
Interfaz Biomédica para el Control de Sistemas Robóticos Utilizando Señales EEG

Rodrigo Ralda González



UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Interfaz Biomédica para el Control de Sistemas Robóticos
Utilizando Señales EEG**

Trabajo de graduación presentado por Rodrigo Ralda González para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2020

Vo.Bo.:

(f) _____
Dr. Luis Alberto Rivera Estrada

Tribunal Examinador:

(f) _____
Dr. Luis Alberto Rivera Estrada

(f) _____

(f) _____

Fecha de aprobación: Guatemala, de diciembre de 2020.

Prefacio

Esta investigación surgió de la oportunidad de pertenecer a la línea de investigación biomédica de la universidad y mi interés por controlar dispositivos robóticos a partir de señales cerebrales. A medida que la tecnología ha estado avanzando, también lo han hecho los proyectos inclusivos que buscan reinserir a personas con algún tipo de discapacidad a la sociedad. Este es el caso de este proyecto, que espera generar interés en futuras generaciones para aplicar las interfaces cerebro computador en prótesis, equipo de rehabilitación o equipo industrial.

Culminar este trabajo de graduación no hubiera sido posible sin el apoyo de varios. Primero que todo agradecer a Dios por haberme guiado, apoyado y darme fortaleza en los momentos difíciles. Quiero agradecer a mis padres César y Michelle, a mi hermano Javier, abuelos y familia en general por creer en mí desde el principio y siempre apoyarme incondicionalmente con amor. También agradecer a mis amigos y compañeros de estudios y felicitarlos por haber culminado este logro tan importante en nuestro desarrollo profesional. Por último, agradecer a la Universidad del Valle de Guatemala y al grupo de profesionales que contribuyeron con el desarrollo de mi carrera, especialmente al Dr. Luis Rivera por su dedicación y paciencia en compartirme sus conocimientos y guiar me en este trabajo de graduación. Pido disculpas si olvido mencionar a alguien, a todos ustedes les reitero mi más grande agradecimiento.

Índice

Prefacio	III
Lista de figuras	VII
Lista de cuadros	VIII
Resumen	IX
Abstract	X
1. Introducción	1
2. Antecedentes	3
3. Justificación	5
4. Objetivos	7
4.1. Objetivo general	7
4.2. Objetivos específicos	7
5. Alcance	8
6. Marco teórico	10
6.1. Ondas Cerebrales	10
6.2. Electroencefalograma	10
6.3. Sistema de posicionamiento 10-20	11
6.4. Interfaz Cerebro Computador	12
6.4.1. Tipos de BCI	13
6.5. Aprendizaje automático	13
6.5.1. Máquina de Soporte Vectorial	14
6.5.2. Redes Neuronales Artificiales	15
6.6. Extracción de características	16
6.6.1. Valor Absoluto Promedio	16
6.6.2. Número de Cruces por Cero	16

6.6.3.	Varianza	16
6.6.4.	Curtosis	17
6.6.5.	Potencia Media	17
6.6.6.	Distancia Máxima Mínima	17
6.6.7.	Ondículas	17
6.7.	Manipulador Serial	18
6.8.	Procesamiento de Señales	18
6.9.	Filtrado de Señales	19
7.	Lectura de señales	22
7.1.	Planteamiento del Proyecto	22
7.2.	Búsqueda Base de Datos	23
7.2.1.	Base de datos EEG Motor Movement/Imagery	23
7.2.2.	Base de datos Sleep-EDF Database	23
7.3.	Procesamiento de señales	24
7.3.1.	Adquisición de señales	24
7.3.2.	Filtrado de señales	24
8.	Clasificadores	26
8.1.	Características en el Dominio del Tiempo	26
8.2.	Características en el Dominio de tiempo frecuencia	30
8.2.1.	Clasificador 3 clases	30
8.2.2.	Clasificador 2 clases	32
8.2.3.	Diferentes canales 3 clases	33
8.3.	Base de datos de sueño	35
8.3.1.	Varianza	35
8.3.2.	Distancia Máxima Mínima	36
8.3.3.	Ondículas	37
9.	Visualización Gráfica	39
9.1.	Robot	39
9.1.1.	R17	40
9.1.2.	UR5	43
9.1.3.	Puma 560	44
9.2.	Interfaz Gráfica	46
9.2.1.	Control Manual	47
9.2.2.	Control Automático	48
10.	Validación de resultados	50
11.	Conclusiones	54
12.	Recomendaciones	55
13.	Bibliografía	56
14.	Glosario	59

