servos Dynamixel [52]

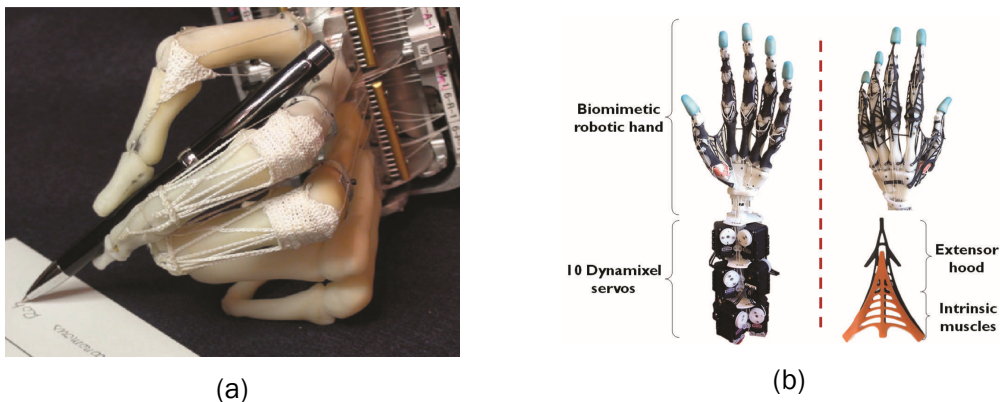


Figura 2.9 : mano ACT (a), extraído de IEEE y mano biomimética con servos Dynamixel, extraído de ICRA

2.2. Sensórica

La parte que más afecta a la fiabilidad de una prótesis, en cuanto a la certeza y destreza de sus movimientos, concierne a los sensores utilizados. Existe un amplio abanico de sensores y de tecnologías usadas en biónica, desde sensores electromiográficos hasta sensores de ultrasonidos. Debido al vasto alcance de este apartado, nos centraremos en describir los primeros: sensores EMG; estos son ampliamente usados en biónica, ya que ofrecen de manera intuitiva un conocimiento sobre el estado de activación de un músculo, de forma relativamente sencilla y barata.

La electromiografía o EMG consiste en la detección del voltaje de un músculo debido a la despolarización de las células musculares, ordenadas por el sistema nervioso [45]. Usando electrodos para medir este potencial, y sistemas de amplificación de la señal y reducción del ruido, podemos obtener datos grafados sobre el estado de activación de un músculo, el cual puede ser usado para interpretar su movimiento, principio aplicado a su uso en prótesis biónicas. Una aplicación más conocida de la medición de campos eléctricos del cuerpo humano es el Electrocardiograma (ECG), el cual mide la actividad eléctrica del corazón y permite observar, diagnosticar y monitorizar, mediante interpretación de las gráficas, el corazón humano [12].

Con el objetivo de diferenciar entre la variedad de dispositivos EMG en el mercado y en investigación, haremos básicamente 2 clasificaciones: según la invasividad (dependerá del sensor), y según su velocidad de muestreo (tendrá más relación con el sistema de muestreo y amplificación). Adicionalmente, realizaremos una clasificación más: según si la transmisión de los datos es mediante cableado o inalámbrica, la cual será útil para enmarcar la selección de un sensor que se pueda aplicar al desarrollo de este trabajo.

Según la invasividad podremos diferenciar los siguientes tipos:

- De contacto: de aplicación superficial. Se basa en el uso de dos electrodos de referencia y un electrodo central que detecta la desviación de la señal emitida por los electrodos de referencia debido a actividad muscular.

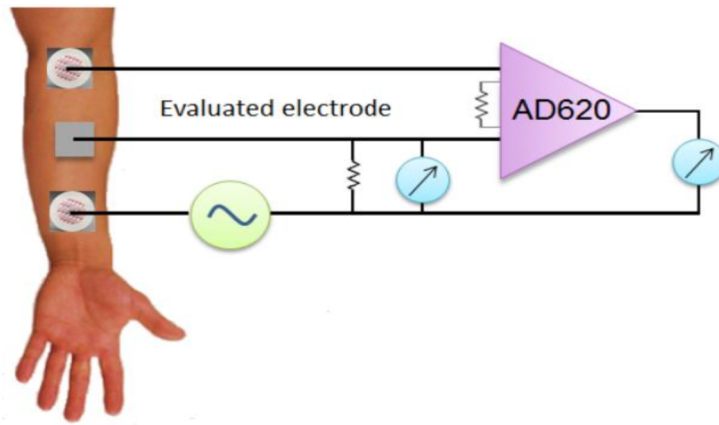


Figura 2.10 : esquema de funcionamiento de un electrodo EMG superficial, extraído de ReseachGate [46]

Dentro de las ventajas destaca que no es invasivo, por lo que no supone ninguna molestia o dolor para el paciente. Su desventaja principal es que no es tan preciso como un sensor invasivo (debido a la fisionomía del paciente) y sólo puede detectarse la actividad de músculos superficiales. Dentro de este tipo de sensor se pueden diferenciar dos variantes:

- Secos: el material conductor del sensor se pone en contacto directo con la piel, sin ningún material que sirva como interfaz para promover la conductividad. Son muy cómodos de quitar y poner, pero poseen la desventaja de menor fiabilidad por las características de la piel (pelo, grasa, suciedad, etc) [46].

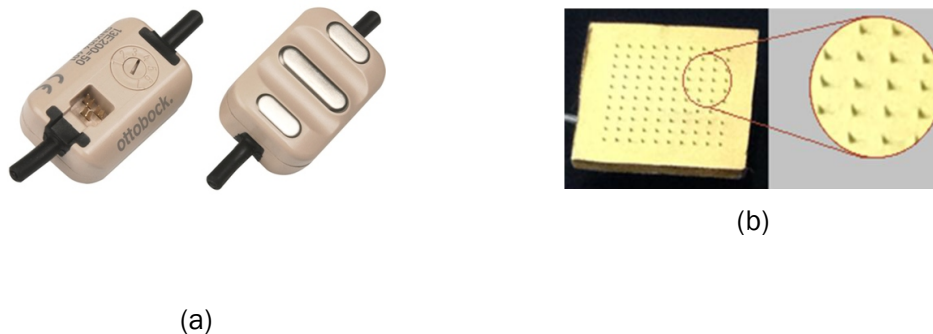


Figura 2.11 : electrodo superficial Ottobock 13E200 (a) y electrodo superficial con microagujas (b), extraído de Ottobock y ResearchGate respectivamente

- Húmedos: el material conductor del sensor no está en contacto directo con la piel, sino que se conecta a la piel mediante una interfaz material de gel conductor, como los geles basados en cloruro de plata. Suelen comercializarse en forma de pads adhesivos con la interfaz húmeda en el medio, en contacto con un conductor metálico accesible por la parte no adhesiva del pad [31].



Figura 2.12 : pad EMG húmedo, extraído de Nissha Medical

- Invasivos: en su forma más común, consisten en un electrodo monopolar en forma de aguja que se inserta en el músculo cuya actividad se desea medir.



Figura 2.13 : electrodo EMG invasivo de aguja, extraído de CNSac [32]

Nuevos desarrollos permiten electrodos invasivos insertables quirúrgicamente y accesibles y alimentados mediante un transmisor magnético transcutáneo, de tal manera que se obtiene la precisión de un electrodo invasivo y la comodidad de un electrodo superficial, con el coste adicional de la operación de inserción de los electrodos [14].

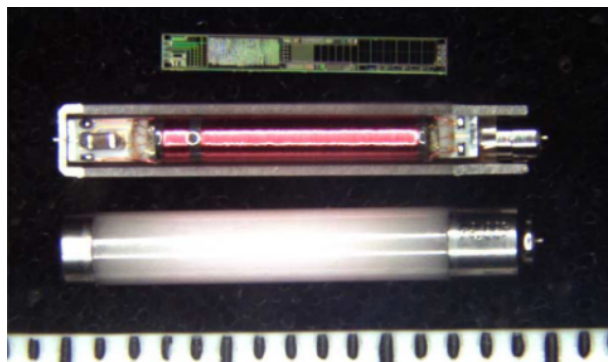


Figura 2.14 : Sensor Mioeléctrico Implantable (IMES), referencia de 1mm, extraído de NIH

Según la velocidad de muestreo, nos moveremos en tres rangos diferenciados:

- $\sim 200\text{Hz}$: sensores con poco ancho de banda y baja precisión, proveen de una estimación no muy fiable del nivel de activación de un músculo, ya que hay partes de la onda EMG que no son detectables al tener componentes mayores de 1000Hz [50].
- $\sim 1000\text{-}4000\text{Hz}$: rango más común a la hora de medir gráficas EMG, ya que, a partir de 1000Hz , se supera la frecuencia de Nyquist de la mayoría de las componentes de un EMG [41].
- $\gg 4000\text{ Hz}$: destinados a aplicaciones de investigación, con una captura y amplificación de la señal potente y cara, usualmente de gran tamaño debido a la electrónica necesaria para el procesado. Serán descartados para su uso en el marco del actual trabajo.

Por último, haremos mención a sensores que puedan diferenciarse por su cableado:

- Dispositivos cableados: electrodos cableados y comunicación con el sistema de captura y amplificación mediante sistemas cableados.

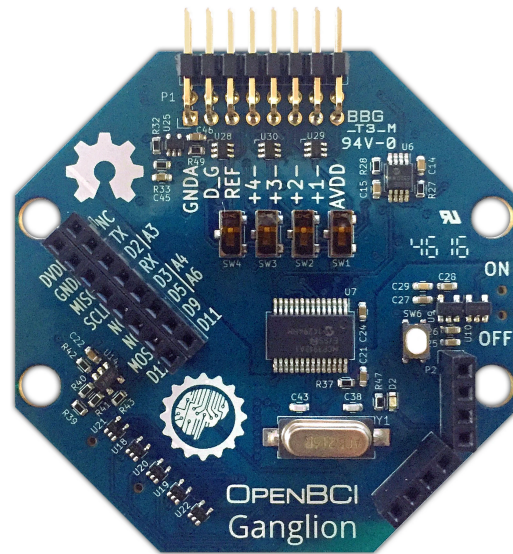


Figura 2.15 : placa de captura de EMG Ganglion Board, extraído de OpenBCI [39]

- Dispositivos inalámbricos: todo el sistema de electrodo, captura y amplificación se encuentra en un paquete, y es capaz de comunicar los datos graficados del EMG mediante conectividad inalámbrica.



Figura 2.16 : sensor de antebrazo con EMG e IMU Myo Armband, extraído de Thalmic Labs [24]

2.3. Sistemas de aprendizaje automático (Machine Learning)

Machine Learning (ML), aprendizaje máquina o Aprendizaje Automático (AA) consiste en una serie de técnicas de Inteligencia Artificial con el objetivo de realizar predicciones a partir de datos alimentados, entrando un modelo matemático determinado de forma supervisada o sin supervisar. El enfoque de este paradigma de implementación de predictores se basa en el uso de datos como fuente de conocimiento sobre las etiquetas a predecir, en lugar del conocimiento del sistema en sí, de tal manera que, teniendo un conjunto de datos de entrada y conociendo las características de los mismos, se implementa un modelo de aprendizaje automático que se adapte correctamente a los mismos, aprendiendo mediante funciones de pérdida (en el caso del entrenamiento supervisado) y mejorando con cada iteración de entrenamiento, dando lugar a un sistema que obtiene resultados con una precisión determinada, el cual no llega nunca a ser determinista. Debido al gran alcance de la temática, nos limitaremos a mencionar tipos de modelos con casos de éxito y uso en el ámbito de uso de clasificación de EMG.

El enfoque principal del ML es el hecho de que los modelos aprenden y pueden ser entrenados, característica mediante la cual se puede construir una clasificación de los principales tipos de sistemas ML:

- Aprendizaje supervisado: consiste en alimentar el modelo con datos de entrada etiquetados con los datos de salida correspondientes, de tal manera que el modelo pueda comprobar si las predicciones que ha realizado son correctas, y en caso contrario corregirlo. Esto se realiza mediante el uso de funciones de pérdida que evalúan si el modelo realiza una predicción cercana a las etiquetas proporcionadas
- Aprendizaje no supervisado: el modelo sólo es entrenado con datos de entrada sin etiquetar, por lo que no tiene referencia sobre lo que ha de obtener, de tal manera que obtenga patrones de los datos de entrada
- Aprendizaje semisupervisado: el modelo puede ser entrenado con datos etiquetados y sin etiquetar. Término medio a los anteriores dos enfoques

Dados los tipos de aprendizaje mencionados, éstos pueden ser implementados en una gran variedad de tipos de modelos matemáticos que sirven como técnicas de clasificación aplicables a ML:

- Regresores lineales: en su forma más sencilla ajustan el resultado de la entrada al resultado estimado mediante una recta con una pendiente y offset determinado [18].

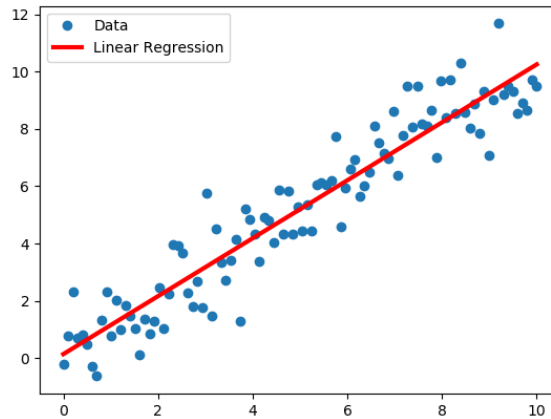


Figura 2.17 : regresor lineal con una entrada y una salida, extraído de medium.com

Este enfoque puede expandirse a una gran cantidad de entradas, asignando pesos diferentes a cada una de las entradas

- Redes neuronales: se basan en el uso de unidades computacionales (neuronas artificiales) que, mediante un peso asignado y una función de activación determinada, se interconectan con otras neuronas artificiales en topología en capas, para lograr construir un sistema capaz de representar diferentes salidas y no linealidades que presenten los datos, sin necesidad de realizar una gran tarea previa de combinación de atributos [18].

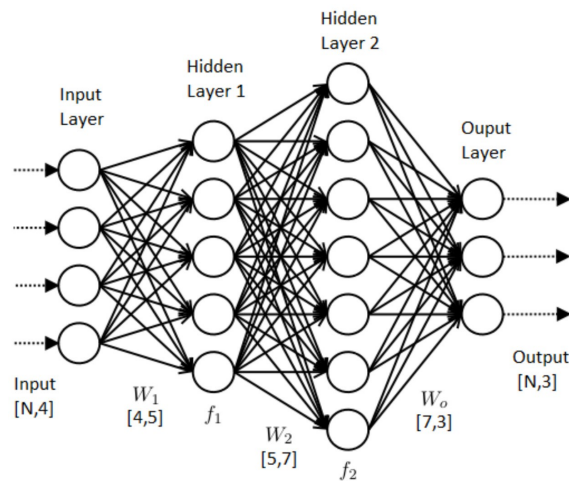


Figura 2.18 : red neuronal con dos capas ocultas y MIMO, extraído de medium.com

- Máquinas de vectores de soporte (SVM): se basa en el uso de vectores en un espacio dado por los datos de entrada, creando planos de separación entre los datos y clasificándolos según cumplan con unas características o con otras, buscando un hiperplano que permita una clasificación exacta de los puntos en el espacio [37].