Lista de figuras

1.	Ubicación electrodos según estándar 10-20 extendido [5].	12
2.	Proceso para BCI convencionales [3]	13
3.	Representación datos de entrada en dos dimensiones y transformados [15]. . .	14
4.	Hiperplano para datos en tres dimensiones [15].	15
5.	Proceso de RNA [3]	15
6.	Tipos de juntas, q_1 revoluta y q_2 prismática [23]	18
7.	Señal continua y discreta [24]	19
8.	Diagrama de bloques Procesamiento Digital de Señales [25]	19
9.	Respuesta ideal cuatro tipo de filtros selectores de frecuencia. (a) Filtro pasa bajas (b) Filtro pasa altas (c) Filtro pasa banda (d) Filtro rechaza banda [26].	20
10.	Respuesta en amplitud para diferentes tipos de filtros [28]	21
11.	Diseño de filtro rechaza banda en el <i>Filter Designer</i> de MATLAB	25
12.	Gráfica amplitud respecto al tiempo y espectro de frecuencia antes y después del filtrado.	25
13.	Matriz de confusión para SVM con kernel de función de base radial con características ZC y MAV.	27
14.	Matriz de confusión para RNA con características ZC y MAV.	27
15.	Matriz de confusión para SVM con kernel de polinomio grado tres con características varianza y curtosis.	28
16.	Matriz de confusión para RNA con características varianza y curtosis.	28
17.	Matriz de confusión para SVM con kernel de polinomio grado tres con las cuatro características (ZC, MAV, k y V).	29
18.	Matriz de confusión para RNA con las cuatro características (ZC, MAV, k y V).	29
19.	Matriz de confusión para KNN con ondículas para canales C3 y C1.	30
20.	Matriz de confusión para SVM grado 3 con ondículas 4 canales (C3, C1, C6 y Cp5).	31
21.	Matriz de confusión para discriminante lineal con ondículas 8 canales (C3, C1, C6, Cp5, T9, T10, Poz y Po4).	31
22.	Matriz de confusión para RNA con ondículas para canales C3 y C1.	31

23.	Matriz de confusión para RNA con ondículas 8 canales (C3, C1, C6, Cp5, T9, T10, Poz y Po4).	32
24.	Matriz de confusión para SVM grado 3 con ondículas para clases 0 y 1.	33
25.	Matriz de confusión para SVM grado 3 con ondículas para canales O1 y P3.	33
26.	Matriz de confusión para RNA con ondículas para canales O1 y P3.	34
27.	Matriz de confusión para SVM grado 3 con ondículas para canales Pz y Fpz.	34
28.	Matriz de confusión para RNA con ondículas para canales Pz y Fpz.	35
29.	Matriz de confusión para SVM grado 3 con 5 características (MAV, ZC, bp, k y v).	36
30.	Matriz de confusión para RNA con 5 características (MAV, ZC, bp, k y v).	36
31.	Matriz de confusión para SVM grado 3 con 5 características (MAV, ZC, bp, k y MMD).	37
32.	Matriz de confusión para RNA con (MAV, ZC, bp, k y MMD).	37
33.	Matriz de confusión para SVM grado 3 con ondículas.	38
34.	Matriz de confusión para RNA con ondículas.	38
35.	Manipulador Serial R17 de 5 GDL (más carrera) [34].	40
36.	Configuración de juntas según clase para R17.	42
37.	Juntas manipulador serial UR5 [35].	43
38.	Configuración de juntas según clase para UR5.	44
39.	Juntas manipulador serial Puma 560 [37].	45
40.	Configuración de juntas según clase para Puma 560.	46
41.	Pestaña modo manual.	47
42.	Pestaña modo automático.	49
43.	Control Manual por Juntas para R17, seleccion comando 2	51
44.	Control Manual por trayectorias para UR5, selección comando 3	51
45.	Control Automático de 3 segundos al detectar ruido.	52
46.	Control Automático de 5 segundos.	53

Lista de cuadros

1.	Clasificación Ondas Cerebrales, [13] y [5].	11
2.	Parámetros de Denavit-Hartenberg para manipulador R17 de 5 GDL (más carrera).	41
3.	Configuración de juntas R17 para cada clase.	41
4.	Configuración de juntas UR5 para cada clase.	43
5.	Configuración de juntas Puma 560 para cada clase.	45

Resumen

Este proyecto de investigación consiste en el desarrollo de una primera fase de una interfaz cerebro-computador que permite controlar sistemas robóticos por medio de señales electroencefalográficas. El proyecto se dividió en cuatro etapas principales. La primera etapa consistió en establecer un mecanismo de adquisición, filtrado y visualización en tiempo real de señales EEG en MATLAB.

La segunda etapa consistió en seleccionar características que redujeran la dimensionaldad de las señales sin afectar la información significativa que estas brindan. Por lo que se calcularon diferentes características como MAV, ZC, curtosis, varianza, potencia media, distancia máxima-mínima y ondículas para probar el rendimiento de 2 algoritmos diferentes de aprendizaje automático supervisado los cuales fueron Redes Neuronales Artificiales y Máquinas de Soporte Vectorial, para encontrar la combinación que menor error presentó. Luego de varias pruebas, se encontró que el mejor rendimiento fue mayor a 80 % y se obtuvo con la máquina de soporte vectorial con kernel polinomial de grado 3 y 5 características en el dominio del tiempo.

La tercera etapa consistió en desarrollar lo gráfico de la BCI. Esto se divide en dos partes, el robot y la interfaz gráfica. Se desarrollaron 3 modelos robóticos los cuales fueron el UR5, el R17 y el Puma 560 con sus respectivas trayectorias para ser accionados por la interfaz. Para la interfaz era una prioridad que esta fuese intuitiva para el usuario y debía mostrar las señales en tiempo real, el resultado de su clasificación y el comando que ejecuta el robot. La cuarta etapa consistió en validar el correcto funcionamiento de las etapas antes mencionadas trabajando conectadas.

Abstract

This research consists of the development of a first phase of a brain-computer interface project that allows to control robotic systems with electroencephalographic signals. The project was divided into four main stages. The first stage consisted in establishing a mechanism for acquisition, filtering and real-time visualization of EEG signals in MATLAB.

The second stage consisted of selecting characteristics that reduce the dimensionality of the signals without affecting the significant information that they provide. Different characteristics such as MAV, ZC, kurtosis, variance, mean power, maximum-minimum distance and wavelets were calculated to test the performance of 2 different types of supervised machine learning algorithms, which were Artificial Neural Networks and Support Vector Machines, to find the combination with the best performance. After several tests, it was found that the best performance up to 80% was presented by the Support Vector Machine with cubic kernel and 5 characteristics.

The third stage consisted of developing the BCI graphics. This is divided into two parts, the robot and the graphical interface. Three robotic models were developed which were the UR5, the R17 and the Puma 560 with their respective trajectories to be activated by the interface. It was a priority that the interface was intuitive with the user and it also had to show the signals in real time, the result of their classification and the command that the robot is executing. The fourth stage consisted of validating the correct operation of all stages working connected.

CAPÍTULO 1

Introducción

La tecnología ha revolucionado la vida de todos. Con el desarrollo de los dispositivos electrónicos ha surgido la necesidad de mejorar la comunicación entre el ser humano y estos. La computadora es un claro ejemplo de ello, razón por la cual se inventó el teclado, el ratón, las interfaces gráficas y las interfaces cerebro computador (*Brain Computer Interface*, BCI). Estas últimas empezaron a desarrollarse en 1973 y se han ido adaptando a diferentes aplicaciones biomédicas, terapéuticas, industriales, de entretenimiento y artísticas. Las BCI han mejorado la forma en que el ser humano se desenvuelve en su entorno realizando actividades de riesgo o precisión en el sector industrial, permitiendo la reinserción en la sociedad de personas con algún tipo de discapacidad y facilitando actividades a la población en general. Este trabajo consiste en diseñar e implementar una BCI no invasiva capaz de controlar dispositivos robóticos. El proceso que realiza una BCI inicia con la captura de señales electroencefalográficas (EEG), el procesamiento, la clasificación de estas y la ejecución de comandos.