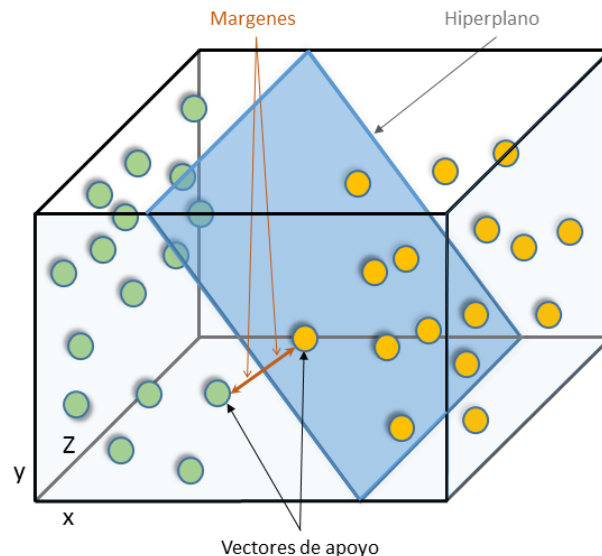


Figura 2.19 : Support Vector Machine en un espacio de tres dimensiones, extraído de numerentur.org

- Algoritmos genéticos: comúnmente usados en búsqueda y optimización, se basan en procesos genéticos naturales, en los cuales se tiene una población inicial de valores para una función los cuales son analizados y evaluados, seleccionando los que más se acercan al resultado objetivo, para después recombinarlos y mutarlos, produciendo así una población nueva y mejor entrenada que sustituye a la anterior, realizando este proceso varias veces para obtener el mejor individuo de la última población, función la cual será capaz de estimar las salidas con la mejor precisión posible [17].

Los modelos más comúnmente usados en el campo de la predicción de la intencionalidad de un sujeto son los clasificadores multiclase, de diferentes topologías y con diferentes tipos de datos, con tipo de aprendizaje supervisado, es decir, los datos están asociados a la etiqueta que representan. Algunos de los clasificadores más utilizados y vistos en trabajos publicados son los siguientes:

- **Perceptrón Multicapa:** consiste en una capa de entrada, una o varias capas ocultas completamente interconectadas y una de salida, con valores regresivos [19].

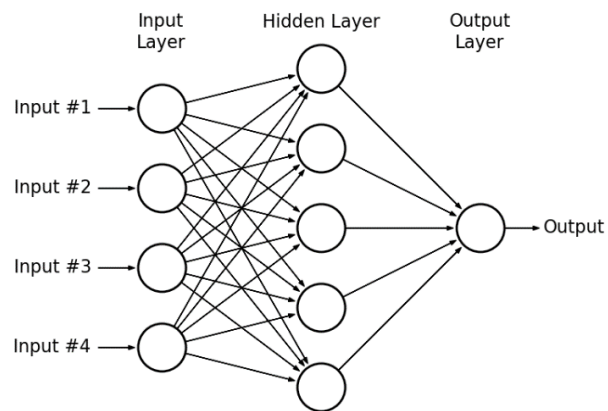


Figura 2.20 : grafo topológico de un Perceptrón Multicapa, extraído de ReseachGate

- **Clasificador SVM:** *Support Vector Machine*, cuya idea consiste en la separación de los conjuntos de datos mediante vectores lineales [53].

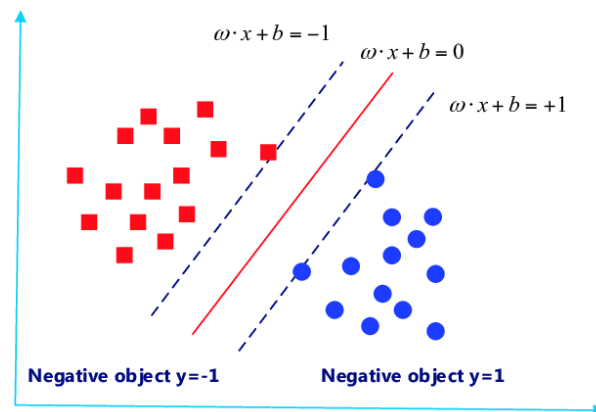


Figura 2.21 : clasificador SVM con 2 clasificaciones, extraído de ReseachGate

- Redes Neuronales Profundas Multiclase: se trata de redes de unidades de cálculo con entrada, peso y función de activación, pueden estar interconectadas completamente, y tienen una capa de salida multiclase que estima la probabilidad de cada una de las posibles etiquetas. El Perceptrón Multicapa es un tipo de red neuronal, mediante la añadidura de la capa Softmax en la salida y funciones de activación Tanh o ReLu, permite clasificar las entradas en diferentes clases.

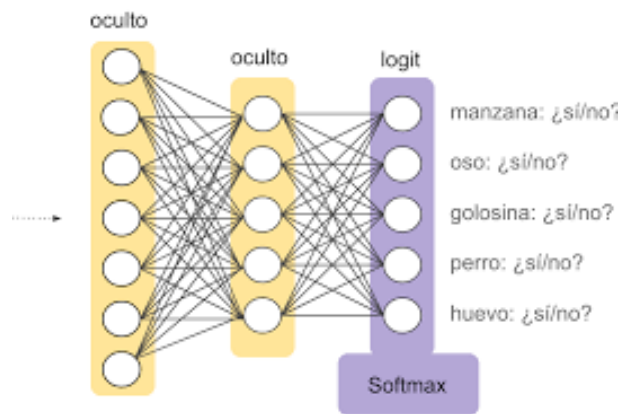


Figura 2.22 : topología de una Red Neuronal Multiclase, extraído de Google

Capítulo 3

Desarrollo

En este capítulo se procederá a la descripción del desarrollo de la solución completa de una prótesis para amputados transradiales, desde la sensorización hasta el dispositivo mecánico.

El objetivo de éste trabajo es la construcción y programación de un dispositivo completo, haciendo uso de desarrollos existentes con el objeto de construir una prótesis barata, fiable y razonablemente diestra, que pueda ser usada por pacientes con amputación transradial y que sea portable, lo que requerirá de sistemas embebidos, pequeños y ligeros, con fuentes de energía autónomas y portátiles.

Para el desarrollo de todo el software que integrará la solución se ha decidido usar el lenguaje de programación Python, el cual nos dará las siguientes ventajas:

- Multiplataforma: nos permitirá ejecutar el software de la solución en equipos diferentes, incluso con sistemas operativos diferentes
- Interpretado: permite la ejecución línea por línea, sin necesidad de compilar el programa
- Amplia gama de librerías: nos da la ventaja de usar librerías realizadas por la comunidad para cubrir necesidades como podrían ser las interfaces BLE y serial, así como otras necesidades asociadas a la extracción de características, etc.
- Sencillez: Python se caracteriza por una rápida curva de aprendizaje, facilidad de uso y legibilidad

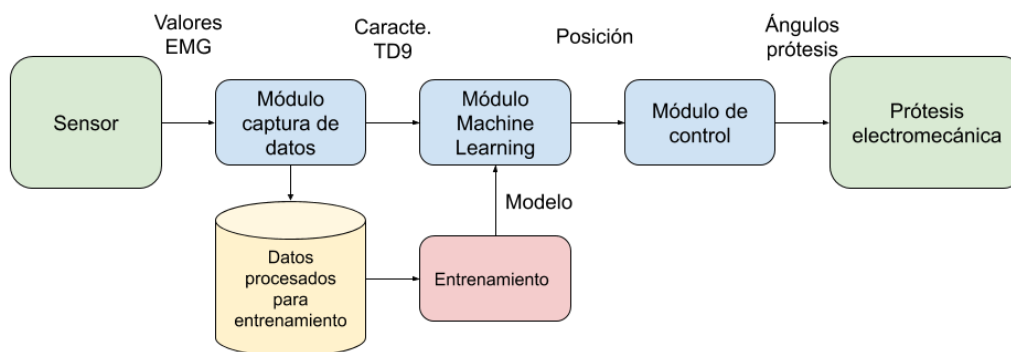


Figura 3.1 : diagrama de bloques de la solución completa

El sistema completo consta de los siguientes módulos principales:

- **Sensor:** representa el dispositivo que se usará para captar las señales electromiográficas del antebrazo, de el se obtiene la gráfica EMG en bruto, la cual se alimenta al módulo de captura de datos
- **Módulo de captura de datos:** se encarga de recibir los datos en bruto del sensor, procesarlos y obtener un conjunto de datos significativo a partir de las gráficas EMG. Estos datos son en forma de características de la gráfica de EMG, como RMS, variancia, etc.
- **Módulo Machine Learning:** procesa los datos provenientes del módulo de captura de datos en tiempo real y predice un número correspondiente a la posición estimada. Para ello necesita un modelo previamente entrenado a partir de datos procesados grabados
- **Módulo de control:** traduce el número de posición a los ángulos de la prótesis electromecánica
- **Prótesis electromecánica:** diseño mecánico capaz de simular el movimiento de una mano humana, adaptable a amputados transradiales. Usa los ángulos recibidos por el control y los transforma en los movimientos correspondientes de los actuadores

Cabe destacar que la barrera que divide alguno de los módulos es ficticia, ya que en los programas definitivos no hay una distinción completa entre ellos y se integran completamente en scripts individuales, como podría ser la división entre el Módulo Machine Learning y el de Control, así como algunos módulos están compuestos a su vez por otros módulos completamente diferenciables, como en el módulo de

sensorización, el cual puede ser dividido en captación de datos en bruto y post-procesado.

3.1. Interfaz mioelétrica

El primer bloque a desarrollar en el orden del trabajo corresponde a la sensorización. En esta sección se describirá el sensor seleccionado y los criterios y alcance establecidos, la obtención de las señales deseadas, el procesamiento de las mismas y la implementación del bloque en el código de la solución.

Sensor seleccionado

Debido al alcance puramente académico del trabajo, se ha seleccionado una sensorización y procesamiento sencillos, sin entrar en toda la profundidad y estudio que requiere una comprensión detallada de la electromiografía y los implicaciones de uso de electrodos independientes y sistemas de captación y amplificación de señal. Por tanto, los requerimientos establecidos para acotar el alcance de la sensorización en este trabajo serán los siguientes:

- Solución completa: deberá incluir tanto los electrodos como el sistema de captación y amplificación
- Digitalización de las señales para su tratamiento y procesamiento
- Comunicaciones: el sensor deberá tener comunicaciones integradas para la obtención de las señales a través de una API. No se impondrán requerimientos en cuanto a la conectividad necesaria

Dentro de todos los dispositivos conocidos y ya descritos o mencionados en el Estado del Arte realizado en este trabajo, se seleccionará la pulsera sensorizada Myo Armband.



Figura 3.2 : foto de pulsera sensorizada Myo Armband

Ésta solución cumple con los requisitos impuestos para el alcance del proyecto de la siguiente manera:

- Consiste en una pulsera con 8 electrodos superficiales secos activos y un sensor IMU (Unidad de Medición Inercial) [50], lo que permite la captación de los movimientos musculares, movimientos del brazo y posición del mismo. En éste trabajo sólo se usarán las capacidades mioeléctricas del dispositivo
- Toma muestras a una frecuencia de 200Hz, las amplifica y las digitaliza, de tal manera que el resultado final es una función llamada "activación", adimensional y no comparable directamente con milivoltios, dando valores discretos en un rango (0, 255) [50]
- Tiene la capacidad de transmitir todos los datos que produce mediante conectividad BLE (Bluetooth Low Energy), usando el protocolo publicado por el fabricante [22]

Posee, además, alimentación propia y autonomía, lo que lo convierte en un dispositivo cómodo para su uso para desarrollo de prótesis autónomas. Aún cumpliendo todos los requerimientos, la Myo Armband posee ciertas desventajas que afectarán al desarrollo de una solución de prótesis diestra:

- Bajo ancho de banda: debido a la baja potencia de la electrónica y la alimentación que es posible embeber en el dispositivo, el dispositivo se ve limitado a una frecuencia de muestreo de 200Hz, notablemente baja, de aproximadamente 5 veces menor que la frecuencia de Nyquist de una señal EMG (rondando los 1000 hz) [41]. Este hecho afecta directamente a la cantidad y calidad de información que podremos sacar del sensor, y hará que la clasificación de las posiciones sea más compleja y menos fiable
- Ruido: debido al tipo de sensor usado, está bastante afectado por factores como la colocación del mismo y las características de la piel del individuo, de tal manera que las señales no serán iguales entre colocaciones del sensor diferentes e individuos diferentes
- Autonomía limitada: debido al reducido tamaño, la batería que acompaña al dispositivo es muy pequeña [50]. Además, la captación de las señales en bruto requeridas para este trabajo consume una cantidad considerable de energía, lo que merma aún más el tiempo de autonomía. Por último, el dispositivo se desconecta automáticamente al ser conectado a una fuente de carga, lo que lo hace inútil mientras recarga las baterías

Éstas desventajas van a originar una serie de problemas previsibles antes del desarrollo de cualquier clasificador Machine Learning: debido al bajo ancho de banda provocado por su baja frecuencia de muestreo y a la imposibilidad de colocar los sensores en exactamente la misma posición en la que se grabaron los datos de entrenamiento, la precisión de clasificación del modelo se verá afectada. Este problema puede solucionarse mediante el uso de clasificadores que nos den la más alta precisión posible con los datos de entrenamiento, usando marcadores para la posición de la pulsera y mantener limpias las superficies de los sensores y la piel. Otro enfoque, dependiendo del hardware utilizado y la destreza final lograda, puede ser considerar el entrenamiento en el arranque, en caso de que se disponga de una capacidad de procesamiento suficiente para que el grabado y entrenamiento ocupe poco más de 15 minutos, de tal manera que el acoplamiento y uso de la prótesis pudiera incorporar esta rutina al usarlo sin requerir demasiado tiempo, esfuerzo o conocimiento del sistema al usuario final.