Este trabajo de investigación consiste en diseñar e implementar una interfaz cerebro-computador (BCI) que cuente con un mecanismo capaz de capturar, filtrar y visualizar señales EEG en tiempo real. Luego, seleccionar características adecuadas para implementar algoritmos de clasificación automática y en base a esta clasificación enviar los comandos respectivos al sistema robótico que se deseé.

Este documento busca sentar un precedente en la investigación de este tipo de interfaces en Guatemala y presentar resultados que sirvan como base y fundamento para el desarrollo de futuras fases del proyecto. Dichos resultados contribuirán a la comunidad científica internacional en temas de clasificación de señales EEG utilizando inteligencia artificial.

Se espera que este trabajo de investigación genere interés en la región y llame la atención de estudiantes potenciales a contribuir con la calidad de vida de las personas. Los resultados de esta investigación servirán como un primer acercamiento a las BCI para el lector y permitirán continuar con la investigación teniendo un punto de partida definido.

Este documento se divide en 4 capítulos los cuales equivalen a cada módulo trabajado

durante la investigación. En el capítulo 7 se encuentra el proceso de lectura y procesamiento de señales. En el capítulo 8 la extracción de características y la validación de clasificadores. En el capítulo 9 la visualización gráfica de la BCI y en el capítulo 10 la validación de resultados donde todos los módulos trabajan conectados entre sí.

CAPÍTULO 2

Antecedentes

A lo largo de los años los seres humanos hemos estado intentando comprender la complejidad del cuerpo humano. Observando comportamientos, haciendo mediciones y de ahí ha surgido la necesidad de crear dispositivos capaces de medir diferentes tipos de fenómenos. Tal fue la situación de Hans Berger, reconocido neurólogo y psiquiatra alemán que en 1929 tratando de relacionar los procesos psíquicos con algún tipo de medición, logró desarrollar un dispositivo capaz de medir las fluctuaciones espontáneas en el potencial eléctrico del cerebro, conocido actualmente como electroencefalograma humano. Las mediciones realizadas por dicho dispositivo han generado interés en nuevas líneas de investigación, en la década de los 70 se empezó a investigar este tipo de señales con el fin de restaurar daños en la audición, visión y movimiento de las personas. En la década de los 90 se instalaron los primeros dispositivos neuro prostáticos que se basaban en la lectura de señales EEG [1].

A partir de ese momento se ha estado buscando mejorar la manera de interpretar dichas señales y saber diferenciar patrones en ellas. Para diferenciar patrones entre ellas y extraer características es necesario aplicar métodos matemáticos a las señales EEG en crudo. En 1973 surge un interés en la investigación de las interfaces cerebro computador (BCI), con énfasis en los métodos para el procesamiento de datos con el fin de comprender de mejor manera la actividad cerebral por el trabajo de Jacques Vidal de la Universidad de California y en los años 90 se hicieron las primeras pruebas en humanos con métodos no invasivos [1].

Parte importante de estas investigaciones es la recopilación de datos, la cual depende del tipo de equipo a utilizar. El equipo médico y profesional tiene costos muy elevados, por lo que en la investigación se ha utilizado equipo no invasivo de baja fidelidad, pero funcionales, tal es el caso de [1], [2], [3] y [4]. Parte de usar equipo relativamente económico es hacer que los estudios sean accesibles para todos, esta accesibilidad también hace referencia a comunidades de código abierto entre las más grandes de estas se puede mencionar OpenBCI, una empresa dedicada a proveer las herramientas necesarias para medir las señales eléctricas del cuerpo a cualquiera con una computadora. En la página de la empresa se encuentra documentación abierta tanto de software como de hardware para lograr medir las señales eléctricas del cuerpo, tal y como se muestra en [1].

Gran parte de las investigaciones en la última década se han centrado en el procesamiento de las señales y los métodos para extraer las características de estas. Al principio se pensaba que las señales EEG eran valores aleatorios, pero se han ido encontrando diferentes modelos matemáticos en el dominio del tiempo, frecuencia y tiempo-frecuencia capaces de clasificar y diferenciar patrones en las señales [3], [5].

El procesamiento de las señales en crudo es fundamental para lograr controlar robots como en el caso de [6] que controlan un robot o en el caso de [2] y [7] que controlan prótesis de mano y brazo. El fin de controlar estos dispositivos es hacer actividades riesgosas o imposibles para los usuarios e implementarlas en robots controlándolos completamente como si fuese parte del cuerpo humano.

Respecto a la Universidad del Valle de Guatemala, se ha utilizado señales EEG para diferentes aplicaciones. En su mayoría la lectura de estas para una neuro retroalimentación y estimular diferentes partes del cerebro. Prueba de ello son los trabajos realizados en el 2018 [8] y [9] que utilizaron todo el equipo de OpenBCI, tanto la CytonBoard como el Ultracortex para mejorar concentración y reducir estrés en atletas de alto rendimiento. En el 2019 se trabajó con la misma CytonBoard, pero con otro tipo de sensores receptores, en este caso el ElectroCap, para generar pulsos binaurales y lograr así disminuir los desordenes del sueño en los atletas de alto rendimiento [10].

CAPÍTULO 3

Justificación

La población mundial con algún tipo de problema motriz o degenerativo ha ido aumentando en los últimos años. Entre las enfermedades que los afectan cabe mencionar lesiones medulares, infarto cerebral, esclerosis, distrofia muscular, paraplejia y problemas del habla, entre otras. En Guatemala no es diferente el caso. No solamente afectan las enfermedades antes mencionadas, sino también existen amputaciones de miembros o extremidades por enfermedades, accidentes y por la violencia. La mayoría de estos accidentes son de tránsito, por ejemplo, en el año 2017 se registraron 10,354 colisiones solo en la ciudad de Guatemala según el Instituto Nacional de Estadística, la causa de la mayoría es el exceso de velocidad y el uso del teléfono [11]. En cuanto a las enfermedades, gran parte de estas se da por gangrena y diabetes. Las amputaciones por el pie diabético son bastante comunes, médicos españoles aseguran que existe una probabilidad 15 veces mayor de sufrir amputación para los diabéticos en comparación con los no diabéticos [12].

Uno de los principales retos sociales en la actualidad es la atención a este segmento de personas que sufren limitaciones importantes, tienen necesidades especiales y cierto grado de dependencia. Los sistemas de salud y atención social alrededor del mundo han estado invirtiendo en hospitales, centros de rehabilitación y asistencia domiciliaria para este grupo de afectados. Todo esto con el fin de mejorar la calidad de vida de los antes mencionados, mejorar su bienestar y darles autonomía en actividades diarias como alimentación, aseo y trabajo. También facilitar su movilidad y comunicación con el mundo [7].

Por otro lado, también tenemos las actividades de riesgo para los seres humanos. Ya sea en ambientes hostiles o situaciones donde se requiera sustituir el trabajo humano por el trabajo robótico por motivos de eficiencia, costos o modernización. Así mismo en el ámbito de la robótica asistencial frecuentemente utilizada en los centros de rehabilitación antes mencionados y asistentes para cirugías, entre otros.

En consecuencia de lo anterior, ha surgido en el mundo la necesidad de investigar la posibilidad de controlar robots con movimientos que parezcan naturales basados en la actividad eléctrica del cuerpo humano. Los estudios con señales electromiográficas (EMG) han mostrado un gran avance y son de las más utilizadas para los casos de prótesis. Sin embargo,

presentan ciertos inconvenientes como el proceso de estudio a cada paciente en específico, por el tipo de problema que tuvo y los músculos de los que se puede obtener la señal. Así mismo, cuando el paciente presenta complicaciones para mover estos músculos o ha pasado mucho tiempo después del accidente, no se pueden captar de manera correcta las señales EMG. Tomando en cuenta también que los métodos no invasivos para obtener señales EMG provocan confusión en la captación de las señales [2].