Obtención de señales

Tal y como se ha descrito en la sección anterior, las señales EMG obtenidas de este dispositivo tienen las siguientes características:

- 8 canales
- Frecuencia de muestreo de 200 Hz
- Función de “activación”, representa la activación del músculo pero no es directamente traducible en mV
- Valores empaquetados en bytes individuales. Valores naturales, rango (0, 255)

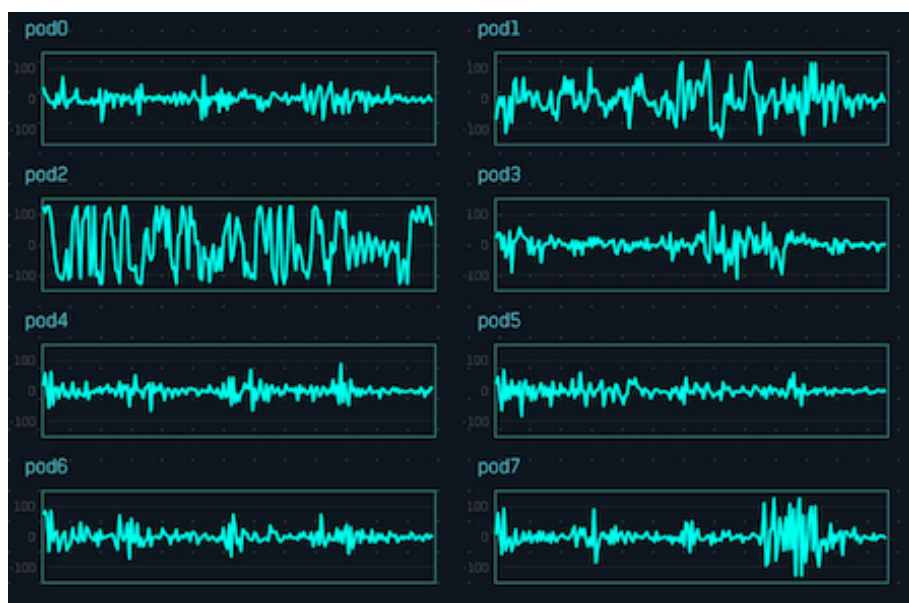


Figura 3.3 : datos graficados de los 8 canales de la Myo Armband, extraído de Thalmic Labs, actual NORTH [49] [25]

El dispositivo posee una conectividad BLE que permite la obtención de las señales EMG en bruto [23]. La Myo Armband se comporta como un servidor GATT (Generic Attribute Profile), permitiendo que un cliente se suscriba a los atributos publicados. En este caso nos interesa suscribirnos únicamente a los servicios concernientes a las señales EMG. Para recibir los valores de EMG ha de suscribirse a las cuatro características de datos de EMG y enviando el siguiente mensaje a la característica de comando:

COMMAND ID	SIZE	DATA			
0x01	0x03	0x02	0x00	0x00	
0	+1	+2	+3	+4	+5

Figura 3.4 : PDU de petición de envío de valores de EMG

Adicionalmente, y para que la pulsera no se apague durante su uso, se enviará un mensaje adicional de hibernación a la característica de comando:

COMMAND ID	SIZE	DATA
0x09	0x01	0x01

0 +1 +2 +3

Figura 3.5 : PDU de comando de “never sleep”

Una vez suscritos a los servicios necesarios, las señales se reciben en paquetes de 16 bytes, los cuales contienen 16 valores de un cluster de 2 EMG, 8 valores por cada sensor.

DATA 1								DATA 2							
0xZZ	0xZZ	0xZZ	0xZZ	0xZZ	0xZZ	0xZZ	0xZZ	0xZZ	0xZZ	0xZZ	0xZZ	0xZZ	0xZZ	0xZZ	0xZZ

0 +8 +16

Figura 3.6 : PDU de valores de EMG

Se envían, por tanto, 4 mensajes de este tipo para completar un paquete de 8 valores para los 8 sensores. Ya que se reciben 200 valores por segundo por cada sensor, se reciben exactamente 100 mensajes por segundo, con una recepción de datos de aproximadamente 1.6 kB/s , sin tener en cuenta los datos de cabecera, comprobación, etc. Aun teniendo en cuenta esto, el protocolo BLE permite un envío y recepción de datos significativamente menor, por lo que la obtención de estos datos no representa un impacto grande en el tiempo de ejecución de cada ciclo de clasificación de posiciones.

Obtención de parámetros de las señales

Una vez obtenidas las señales, el siguiente paso podría ser alimentarlas directamente al clasificador. Ha de tenerse en cuenta que, debido a la naturaleza oscilatoria de una gráfica EMG, no tiene sentido dar valores individuales de las 8 gráficas de EMG a un clasificador, ya que no tendría oportunidad alguna de sacar un patrón de datos aparentemente inconexos, sacados del contexto de la onda que forman por cada canal.

La siguiente posibilidad sería usar una porción de onda de cada sensor en un período de tiempo, de tal manera que se le alimentaran n valores de cada gráfica en un espacio t de tiempo. Éste enfoque, aunque más razonable, presenta un problema en cuanto a la cantidad de datos que el clasificador habrá de procesar.

Haciendo un ejercicio de comprobación, asumiremos que se van a usar tramos de 500 ms, solapados 250 ms con el anterior tramo, lo que daría una cantidad de 100 valores individuales por cada sensor, 800 datos cada 250 ms, lo cual, teniendo además en cuenta las profundas no linealidades de los patrones de estos datos, podría hacer que el cálculo y clasificación por cada conjunto de datos llevara una ingente cantidad de recursos, y por tanto, tiempo.

El enfoque más usado y más razonable es reducir este conjunto de datos en forma de onda, obteniendo características de la misma, ya sean en el dominio de la frecuencia o el tiempo, así como otro tipo de características. De esta manera, seleccionando características relevantes para su uso en clasificadores, podremos usar conjuntos mucho más pequeños de datos y con patrones mucho más fáciles de discernir, lo que reduciría considerablemente la complejidad y recursos consumidos por el clasificador. En el caso de este trabajo, se obtendrán conjuntos de datos de la siguiente manera:

- Tramos de 50 datos (equivalentes a 25 ms)
- Solapamiento de 25 datos, la mitad del conjunto anterior (equivalente a 0.125 ms)

Por tanto, tendremos conjuntos de datos de los que obtener características cada 0.125 ms, lo que nos brindará la posibilidad de obtener predicciones de posición en tiempos razonables para una alta destreza.

Una vez obtenido el conjunto de datos, se pasaría al cálculo de las características de la gráfica EMG por cada uno de los 8 canales. Para esto, se usará un conjunto de características lo más relevante posible para su uso en clasificadores.

Hay una gran cantidad de características en tiempo y frecuencia que se pueden obtener de una gráfica. En este trabajo nos limitaremos al uso de un conjunto de características con relevancia comprobada [42] [41]:

- L-Scale (LS): estadística de orden 2 de momentos-L, usada para resumir la forma de una distribución de probabilidad [50] [20]. Su fórmula de cálculo es la siguiente:

$$\lambda_r = r^{-1} \sum_{k=0}^{r-1} (-1)^k \binom{r-1}{k} EX_{r-k:r} \quad (3.1)$$

Siendo r un valor de 2.

- Maximum Fractal Length (MFL): máxima longitud de fractal de la onda [43] [1]. Su fórmula de cálculo es la siguiente:

$$MFL = \log_{10} \left(\sqrt{\sum_{n=1}^{N-1} (x(n+1) - x(n))^2} \right) \quad (3.2)$$

- Mean value of the Square Root (MSR): Valor medio de la raíz cuadrado de los valores de la gráfica [42]. Su fórmula de cálculo es la siguiente:

$$MSR = \frac{\sum_{i=1}^N \sqrt{x_i}}{N} \quad (3.3)$$

- Willison Amplitude (WAMP): se define como las veces que la amplitud de señal entre dos segmentos adyacentes supera un umbral predefinido [42]. Su fórmula de cálculo es la siguiente:

$$WAMP = \sum_{i=1}^{N-1} [f(|x_n - x_{n+1}|)];$$

$$f(x) = \begin{cases} 1, & \text{if } x \geq \text{threshold} \\ 0, & \text{otherwise.} \end{cases} \quad (3.4)$$

- Zero Crossing (ZC): se define como el número de veces que una onda cruza el valor de amplitud cero. [42]. Su fórmula de cálculo es la siguiente:

$$ZC = \sum_{i=1}^{N-1} [sgn(x_i \times x_{i+1}) \cap |x_i - x_{i+1}| \geq \text{threshold}];$$

$$sgn(x) = \begin{cases} 1, & \text{if } x \geq \text{threshold} \\ 0, & \text{otherwise.} \end{cases} \quad (3.5)$$

- Root Mean Square (RMS): se define como la raíz cuadrada de la media de los cuadrados de los valores de la onda [42]. Su fórmula de cálculo es la siguiente:

$$RMS = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2} \quad (3.6)$$

- Integral Absoulte Value (IAV): se define como el valor integral absoluto de una curva que se cruza con el valor de amplitud cero [42]. Su fórmula de cálculo es la siguiente:

$$IAV = \int_a^b |f(x)| \quad (3.7)$$

- Difference Absolute Standard Deviation Value (DASDV): se define como la desviación estándar de la longitud de onda [42]. Su fórmula de cálculo es la siguiente:

$$DASDV = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N-1} (x_{i+1} - x_i)^2} \quad (3.8)$$

- Variance (VAR): se define generalmente como la media de la suma de los valores de una variable [42]. Su fórmula de cálculo es la siguiente:

$$DASDV = \frac{1}{N-1} \sum_{i=1}^N x_i^2 \quad (3.9)$$

Este conjunto de características se denomina TD9 [50]. De esta manera, obtendremos 9 valores característicos de cada canal de EMG, es decir, 72 datos por cada período de clasificación.

Adicionalmente, y como último paso, debido a que algunas de estas características se encuentran en órdenes de magnitud de alrededor de 10_4 y 10_5 , se dividirán por un número arbitrario con el objetivo de equilibrarlas a la hora de introducirlas como dato en un clasificador mediante Machine learning. Esto se hace con el objetivo de que el valor numérico que puedan alcanzar estas variables no sea excesivamente grande y provoque errores numéricos y de desbordamiento, sobre todo en los primeros pasos de entrenamiento de un algoritmo de ML.

Implementación de la interfaz

Como se ha descrito en el comienzo del capítulo, ésta sección será completamente implementada en Python. La misma estará dividida en varias partes diferenciadas:

```
+---driver
|   |   myo.py
|   |   myo_codes.py
|   |   myo_protocol.py
|   |   myo_protocol_simplified.py
|   |   myo_simplified.py
|   |   __init__.py
|   |
|   \---__pycache__
|           myo.cpython-37.pyc
|           __init__.cpython-37.pyc
|
+---feature
|       feature_extraction.py
|
```

■ Driver BLE:

- Protocolo Bluetooth completo: contiene todas las funciones y clases necesarias para el uso de todas las características de la Myo Armband, construyendo un API que permita la construcción de mensajes y comandos, codificación y decodificación de la manera más sencilla posible al usuario.

```
+---driver
|   |   myo_protocol.py
|   |
|
```

- Protocolo Bluetooth simplificado: contiene las funciones y clases mínimamente necesarias para la obtención de los valores de los sensores EMG, de la manera más rápida y eficiente posible, con una API menos orientada al usuario.

```

+---driver
|   |   myo_protocol_simplified.py
|   |
|

```

- Driver completo: controlador con todas las funcionalidades, características y servicios disponibles por la Myo Armband

```

+---driver
|   |   myo.py
|   |
|

```

- Driver simplificado: controlador simplificado, con únicamente las funciones necesarias para la obtención de valores de los sensores EMG de la manera más rápida y eficiente posible

```

+---driver
|   |   myo_simplified.py
|   |
|

```

- Obtención de características EMG: funciones necesarias para obtener las características o conjunto de características dados los valores de un tramo de gráfica, sea EMG o cualquier otro tipo de gráfica.

```

+---feature
|       feature_extraction.py
|

```

Además de los módulos anteriormente descritos, se ha implementado un fichero de códigos que define todos los diferentes códigos del protocolo, haciendo el código más legible y descriptivo. Este fichero es usado por los módulos de protocolo y drivers.

```

+---driver
|   |   myo_codes.py
|   |
|

```

Durante el desarrollo de los módulos anteriores se han generado los siguientes *scripts* de test:

```
+---test
|  |  test_features.py
|  |  test_feature_extraction.py
|  |  test_myo.py
|  |  test_myo_protocol.py
|  |
|
```

3.2. Controlador Machine Learning (Clasificador de movimientos)

El siguiente bloque en orden a desarrollar corresponde a la obtención de las posiciones del brazo mediante el procesamiento de las señales EMG preprocesadas mediante algoritmos Machine Learning, de tal manera que sea capaz de clasificar la posición del brazo con una precisión razonable, la cual estará limitada por la capacidad de clasificación del algoritmo seleccionado y la calidad de los datos, así como, en última instancia, factores dependientes de la situación del sensor en el brazo y de las condiciones cutáneas en el momento de su uso.

Plataforma seleccionada

Con el objetivo de acelerar el desarrollo del controlador se ha decidido usar la librería de Machine Learning Tensorflow, que permite implementar, entrenar y ejecutar una gran variedad de modelos y tipos de modelos de aprendizaje automático, como pueden ser regresores lineales, clasificadores multiclase, redes neuronales profundas, etc [48]. Es de código abierto y cuenta con un ecosistema integral que permite desarrollar un sistema Machine Learning de principio a fin, además de tener una gran comunidad y recursos y ejemplos de los que tomar referencia para aprender su funcionamiento de manera rápida.

Esta librería es compatible con Python y compatible con la inmensa mayoría de sistemas en los que Python puede ejecutarse (Windows y basados en Linux), lo que lo hace un candidato idea para su implementación en sistemas embebidos con sistemas Linux ligeros, tales como la Raspberry Pi [44] y la Jetson Nano de Nvidia [38], la última poseyendo una mayor capacidad de computación y dedicada específicamente a tareas de ML.

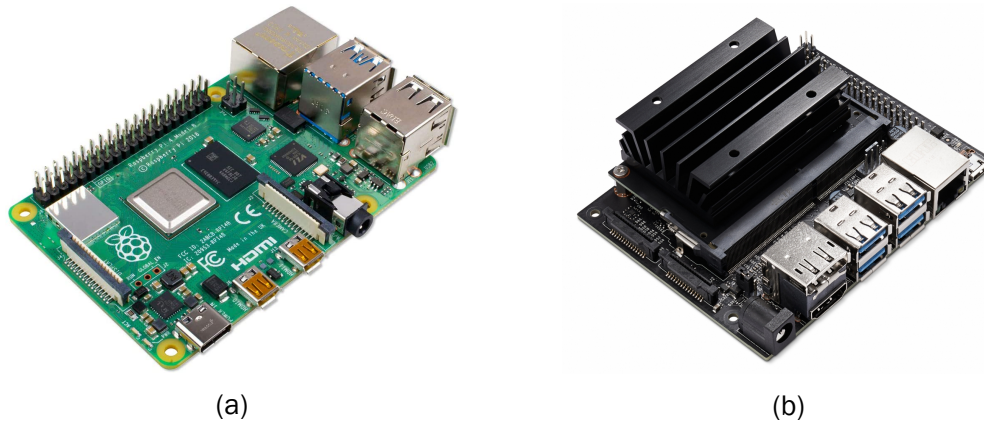


Figura 3.7 : Raspberry Pi 4b, extraído de Raspberry Pi (a) y Jetson Nano, extraído de Nvidia (b)

El hecho mencionado anteriormente también nos permite, comprobando previamente la compatibilidad con los sistemas embebidos mencionados anteriormente, desarrollar y probar el sistema desarrollado en este trabajo en equipos de escritorio con mayor potencia de computación que permitan un entrenamiento rápido del modelo de aprendizaje automático y una prueba fluida del mismo.

En este trabajo se ha decidido usar un ordenador de escritorio provisto de potencia suficiente para estas tareas y la versión 1.13.1 de la librería Tensorflow, maximizando así la compatibilidad con otros sistemas. La decisión sobre la librería se tomó bajo el conocimiento de la incapacidad del sistema Raspbian para Raspberry Pi de ejecutar versiones superiores a 1.14 usando Python 3.7.

Características del ordenador en el que se ejecutará el entrenamiento, pruebas y funcionamiento del sistema:

- Procesador AMD Ryzen 7 3800X, con velocidad de reloj de 3.9 GHz, 8 núcleos y 16 hilos, con potencia de cálculo de 648GFLOPS (según gadgetversus.com [16])
- 32 GB RAM DDR4 a 2666MHz
- Procesador gráfico Nvidia Gigabyte GeForce RTX2080 SUPER Gaming OC, con 8GB de memoria GDDR6 dedicada y 384 Tensor Cores, con una potencia de cálculo de 22.3TFLOPS 11.15TFLOPS y 348.5GFLOPS para FP16, FP32 y FP64 respectivamente (según techpowerup.com [47])

Algoritmo seleccionado

Debido a las no linealidades que inevitablemente poseerán las características extraídas de las gráficas EMG, se ha decidido usar un clasificador multiclase basado en Redes Neuronales Profundas (DNN, Deep Neural Network), con el objetivo de, mediante la definición de las suficientes capas y tamaño de las mismas, se pueda entrenar un sistema con una fiabilidad en datos de validación de un 80 % o mayor.

Para la implementación del clasificador antes se debe definir las entradas de datos que van a usarse y las clases en las que clasificará los datos de entrada, las cuales corresponderán a las posiciones que queramos predecir. Las entradas quedaron definidas en el desarrollo de la sensórica, teniendo 72 entradas de datos en punto flotante de 32 bits.

Las clases, o posiciones, se han elegido en base a posiciones habituales usadas en actividades de vida diaria, incluyendo un número razonable de posiciones que provean de una cierta destreza, pero no sobrepasen la capacidad de los datos obtenidos o la red neuronal que va a clasificarlos. En éste caso se ha decidido usar las siguientes posiciones:

1. Mano relajada: posición de mano con los músculos del antebrazo no activos.



Figura 3.8 : posición de mano relajada

2. Mano abierta: extensión completa de la mano.



Figura 3.9 : posición de mano extendida

3. Mano en puño: cierre completo y con fuerza de la mano.



Figura 3.10 : posición de mano en puño

4. Mano apuntando: puño cerrado a excepción del índice extendido, útil para teclear.



Figura 3.11 : posición de mano apuntando

5. Mano pellizcando: pulgar opuesto con yema junto con dedo corazón, realizando una acción de pellizcado, útil para agarre de objetos pequeños, finos, etc.



Figura 3.12 : posición de mano pellizcando

6. Mano agarrando: mano cerrada sin fuerza completa, útil para agarrar objetos como vasos, palos, etc.



Figura 3.13 : posición de mano agarrando

7. Mano agarrando: puño cerrado a excepción del pulgar levantado.



Figura 3.14 : posición de mano con pulgar levantado

Una vez definidas las 72 entradas y 7 clases, podemos definir el resto de características del clasificador:

- Capas:
 - Entrada: 72 nodos de entrada completamente interconectados.

- Salida: 7 clases mediante una capa softmax.
 - Ocultas: 3 capas de 72, 36 y 18 modos respectivamente, completamente interconectados.
- Topología: nodos completamente interconectados
 - Función de activación: Tanh

Una vez definida la red, se ha procedido a su entrenamiento, partiendo de datos grabados mediante la implementación de la sensórica y scripts de grabación de datos en .csv. Se han grabado 60 segundos de cada posición en tandas de 10 segundos cada una, generando alrededor de 80 datos en ese período por posición, sumando un total de datos aproximado de 3360 entradas.

La red entrenada más exitosa posee una fiabilidad del 87 % con la siguiente función de pérdida y matriz de confusión:

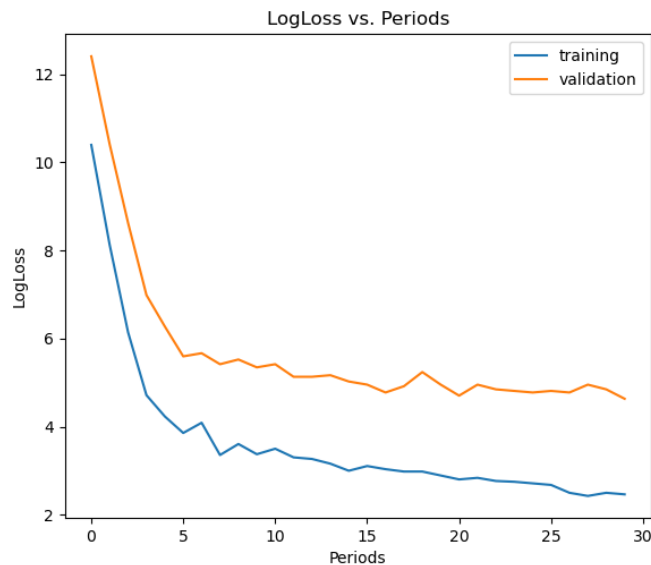


Figura 3.15 : evolución de la función de pérdida logarítmica a través los 30 períodos de entrenamiento

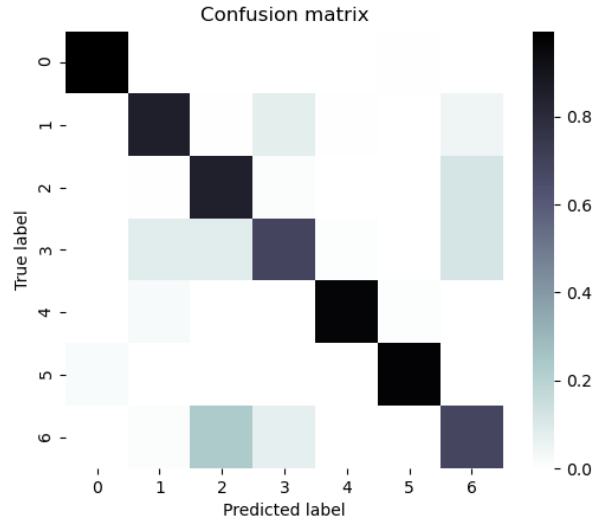


Figura 3.16 : matriz de confusión de las posiciones entrenadas

Con los siguientes hiperparámetros:

- Tasa de aprendizaje: 0.01
- Pasos: 200000
- Períodos: 30 (divide en 30 los pasos)
- Tamaño de lote: 256

Como se puede observar, la red responde con una fiabilidad relativamente estable a todas las posiciones, siendo menos fiable la posición de apuntar y pulgar levantado, ya que difieren en la activación/desactivación de grupos musculares más pequeños que el resto de posiciones.

Ésta red será usada por el sistema completo y calculará las predicciones de los datos recibidos en tiempo real a través del driver implementado en la sección de sensorica.

Implementación del controlador

La implementación de la red neuronal, debido a la necesidad de separar su entrenamiento y su uso, se ha dividido en varios módulos:

```
D: .
|  gui.py
|  __init__.py
|  __main__.py
|
+---images
|     hand.jpg
|
+---scripts
|     startup.py
|
+---test
|  |  test_dnn.py
|
+---training
|  |  basic_recording.py
|  |  complete_recording.py
|  |  data_visualization.py
|  |  model_generator_td9_7.py
|  |
|  +---data
|  |     rec_td9_0_0.csv
|  |     rec_td9_0_1.csv
|  |     rec_td9_0_2.csv
|  |     rec_td9_0_3.csv
|  |     rec_td9_0_4.csv
|  |     rec_td9_0_5.csv
|  |     rec_td9_1_0.csv
|  |     rec_td9_1_1.csv
|  |     rec_td9_1_2.csv
|  |     ...
|  |
|  \---models
```



```

|      +---1592053653
|      | | saved_model.pb
|      | |
|      | | \---variables
|      | |     variables.data-00000-of-00002
|      | |     variables.data-00001-of-00002
|      | |     variables.index
|      | |
|      +---1592054158
|      | | saved_model.pb
|      | |
|      | | \---variables
|      | |     variables.data-00000-of-00002
|      | |     variables.data-00001-of-00002
|      | |     variables.index
|      | |
|      +---...
|      |
|

```

El cual se puede dividir en las siguientes partes:

- Script de startup: puede usarse para el inicio del controlador y lanza todas las rutinas necesarias para conectar los sensores, cargar la red y conectarse al controlador del dispositivo mecánico, preparando el funcionamiento del brazo de forma completamente automática.

```

+---scripts
|     startup.py
|

```

- GUI completa: permite la selección de modelos, datos usados, visualización de la posición, etc. Para uso como demostrador y para entornos de prueba y desarrollo, no como uso final.

```

D: .
|  gui.py
|  __init__.py
|  __main__.py
|

```

- Scripts de captura de datos: capturan los datos de salida del módulo de sensorica y los guardan en fichero .csv para su uso en entrenamiento.

```

+---training
|   |   basic_recording.py
|   |   complete_recording.py
|   |
|   +---data
|   |   rec_td9_0_0.csv
|   |   rec_td9_0_1.csv
|   |   rec_td9_0_2.csv
|   |   rec_td9_0_3.csv
|   |   rec_td9_0_4.csv
|   |   rec_td9_0_5.csv
|   |   rec_td9_1_0.csv
|   |   rec_td9_1_1.csv
|   |   rec_td9_1_2.csv
|   |   ...
|   |
|   |
|   |

```

- Entrenamiento: script de entrenamiento de la red neuronal.

```

+---training
|   |   model_generator_td9_7.py
|   |
|   |
|   |

```

- Modelos entrenados: carpeta con los modelos entrenados que pueden ser cargados por el script de startup y la GUI de prueba.

```

+---training
|   \---models
|       +---1592053653
|           |   |   saved_model.pb
|           |   |
|           |   \---variables
|           |       variables.data-00000-of-00002
|           |       variables.data-00001-of-00002
|           |       variables.index
|           |
|           |
|           |

```

3.2. CONTROLADOR MACHINE LEARNING (CLASIFICADOR DE MOVIMIENTOS) 55

```
|      |
|      +---1592054158
|      |   |   saved_model.pb
|      |   |
|      |   \---variables
|      |           variables.data-00000-of-00002
|      |           variables.data-00001-of-00002
|      |           variables.index
|      |
|      +---...
|      |
|
```

Adicionalmente se han usado los siguientes scripts a la hora de probar las funciones de Tensorflow, con finalidad didáctica y de prueba.

```
+---test
|   |   test_dnn.py
|
```

3.3. Prótesis electromecánica

El siguiente orden en el desarrollo es el uso de un dispositivo mecánico que traduzca las posiciones obtenidas mediante el sensor y el clasificador en movimientos reales, de tal manera que sea útil y aplicable como prótesis en amputados transradiales.

Selección del diseño mecánico

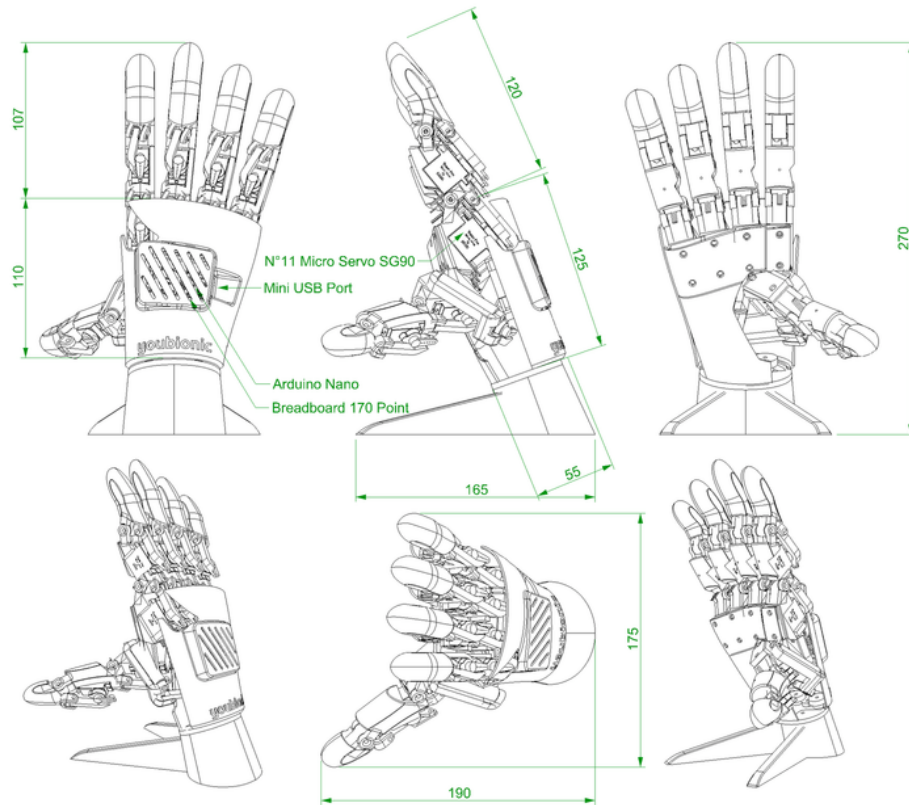
Teniendo en cuenta los objetivos del trabajo, y para acotar el alcance del trabajo, se ha seleccionado un diseño mecánico existente, que cubra los siguientes requisitos:

- Control simple: deberá disponer de actuadores lo suficientemente simples para que la implementación de un controlador sea inmediata de implementar, o que incluya una solución de control ya preparada.
- Destreza razonable: deberá tener los suficientes grados de libertad como para ejecutar, al menos, las posiciones descritas durante el desarrollo del controlador Machine Learning.
- Barato: el brazo deberá ser fácil y barato de fabricar, con piezas impresas en 3D, actuadores baratos y con una complejidad de fabricación y montaje mínima. No se usará como criterio la posibilidad de producción en serie del mismo, ya que, como prototipo, este requisito escapa el alcance del trabajo.
- Aplicable a amputados transradiales: éste requisito resulta más complejo de satisfacer, por lo que lo consideraremos como deseable en el alcance de este trabajo. El brazo debería poder ser directamente aplicable en el brazo de un amputado transradial, con ninguna o mínima modificación y adaptación sobre el diseño para el correcto acoplamiento del mismo al paciente.

Dentro del abanico de diseños mecánicos cubiertos en el estado del arte, se ha decidido usar la mano robótica *Handy* diseñada por Federico Cicaresse en Youbionic [8]. El diseño tiene un coste de 79€ [9] y las piezas y fabricación ascienden el coste total a 150€ aproximadamente.



(a)



(b)

Figura 3.17 : mano robótica Youbionic *Handy*, extraído de Youbionic

El dispositivo cumple con los requisitos establecidos gracias a las siguientes características:

- El control del brazo se puede realizar alimentando los servomotores SG90 que lo componen y señalizando la posición mediante PWM, fácilmente implementable en microcontroladores, tales como plataformas basadas en Arduino.
- Posee 11GDL, 3 en el pulgar y 2 en cada uno de los dedos restantes, permitiendo un movimiento suficientemente diestro como para ejecutar las posiciones definidas en el desarrollo del controlador Machine Learning.
- El coste total es muy bajo, y la fabricación del mismo es sencilla, mediante el uso de impresión 3D (FDM) con materiales de bajo coste, como PLA y ABS.
- Gracias a la modularidad del diseño, permite que la estructura principal de la mano se acople a:
 - Pieza de manga adaptable a amputados transradiales.



(a)



(b)

Figura 3.18 : manga adaptable a amputados transradiales, extraído de Youbionic

- Plataforma de apoyo para superficies planas.

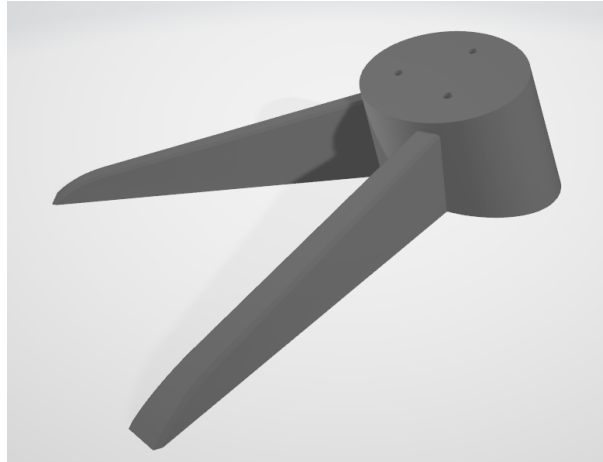


Figura 3.19 : soporte para *Handy*, modelo extraído de Youbionic

- Antebrazo robótico *Human Arm*, de Youbionic [6].



Figura 3.20 : antebrazo robótico *Human Arm*, extraído de Youbionic

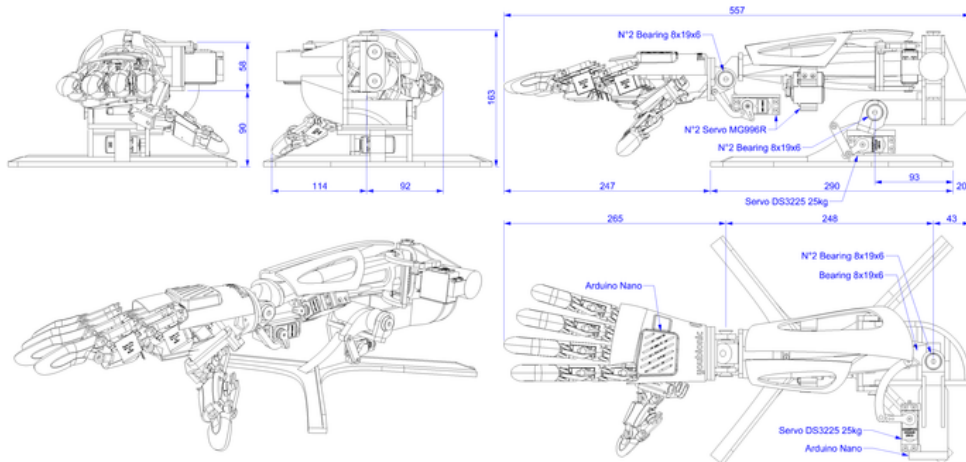


Figura 3.21 : diagrama de antebrazo robótico *Human Arm*, extraído de Youbionic

Posee las siguientes características técnicas, descritas la página de venta del diseño [8] [9]:

- Tipo de mano robótica: movimiento activo
- N° de servos activos: 11
- Voltaje: 5V
- Tipos de actuadores: Servomotor SG-90
- Par por servo: 2.5 kg
- Materiales: PLA impreso en 3D

Cabe destacar que uno de los diseños mecánicos originalmente considerados para este trabajo fue el brazo mecánico InMoov [27], el cual posee 6 servomotores de alta carga, con independencia en todos los dedos y movimiento de muñeca, para el cual se desarrollaron ciertas pruebas, pero se descartó en última instancia al no cumplir con el requisito de aplicabilidad a amputados transradiales, debido a su gran tamaño, peso y alojamiento de los servos en la sección del antebrazo, lo que hace imposible implantar de forma más o menos directa el brazo a un sujeto.

Fabricación, montaje y prueba del brazo

Posterior a la compra, el vendedor ha proporcionado todos los modelos 3D del dispositivo mecánico, así como un somero manual de instrucciones de montaje, junto con las piezas adicionales necesarias, disponible también en la página web del producto [5].

- Piezas de impresión 3D:
 - (1x) *Handy_Back - Right*
 - (1x) *Handy_Palm and Rotation - Right*
 - (1x) *Wearable Part Complete*
 - (1x) *Handy_Dock Station*
 - (1x) *Handy_Thumb*
 - (1x) *Handy_Index*
 - (1x) *Handy_Middle*
 - (1x) *Handy_Ring*
 - (1x) *Handy_Pinkie*

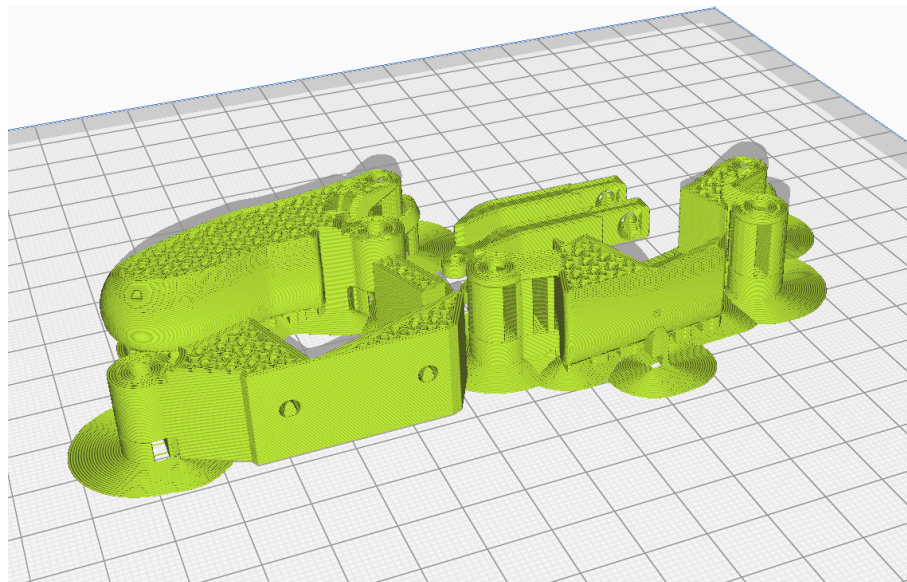


Figura 3.22 : *Handy_Pinkie* laminado mediante Cura 4.6, modelo 3D extraído de Youbionic

- (x11) Servomotor SG-90



Figura 3.23 : servomotor SG-90, extraído de Iberobotics

- (x10) Tornillo cabeza avellanada Hex M3x20
- (x23) Tornillo cabeza avellanada Hex M3x15
- (x6) Tornillo cabeza avellanada Hex M3x10
- (x11) Tornillo cabeza redondeada Philips M1.5x6

Las piezas 3D han sido fabricadas con los siguientes equipos de impresión 3D:

- Anet A8-M altamente modificada: impresora 3D de doble extrusor Chimera con gargantas metálicas, temperatura máxima de 320°C por extrusor, con cama caliente Ultrabase y borosilicato con lámina de PEI lijada, enclaustrada. Electronica BIQU KFB2.0, firmware Marlin 2.0 y mosfets para los extrusores y cama. Volumen (X, Y, Z): 220mm, 220mm, 240mm.

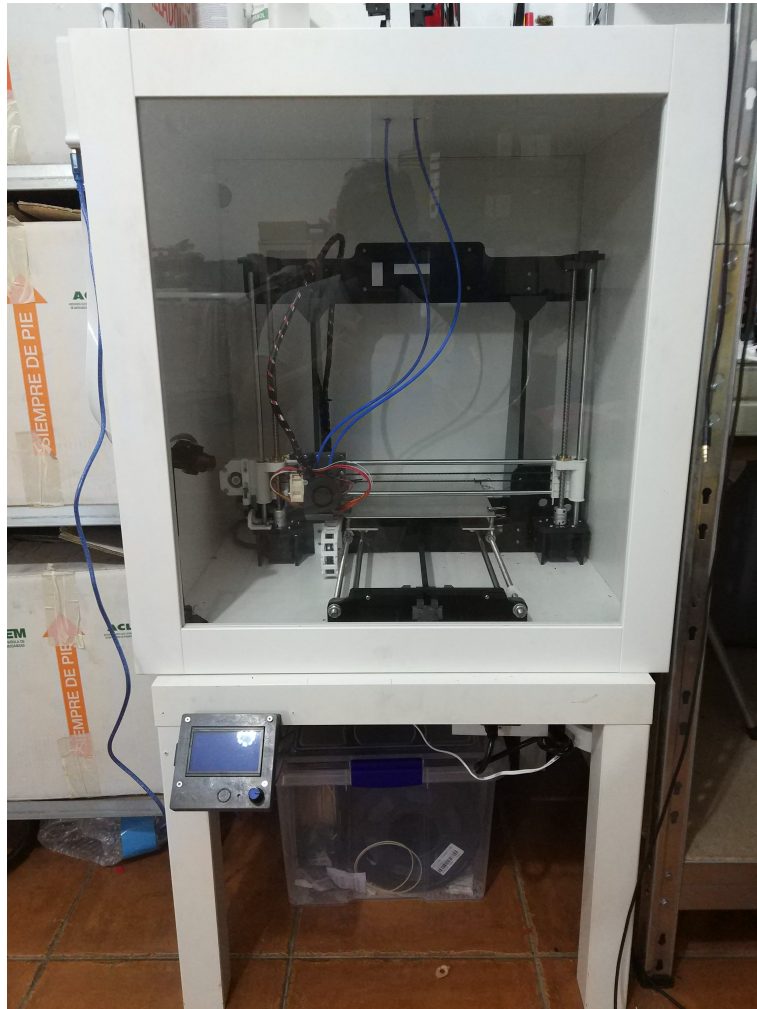


Figura 3.24 : impresora 3D Anet A8-M altamente modificada

- CR-10 S4 altamente modificada: impresora 3D de extrusor simple Creality con garganta PTFE, temperatura máxima de 250°C, con cama caliente de calentador de silicón de 1000W 230V, aislamiento inferior y lámina de PEI lijada, con rigidizadores de estructura pórtico. Electrónica BigTreeTech SKR1.4, firmware Marlin 2.0 y relé de estado sólido para control de la cama caliente. Volumen (X, Y, Z): 400mm, 400mm, 400mm.

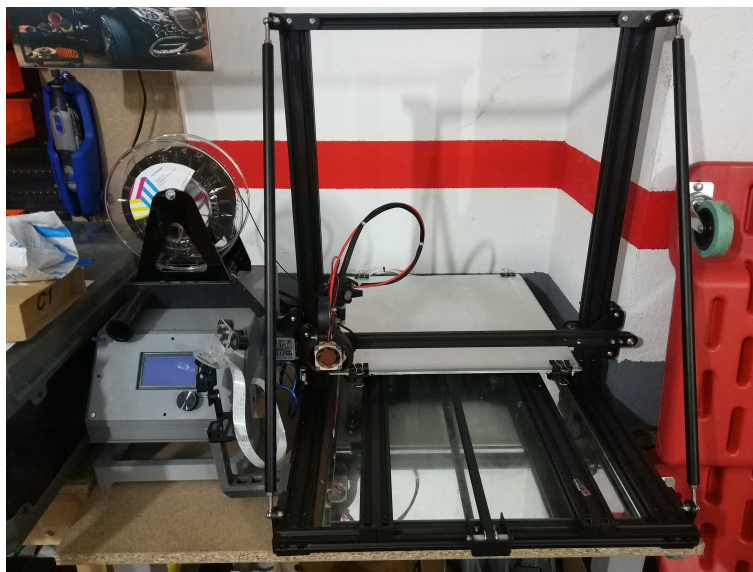


Figura 3.25 : impresora 3D Creality CR-10 S4 altamente modificada

Todas las piezas han sido fabricadas en la A8-M en ABS negro, consumiendo aproximadamente 400g, sumando un coste de 8€, a excepción de la pieza *Handy_Back-Right*, impresa por la CR-10 S4 en PLA negro, consumiendo aproximadamente 40g, sumando un coste de 0.80€, por lo que el coste total de las piezas impresas en 3D asciende a 8.80€ aproximadamente.

El proceso de fabricación está detallado en el *Handbook* provisto por Youbionic con la compra de los diseños [5].

Los costes individuales de las piezas quedan detallados en los anexos de este proyecto, en el BOM del diseño mecánico A.

3.4. Controlador del brazo

En último orden, una vez disponemos de un dispositivo mecánico provisto de actuadores para realizar los movimientos pertinentes, necesitaremos un controlador, el cual tome los datos obtenidos del clasificador y los implemente en un lazo de control, ejecutado por un hardware que sea capaz de alimentar los actuadores y moverlos acorde al modelo diseñado.

Algoritmo seleccionado

Se ha seleccionado un sistema de control en lazo abierto, al cual se le alimenta el número correspondiente a la posición estimada y como salidas obtiene los ángulos de los 11 servomotores que componen la prótesis electromecánica.

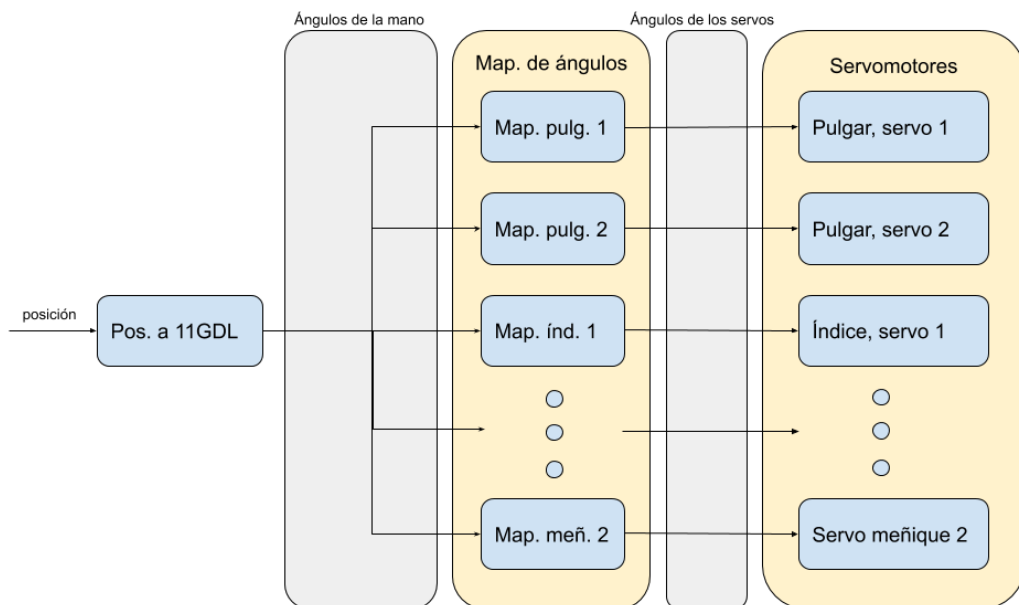


Figura 3.26 : diagrama de bloques del algoritmo de control del brazo

En este caso, se ha considerado el lazo abierto por las siguientes razones:

- Sin sensorización en el diseño mecánico: debido a las características del diseño mecánico seleccionado no existe la posibilidad de sensorización de la

fuerza ejercida por los servomotores

- Posicionamiento de los servomotores: debido a que los servomotores siempre adaptan el ángulo dado por la consigna alimentada mediante PWM, no se ha considerado necesario un control de la posición que se alimenta a los mismos, ya que, si se considera que los servos no han sido forzados manualmente e ignorando el posible juego de la reducción, es aceptable considerar que seguirán la misma consigna siempre, ya que los mismos servos son capaces de controlar su posición.
- Alcance del trabajo: en el trabajo actual no se han considerado paradigmas de control más complejos, como podría ser un control háptico con feedbacks de fuerza y tacto, u otro enfoque similar, sino que se ha optado por sistemas ya desarrollados o comerciales para su inclusión en el sistema completo de la prótesis, para acotar el marco del trabajo

No tendremos ninguna manera de controlar la fuerza ejercida por los actuadores, por lo que, en períodos extendidos de uso y en posiciones específicas, los servomotores pueden dañarse por sobrecorrientes prolongadas.

El sistema es de tipo SIMO (Entrada única y múltiples salidas) discreto, cuyas entradas y salidas serán descritas a continuación:

- Entradas:
 - Posición: entrada entera, de rango discreto $[0, 6]$, la cual indica la posición estimada por el clasificador Machine Learning
- Salidas:
 - Ángulos de la prótesis electromecánica: salida múltiple, discretizada para diferentes posiciones, rango continuo dependiendo del grado de libertad, entre $[0, 90]$ grados
 - Pulgar Servo 1: rango $[0, 90]$
 - Pulgar Servo 2: rango $[0, 90]$
 - Pulgar Servo 3: rango $[0, 30]$
 - Índice Servo 1: rango $[0, 90]$
 - Índice Servo 2: rango $[0, 90]$
 - Corazón Servo 1: rango $[0, 90]$

- Corazón Servo 2: rango [0, 90]
- Anular Servo 1: rango [0, 90]
- Anular Servo 2: rango [0, 90]
- Meñique Servo 1: rango [0, 90]
- Meñique Servo 2: rango [0, 90]
- Ángulos de los servomotores: a partir de los ángulos de la prótesis se obtiene mediante un mapeado el ángulo correspondiente para el servomotor, entre [0, 180] grados de giro, máximo y mínimo para los actuadores seleccionados en el diseño mecánico. Los servos tienen el rango invertido respecto a los ángulos de la prótesis, por lo que han de ser mapeados a la inversa (ej. 0 corresponderá a 50, 90 corresponderá a 180)
 - Pulgar Servo 1: rango [180, 50]
 - Pulgar Servo 2: rango [180, 50]
 - Pulgar Servo 3: rango [0, 120]
 - Índice Servo 1: rango [180, 50]
 - Índice Servo 2: rango [180, 50]
 - Corazón Servo 1: rango [180, 50]
 - Corazón Servo 2: rango [180, 50]
 - Anular Servo 1: rango [180, 50]
 - Anular Servo 2: rango [180, 50]
 - Meñique Servo 1: rango [180, 50]
 - Meñique Servo 2: rango [180, 50]

Los servos están ordenados de la falange exterior a la falange interior, de pulgar a meñique.

Consta de los siguientes estratos diferenciables:

- Pos a 11GDL: transforma la entrada del número de posición en ángulos de la prótesis electromecánica.
- Mapeo ángulos: un mapeado por cada servo disponible, transforma el ángulo de la prótesis electromecánica al ángulo del servomotor
- Servos: uno por cada grado de libertad de la prótesis. Recibe el ángulo de consigna y mueve el servomotor correspondiente

Plataforma seleccionada

Como se ha mencionado en las características del diseño mecánico elegido, el control es sencillo, realizable por un microcontrolador basado en Arduino.

Debido a esto, y bajo recomendación del diseñador de la mano, se ha decidido usar una Arduino Nano, el cual puede controlar la posición de 12 servomotores diferenciados, lo que se ajusta perfectamente a las necesidades del diseño, el cual tiene 11 servomotores.

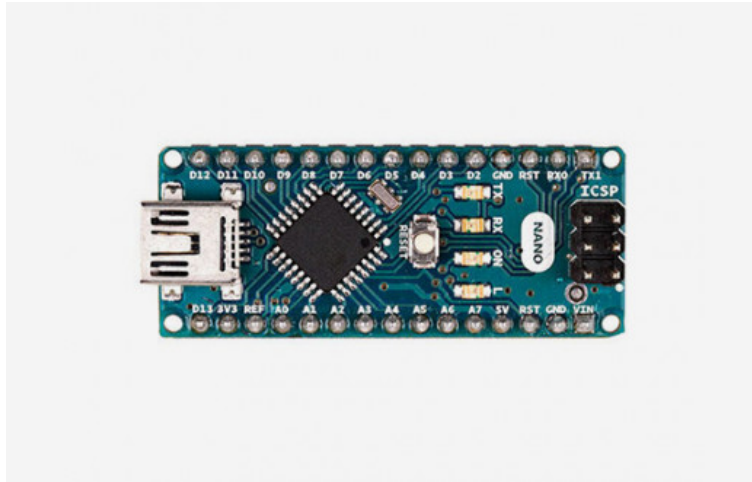


Figura 3.27 : microcontrolador Arduino Nano, extraído de Arduino

Para un uso y conexión simple, se ha fabricado un circuito de alimentación y conexión del Arduino Nano, así como los conectores necesarios para los 11 servomotores, usando una placa de pruebas.

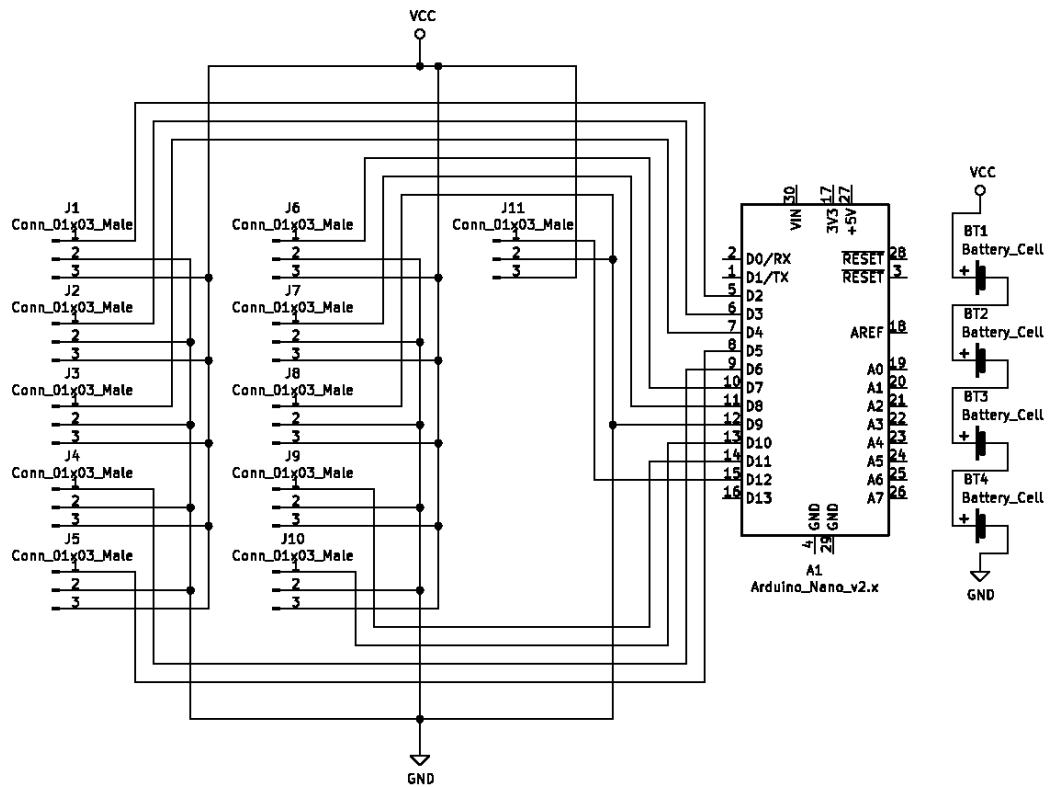


Figura 3.28 : esquema del circuito de alimentación y control de los servomotores de la mano

Este diseño provee de autonomía a los servomotores, pero el Arduino se mantiene alimentado mediante la conexión por USB al dispositivo que ejecutará el controlador Machine Learning, el cual será necesario también para la comunicación de las posiciones de los servomotores.



Figura 3.29 : circuito de alimentación y control de los servomotores de la mano

La alimentación de los servos se forma mediante 4 baterías AA Ni-MH PowerOwl de 1.25V en serie, las cuales proveen un voltaje de 6V, dentro del rango de alimentación de los servos. La corriente de descarga máxima es más grande que la consumida por los 11 servomotores, y la caída de tensión bajo esta corriente se considera despreciable, encontrándose en el peor de los casos dentro del rango de alimentación de los servos.

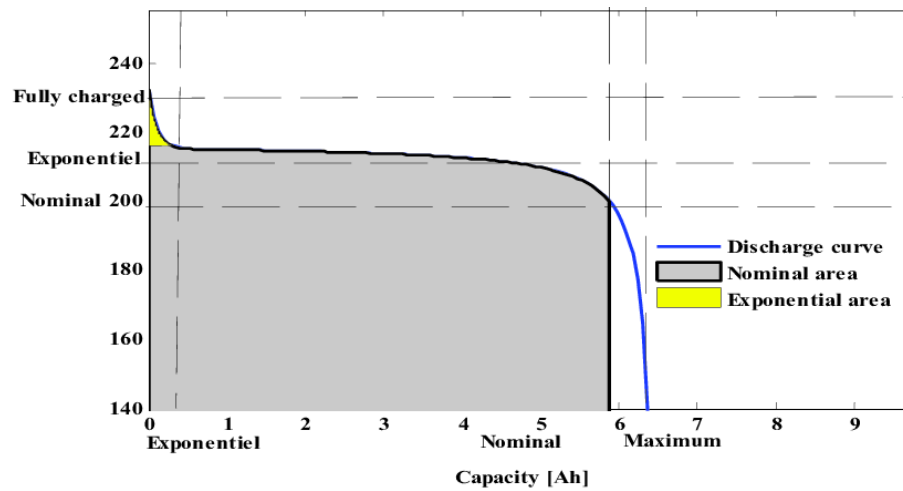


Figura 3.30 : curva de descarga de una batería Ni-MH estándar, extraído de Researchgate [30]

- Material de construcción: Ni-MH
- Factor forma: AA
- Tensión en vacío: 1.5V
- Capacidad anunciada: 2800mAh

Implementación del controlador

El control del brazo, tal y como está descrito en el diagrama de bloques del algoritmo, está dividido en un módulo del programa en python, el cual transforma el identificador de la posición en los ángulos de los 12 GDL y un programa de Arduino, el cual toma las posiciones enviadas por el puerto serie mediante un protocolo de comunicación por puerto serie con comprobaciones de tamaño, códigos de error y CRC, delimitados por caracteres especiales.

```

+---controller
|       pos_to_gdl11.py
|
+---youbionic_arm
|       youbionic_arm.ino
|
\---youbionic_test
      youbionic_test.ino

```

- Programa de Arduino para comunicación serial y control de servos: programa capaz de recibir mensajes con los ángulos de los GDL de la mano, y transformarlos en los ángulos de los servos.

```

+---youbionic_arm
|       youbionic_arm.ino
|

```

El mensaje usado para recibir estos ángulos se configura de la siguiente manera.

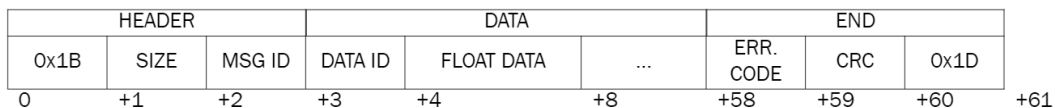


Figura 3.31 : PDU de ángulos de posición

Se pueden cargar menos o más datos en el mensaje, permitiendo flexibilidad en el tamaño.

Tendremos los siguientes IDs:

- Mensaje:
 - Petición de escritura de datos: 0x01
- Datos:
 - Servo pulgar 1: 0x01
 - Servo pulgar 2: 0x02
 - Servo pulgar 3: 0x03
 - Servo índice 1: 0x04
 - Servo índice 2: 0x05
 - Servo corazón 1: 0x06
 - Servo corazón 2: 0x07
 - Servo anular 1: 0x08
 - Servo anular 2: 0x09
 - Servo meñique 1: 0x0A
 - Servo meñique 2: 0x0B

El tamaño del mensaje se calcula como el tamaño total del mensaje menos los bytes de comienzo y final.

$$SIZE = \text{Tamaño total del mensaje en bytes} - 2 \quad (3.10)$$

El CRC checksum se calcula sumando el valor de todos los bytes, aislando el byte menos significativo, y restando al máximo sin signo de un byte el valor obtenido de la suma y sumando 1.

$$CRC = 255 - \left[\left(\sum_{i=0}^{N=SIZE} b \right) \bmod 256 \right] + 1 \quad (3.11)$$

Para cada valor en decimal del byte b del mensaje enviado.

```
def get_checksum(msg):
    checksum = 0
    for i in range(1, len(msg) - 2):
        checksum += msg[i]
    checksum = checksum & 0xFF
    checksum = 0xFF - checksum + 1
    return checksum
```

El código de error no es relevante, ya que no se usa en este contexto, y puede igualarse a cero.

- Módulo de conversión de posiciones a ángulos: transforma cada una de las posiciones definidas en el apartado del controlador Machine Learning a los ángulos de la mano correspondientes.

```
+---controller
|       pos_to_gdl11.py
|
```

Se han realizado adicionalmente los siguientes módulos y programas de Arduino para probar el funcionamiento de la mano:

```
+---controller
|       pos_to_gdl11.py
|
+---youbionic_arm
|       youbionic_arm.ino
|
\---youbionic_test
|       youbionic_test.ino
```

Por último, se han incluido unos programas de Arduino para el control y prueba de la mano InMoov, la cual no llegó a usarse en este trabajo:

```
+---inmoov_arm
|       inmoov_arm.ino
|
+---inmoov_test
|       inmoov_test.ino
|
```

3.5. Desarrollo software “arm_prosthesis”

Para agrupar todos estos módulos se ha creado un módulo principal, *arm_prosthesis*, el cual engloba todos los módulos anteriores, a excepción del control implementado en Arduino.

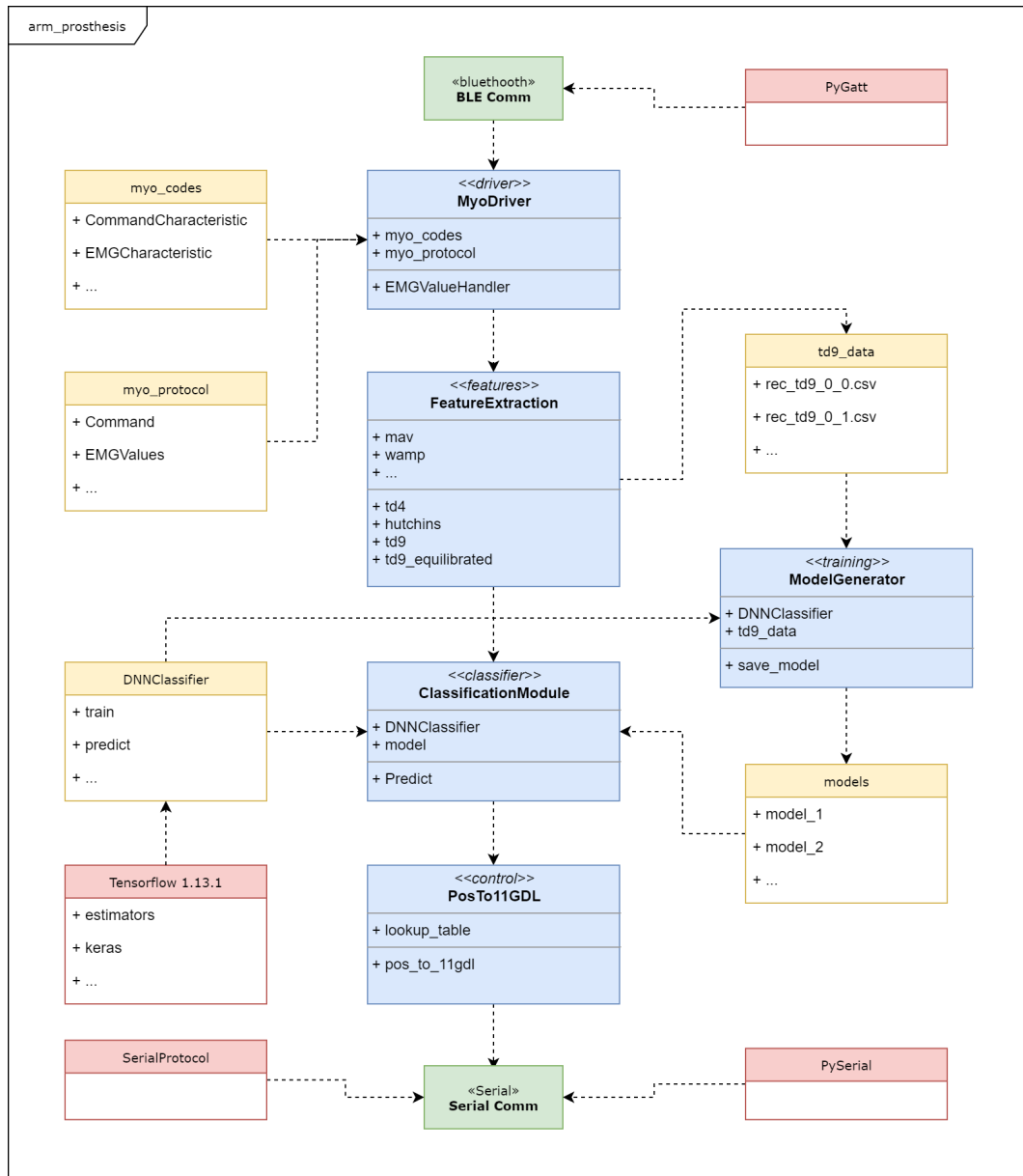


Figura 3.32 : diagrama UML del módulo software *arm_prosthesis*

Entradas y salidas del sistema:

- BLE Connector: conector Bluetooth de baja energía, implementado mediante la librería *PyGatt* para Python en sistemas Linux
- Serial Connector: conector de puerto serial para el envío de mensajes de posiciones angulares de la prótesis, usando la librería *PySerial* para Python

Módulos principales:

- MyoDriver: es capaz de obtener las señales del conector BLE y exponer las características y los datos que el dispositivo proporciona. Usa las clases *myo_codes* y *myo_protocol* para crear la interfaz de obtención de datos, a través de la función manejadora de datos EMG, *EMGValueHandler*
- FeatureExtraction: obtiene las características de las gráficas EMG y mediante funciones matemáticas devuelve un vector de características a partir de un vector de datos EMG
- ModelGenerator: se alimenta de datos de características TD9 para diferentes posiciones y entrena un *DNNClassifier* para su uso en tiempo real por el *ClassificationModule*, el cual predice las posiciones
- ClassificationModule: obtiene, a partir de los datos característicos del EMG en tiempo real, una estimación de la intención de posición del usuario, usando la clase *DNNClassifier* y el modelo entrenado para la misma
- PosTo11GDL: transforma un número de posición en un vector de 11 ángulos para su alimentación directa al controlador mediante puerto serie

Clases adicionales:

- *myo_codes*: códigos de características, UUIDs, etc. usados por la Myo Armband para la comunicación BLE
- *myo_protocol*: API de mensajería para la obtención de datos y envío de comandos a la Myo Armband
- *td9_data*: datos de características TD9 guardados para su uso en entrenamiento de modelos
- *models*: modelos entrenados guardados para su uso en la estimación de posiciones

- DNNClassifier: clase estimador usada para implementar el controlador Machine Learning

El desarrollo software completo se encuentra en el repositorio “*arm_prosthesis*”, bajo licencia MIT, en el siguiente enlace: github.com/darealcipher/arm_prosthesis.

Capítulo 4

Conclusiones

Una vez completado el desarrollo y probado el sistema completo, se han llegado a las siguientes conclusiones concernientes a los dispositivos elegidos y la implementación del sistema:

- El dispositivo Myo Armband es un excelente paquete de captura de datos para biónica, pero carece de la sensibilidad, ancho de banda y autonomía necesaria para funcionar como capturador de señales para prótesis. La dificultad de clasificación de más de 7 posiciones diferentes y la difícil diferenciación y gran variabilidad de algunas de las características evidencian las carencias del dispositivo.
- El clasificador elegido y la implementación realizada carecen de una fiabilidad aplicable a un producto final, en parte debida al punto anterior. Además, mediante el uso del equipo descrito en el desarrollo se han logrado tiempos de predicción no inferiores a 300ms, más del doble del período de solapamiento usado en las señales, por lo que hay conjuntos de datos que no son procesados a tiempo, aunque el sistema no deja en cola los mismos, representando la posición del brazo en el momento dado.
- El dispositivo elegido carece de un control de fuerza, lo que lo hace inviable para un agarre suave y consistente de objetos para la realización de tareas de vida diaria, y es ligeramente más grande que una mano humana de tamaño medio, aunque posee una gran ventaja de aplicación, ya que el diseño se puede adaptar muy fácilmente a amputados transradiales.

El trabajo actual permite su uso en entornos de prueba y en entornos controlados, por lo que no podría considerarse que la prótesis es completamente autónoma,

aunque requiere de un estudio de viabilidad en grupos de sujetos con amputaciones transradiales, el cual escapa el alcance del trabajo.

Capítulo 5

Lineas futuras

Mencionadas estas conclusiones, cabe destacar que el objetivo de este trabajo es eminentemente académico, con la pretensión de generar un trabajo no pionero pero sólido, y con el objetivo de expandir sobre el mismo en futuros desarrollos, ya que el sistema implementado y el uso de características de EMG servirán como base para el uso de diferentes dispositivos mecánicos y diferentes sensores, incluso la aplicación de modelos más eficientes y precisos que el usado en este trabajo. Quedan asentadas las bases para desarrollos en la siguientes líneas generales:

- Diseños mecánicos: uso de mecanismos biomiméticos, e investigar en profundidad el uso de actuadores electropoliméricos, por su atractivo de tamaño, tipo de actuación y fuente de energía usada.
- Sensorización: investigar sobre la capacidad de implementación de un sistema de posicionamiento de mano diestro y con con ángulos de posicionamiento continuo mediante diferentes tipos de electrodos, centrándose en electrodos invasivos implantables (IMESs).
- Control: uso de dispositivos hápticos que capten la fuerza ejercida y puedan regular la fuerza de actuación según la misma, implementando sistemas de lazo cerrado.

El trabajo actual puede ser continuado implementando el sistema completo como prototipo completamente autónomo, con un sistema de alimentación y un sistema de procesamiento suficientemente potente para ejecutar el controlador Machine Learning, el cual, como se ha dicho en versiones anteriores, puede basarse en los sistemas Jetson de Nvidia [38]. También podría mejorarse usando redundancia de sensores, duplicando la entrada de datos y haciendo menos sensible a ruidos en

sistema, pero este enfoque aumenta el peso y necesidad de capacidad de procesamiento, debido al aumento de datos de entrada al clasificador.

Por último, y una vez el trabajo hubiera progresado del estado de prototipo, se plantea la posibilidad de aplicación en amputados transradiales, análisis de su funcionamiento y posible uso por éste tipo de población como prótesis, lo cual requeriría inicialmente la búsqueda de voluntarios dentro de este grupo de sujetos, produciendo estudios de viabilidad en cuanto su uso continuado, destreza, utilidad evaluada por el usuario, etc.

Bibliografía

- [1] Sridhar Poosapadi Arjunan and Dinesh Kant Kumar. Decoding subtle forearm flexions using fractal features of surface electromyogram from single and multiple sensors. *Journal of NeuroEngineering and Rehabilitation*, 2010.
- [2] Yoseph Bar-Cohen. Artificial muscles using electroactive polymers (eap): Capabilities, challenges and potential. *JPL/Caltech*, 2005.
- [3] Open Bionics. Página oficial de open bionics. <https://openbionics.com/>. Accedida: 2019-07-12.
- [4] Claudio Castellini, Katharina Hertkorn, Mikel Sagardia, David Sierra González, and Markus Nowa. A virtual piano-playing environment for rehabilitation based upon ultrasound imaging. *5th IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, 2014.
- [5] Federico Ciccarese. *Youbionic Handy Handbook*. Youbionic, 2019.
- [6] Federico Ciccarese. Antebrazo robótico human arm. <https://www.youbionic.com/store/human>. Accedida: 2020-06-21.
- [7] Federico Ciccarese. Conceptos y dispositivos imprimibles en 3d aplicables a biónica. <https://www.youbionic.com/>. Accedida: 2020-06-20.
- [8] Federico Ciccarese. Mano robótica handy. <https://www.youbionic.com/handy>. Accedida: 2020-06-21.
- [9] Federico Ciccarese. Página de compra de mano robótica handy. <https://www.youbionic.com/store/buyhandy3d>. Accedida: 2020-06-21.
- [10] Ana Cignal, Víctor Lobo, Víctor Moreno, Juan C. Fraile, Rubén Alonso, and Javier P. Turiel. Robhand, un exoesqueleto de mano para la rehabilitación neuromotora aplicando terapias activas y pasivas. *Actas de las XXXIX Jornadas de Automática*, 2018.

- [11] Amputee Coalition. Coalición de amputados en estados unidos. <https://www.amputee-coalition.org/>. Accedida: 2020-06-24.
- [12] Francisco de Borja de la Quintana Gordon; Carlos Chamorro; Antonio Planas; Eloísa López. *Monitorización en anestesia, cuidados críticos y medicina de urgencias*. Elsevier, 2004.
- [13] e Nable. Proyecto e-nable. <http://enablingthefuture.org/>. Accedida: 2020-06-22.
- [14] Richard F., FF. Weir, Phil R. Troyk, Jack F. Schorsch, and Huub Maas. Implantable myoelectric sensors (imes) for intramuscular electromyogram recording. *National Institutes of Health (NIH)*, 2011.
- [15] Festo. Músculos biomiméticos neumáticos de festo. <https://www.festo.com/group/es/cms/10248.htm>. Accedida: 2020-06-22.
- [16] Gadgetversus. Benchmark procesadores en gadgetversus.com. <https://gadgetversus.com/processor/amd-ryzen-7-3700x-gflops-performance/>. Accedida: 2020-06-23.
- [17] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989.
- [18] Google. Curso intensivo de aprendizaje automático (machine learning). <https://developers.google.com/machine-learning/crash-course/>. Accedida: 2020-06-24.
- [19] Hassan Mohamed Hassan, Abdelazim M Negm, Mohamed Zahran, and Oliver Saavedra. Assessment of artificial neural network for bathymetry estimation using high resolution satellite imagery in shallow lakes: Case study el burullus lake. *ResearchGate*, 2015.
- [20] J.R.M. Hosking. L-moments: analysis and estimation of distributions using linear combinations of order statistics. *Journal of the Royal Statistical Society*, 1990.
- [21] A. Kargov, C. Pylatiuk, R. Oberle, H. Klosek, T. Werner, W. Roessler, and S. Schulz. Development of a multifunctional cosmetic prosthetic hand. *IEEE*, 2007.
- [22] Thalmic Labs. Documentos de soporte de myo armband. <https://support.getmyo.com/hc/en-us>. Accedida: 2020-06-20.

- [23] Thalmic Labs. Protocolo ble del myo armband. <https://developerblog.myo.com/myo-bluetooth-spec-released/>. Accedida: 2020-06-20.
- [24] Thalmic Labs. Página oficial de thalmic labs. <https://developerblog.myo.com/author/thalmic-labs/>. Accedida: 2020-06-20.
- [25] Thalmic Labs. Página web de diagnósticos para myo armband. <http://diagnostics.myo.com/>. Accedida: 2020-06-20.
- [26] Thalmic Labs. *Lesson One: Getting Started with Myo*. Thalmic Labs, 2015.
- [27] Gaël Langevin. Robot de tamaño humano inmoov. <http://inmoov.fr/>. Accedida: 2019-07-12.
- [28] LinMot. Motores lineales, fabricante linmot. <https://linmot.com/products/linear-motors/>. Accedida: 2020-06-22.
- [29] Bartjan Maat, Gerwin Smit, Dick Plettenburg, and Paul Breedveld. Passive prosthetic hands and tools: A literature review. *International Society For Prosthetics And Orthotics (ISPO)*, 2017.
- [30] Brahim Mebarki, Draoui Belkacem, Tahri Mohammed Béchar, Lakhdar Rahmani, Tahri Mohammed Béchar, Boumediene Allaoua, and Tahri Mohammed Béchar. Electric automobile ni-mh battery investigation in diverse situations. *ResearchGate*, 2013.
- [31] Nissha Medical. Electrodo silverest™ de nissha. <https://hs.nisshamedical.com/Electrodes/SilveRest.aspx>. Accedida: 2020-06-23.
- [32] CNSac Medshop. Consumibles y accesorios para cardiología cnsac. <https://www.cnsac-medshop.com/es/>. Accedida: 2020-06-23.
- [33] Federico Montagnani, Marco Controzzi, and Christian Cipriani. Preliminary design and development of a two degrees of freedom passive compliant prosthetic wrist with switchable stiffness. *IEEE*, 2013.
- [34] Federico Montagnani, Marco Controzzi Gerwin Smit, Christian Cipriani, and Dick H. Plettenburg. A passive wrist with switchable stiffness for a body-powered hydraulically actuated hand prosthesis. *IEEE*, 2017.

- [35] Rahim Mutlua, Gursel Alici, Xingcan Xiang, and Weihua Li. Electro-mechanical modelling and identification of electroactive polymer actuators as smart robotic manipulators. *Elsevier*, 2014.
- [36] NORTH. Página oficial de north. <https://www.bynorth.com/>. Accedida: 2020-06-20.
- [37] Numerentur. Historia del desarrollo de las svm. <http://numerentur.org/svm/>. Accedida: 2020-06-24.
- [38] Nvidia. Ordenador embebido para aa jeston nano. <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>. Accedida: 2020-06-23.
- [39] OpenBCI. Página tienda de openbci. <https://shop.openbci.com/>. Accedida: 2020-06-23.
- [40] Ottobock. Cubrebrazo cosmético de silicona de ottobock. <https://www.ottobockus.com/prosthetics/upper-limb-prosthetics/>. Accedida: 2020-06-22.
- [41] Angkoon Phinyomark, Rami N. Khushaba, and Erik Scheme. Feature extraction and selection for myoelectric control based on wearable emg sensors. *Sensors*, 2018.
- [42] Angkoon Phinyomark, Pornchai Phukpattaranont, and Chusak Limsakul. Feature reduction and selection for emg signal classification. *Elsevier*, 2012.
- [43] Angkoon Phinyomark, Pornchai Phukpattaranont, and Chusak Limsakul. Fractal analysis features for weak and single-channel upper-limb emg signals. *Elsevier*, 2012.
- [44] Raspberry Pi. Ordenador de escritorio embebido raspberry pi. <https://www.raspberrypi.org/>. Accedida: 2020-06-23.
- [45] Rangaraj M. Rangayyan. *Biomedical Signal Anaysis: A Case-Study Approach*. IEEE, Wiley Inter-Science, 2002.
- [46] Araceli Guadalupe Santana Rayo, Luis Hector Hernandez Gomez, Alejandro Tonatiu Velázquez Sánchez, J. A. Beltrán-Fernández, Juan Alejandro Flores Campos, Guillermo Urriolagoitia Calderón, Víctor Manuel Santana Rayo, and Arturo Enrique Flores Peñaloza. Design and manufacturing of a dry electrode for emg signals recording with microneedles. *ResearchGate*, 2018.

- [47] Techpowerup. Benchmark gpu rtx2080 super en techpowerup.com. <https://www.techpowerup.com/gpu-specs/geforce-rtx-2080-super.c3439>. Accedida: 2020-06-23.
- [48] Tensorflow. Paquete de software de aprendizaje automático tensorflow. <https://www.tensorflow.org/>. Accedida: 2020-06-23.
- [49] Scott Tolksdorf. Repositorio de github del proyecto myo.js. <https://github.com/thalmiclabs/myo.js>. Accedida: 2020-06-20.
- [50] P. Visconti, F. Gaetani, G.A. Zappatore, and P. Primiceri. Technical features and functionalities of myo armband: An overview on related literature and advanced applications of myoelectric armbands mainly focused on arm prostheses. *Exeley*, 2018.
- [51] Zhe Xu, Yoky Matsuoka, and Ashish D. Deshpande. Crocheted artificial tendons and ligaments for the anatomically correct testbed (act) hand. *IEEE Conference on Robotics and Biomimetics*, 2015.
- [52] Zhe Xu and Emanuel Todorov. Design of a highly biomimetic anthropomorphic robotic hand towards artificial limb regeneration. *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.
- [53] Qifan Zhou, Hai Zhang, Zahra Lari, Zhenbo Liu, and Naser El-Sheimy. Design and implementation of foot-mounted inertial sensor based wearable electronic device for game play application. *ResearchGate*, 2016.

Apéndice A

Anexos

A.1. BOMs

- Youbionic Handy:

Nombre	N°	Ref Dir.	Cant.	Coste	
				Ud.	Total
Piezas 3D	-	youbionic	1	10€	10€
Tornillo cabeza avellanada Hex M3x20	32811514698	aliexpress	10	0,027€	0,27€
Tornillo cabeza avellanada Hex M3x15	32811514698	aliexpress	23	0,027€	0,62€
Tornillo cabeza avellanada Hex M3x10	32811514698	aliexpress	6	0,027€	0,16€
Tornillo cabeza redondeada Philips M1.5x6	32811514698	aliexpress	11	0,027€	0,29€
Servomotor SG-90	32968639735	aliexpress	11	0,71 €	7,81€

Cuadro A.1 : BOM mano mecánica *Handy* de Youbionic

- Circuito de alimentación:

Nombre	Nº	Ref Dir.	Cant.	Coste	
				Ud.	Total
Placa de matriz RE200-LF	RE200-LF	RS - 518-6610	1	7,28€	7,28€
Portapilas para PCB para 1 pila AA	611-9576	RS - 611-9576	4	0,62€	2,48€
Cabezal de pines, 3 pines	251-8092	RS - 251-8092	11	0,168€	1,85€
Conector hembra para PCB Samtec Recta 16 pines	SSQ-116-01-G- S	RS - 765-5556	2	4,02€	8,04€

Cuadro A.2 : BOM placa de alimentación

A.2. Estudio económico

- Coste de desarrollo:

Actividad	Resultado	Horas	Coste
Desarrollo módulos Arduino	Programas Arduino	10	227.27€
Desarrollo módulo Sensorización	Módulos Python de captación de datos de Myo Armband	35	795.45€
Desarrollo módulo Machine Learning	Módulos Python de procesado de datos con Tensorflow	35	795.45€
Desarrollo módulo de control	Módulos Python de procesado de datos de posición	15	340.91€
Toma de datos de EMG	Ficheros de datos EMG procesados para entrenamiento	5	113.64€
Entrenamiento módulo Machine Learning	Modelos específicos entrenados	6	136.36€
Montaje mano Youbionic	Prótesis electromecánica <i>Handy</i>	4	90.91€
Montaje placa circuito de alimentación	Placa circuito de alimentación	4	90.91€

Coste P/M: 4000€

Coste total

2590.91€

Cuadro A.3 : costes de desarrollo

- Coste de fabricación mediante impresión 3D:

Creality CR-10 S4	
Coste material	19.99€
Coste Luz	0.12€
Consumo medio	1000W
Coste impresora	1000€
Tiempo amortización (años)	5
Horas de impresión diarias (h)	22
Tasa de fallos (%)	2 %
Mano de obra	18.00€

Cuadro A.4 : costes de operación de la máquina Creality CR-10 S4

Pieza	Peso (g)	Tiempo impresión (h)	Tiempo pre/post procesamiento (h)	Coste
Handy_Back - Right	22	3.3	1	19.30€

Coste total:	19.30€
--------------	--------

Cuadro A.5 : coste de piezas impresas de la máquina Creality CR-10 S4

Anet A8-M	
Coste material	19.99€
Coste Luz	0.12€
Consumo medio	200W
Coste impresora	1000€
Tiempo amortización (años)	5
Horas de impresión diarias (h)	18
Tasa de fallos (%)	2 %
Mano de obra	18.00€

Cuadro A.6 : costes de operación de la máquina Anet A8-M

Pieza	Peso (g)	Tiempo impresión (h)	Tiempo pre/post procesamiento (h)	Coste
Handy_Palm and Rotation - Right	53	7.05	1	19.84€
Wearable Part Complete	98	12.37	0.75	16.46€
Handy_Dock Station	64	7.65	0.5	10.91€
Handy_Thumb	26	3.43	0.5	9.90€
Handy_Index	30	3.83	0.5	10.01€
Handy_Middle	32	4.03	0.5	10.06€
Handy_Ring	30	3.83	0.5	10.01€
Handy_Pinkie	29	3.68	0.5	9.98€

Coste total:	87.20€
--------------	--------

Cuadro A.7 : coste de piezas impresas de la máquina Anet A8-M

■ Coste de materiales:

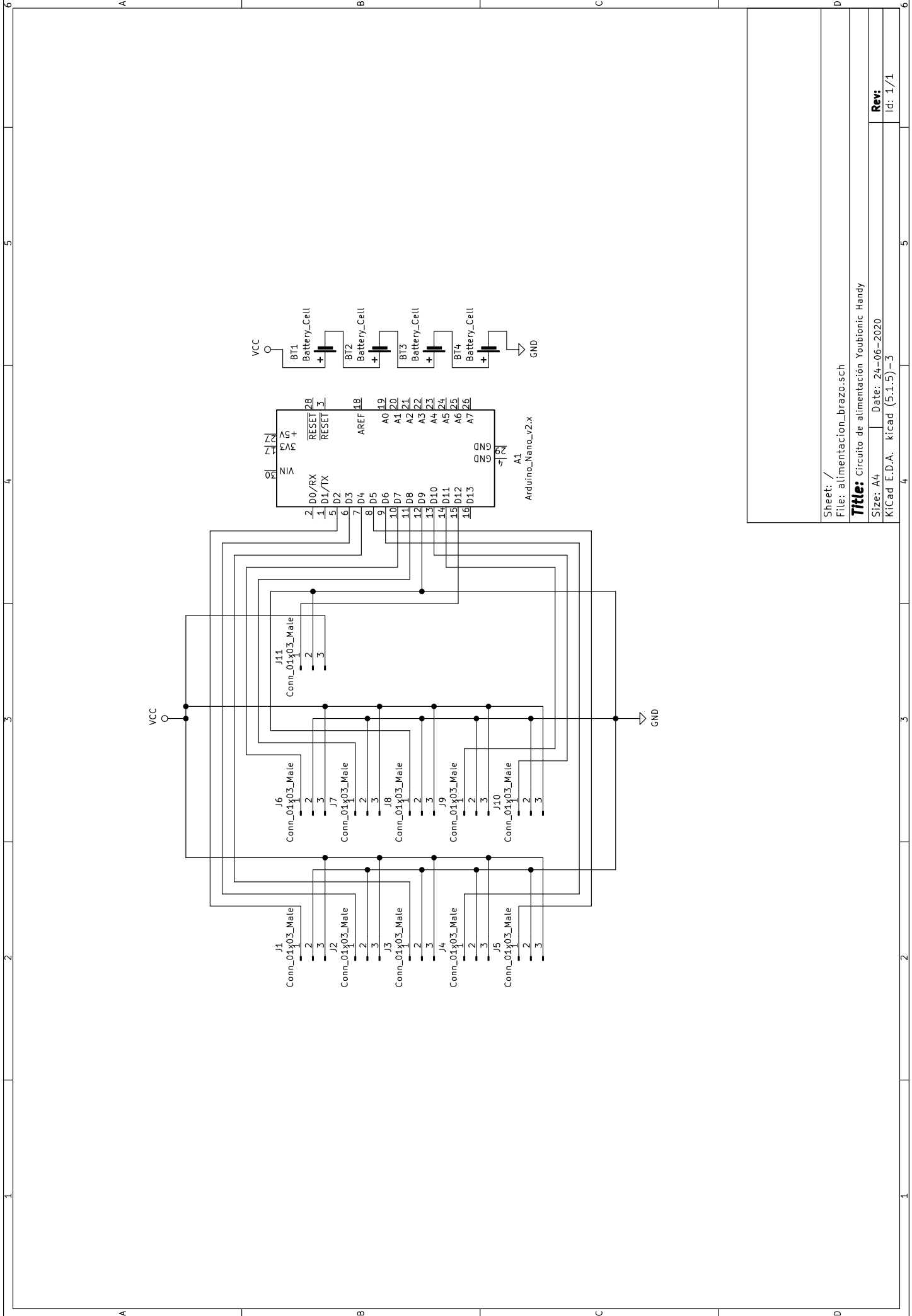
Material	Tipo	Uds.	Coste ud.	Coste
Modelos 3D	Inventariable	1	79.00€	79.00€
Tornillo cabeza avellanada Hex M3x20	Fungible	10	0.03€	0.27€
Tornillo cabeza avellanada Hex M3x15	Fungible	23	0.03€	0.62€
Tornillo cabeza avellanada Hex M3x10	Fungible	6	0.03€	0.16€
Tornillo cabeza redondeada Philips M1.5x6	Fungible	11	0.03€	0.30€
Servomotor SG-90	Fungible	11	0.71€	7.81€
Placa de matriz RE200-LF	Fungible	1	7.28€	7.28€
Portapilas para PCB para 1 pila AA	Fungible	4	0.62€	2.48€
Cabezal de pines, 3 pines	Fungible	11	0.17€	1.85€
Conector hembra para PCB Samtec Recta 16 pines	Fungible	2	4.02€	8.04€
Myo Armband	Inventariable	1	149.00€	149.00€
Arduino Nano	Inventariable	1	2.90€	2.90€

Coste total:	259.71€
--------------	---------

Cuadro A.8 : coste de materiales

A.3. Esquemas electrónicos

- Circuito de alimentación:



Sheet: /
 File: alimentacion_brazo.sch
Title: Circuito de alimentacion Youbionic Handy
 Size: A4 | Date: 24-06-2020
 KICad E.D.A. kicad (5.1.5)-3
Rev:
 Id: 1/1