Tal es el caso de los asistentes robóticos o control de robots en general, donde se necesita que al menos una extremidad este disponible para el control del robot. Lo cual hace que se pierda una extremidad para el trabajo que se esta realizando, o en el caso de pacientes con enfermedades degenerativas ni siquiera se puedan mover estas extremidades. Es por esto que a partir de la década pasada se ha mostrado interés en el control de los dispositivos con señales EEG, las cuales no presentan los problemas anteriormente mencionados.

En la Universidad del Valle de Guatemala se cuenta con dispositivos como el robot R17, dispositivos prostéticos y dispositivos animatrónicos, pero ninguno controlado por señales EEG. Tener control con señales EEG de estos podría ayudar a cursos de carreras como ingeniería Biomédica, Mecatrónica o Electrónica. También se debe tomar en cuenta el interés que estos generarían en estudiantes potenciales y así continuar con las líneas de investigación de varios años del departamento, con la finalidad de entregar soluciones a la sociedad capaces de resolver problemas reales como los antes mencionados.

CAPÍTULO 4

Objetivos

4.1. Objetivo general

Diseñar e implementar una interfaz cerebro-computador (BCI) que permita controlar sistemas robóticos por medio de señales electroencefalográficas (EEG).

4.2. Objetivos específicos

- Establecer un mecanismo de adquisición, filtrado y visualización en tiempo real de señales EEG.
- Seleccionar características adecuadas para el reconocimiento de patrones en señales EEG.
- Evaluar e implementar algoritmos para la clasificación de movimientos o gestos, a partir de características y patrones de señales EEG.
- Desarrollar una BCI que traduzca los resultados de la clasificación de señales en mandos para un sistema robótico.

CAPÍTULO 5

Alcance

Este proyecto consiste en el desarrollo de una primera fase de una BCI, donde se diseña e implementa una BCI no invasiva capaz de controlar dispositivos robóticos y animatrónicos en tiempo real a nivel de simulación. Para controlar dichos dispositivos se tomarán señales EEG de individuos realizando diferentes actividades. Estas señales se procesarán y se probarán con diferentes métodos de clasificación para obtener el mejor rendimiento posible. Los métodos de clasificación anteriormente mencionados se basarán en el reconocimiento de patrones con aprendizaje automático. En base a la clasificación realizada se enviarán comandos a ejecutar por el robot o dispositivo animatrónico al que esté conectado la interfaz. Cabe mencionar que esta fase no es la final del proyecto, se espera que los resultados sirvan como base para implementar el diseño físicamente y lograr controlar el dispositivo robótico con señales EEG recibidas del cerebro humano en tiempo real. El proyecto está dividido en cuatro módulos principales. El primero es el encargado de la obtención y procesamiento de las señales electroencefalográficas en tiempo real, para este módulo se espera observar en tiempo real la actividad cerebral y filtrarla de cualquier ruido o interferencia no deseada. El segundo módulo es la clasificación de las señales. En este módulo se obtendrán diversas características. Las características se utilizarán para reconocer patrones y así clasificar con aprendizaje automático. El tercer módulo a implementar es el desarrollo e implementación de la parte gráfica. Primero se realizará la interfaz gráfica capaz de desplegar las señales en tiempo real y las características obtenidas. Segundo, se desarrollará un simulador capaz de generar el movimiento que se espera realice el robot físico, en base a la actividad cerebral recopilada. El último módulo es la validación del correcto funcionamiento de los 3 módulos antes mencionados funcionando correctamente conectados entre sí.

El desarrollo de los cuatro módulos anteriormente mencionados se vio afectado por la pandemia de COVID-19, a continuación, se detalla de qué forma afectó la situación de la pandemia cada módulo en específico, lo cual repercutió de manera significativa en el alcance que se tenía previsto para este proyecto de investigación. El primer módulo se vio afectado debido a que no se tuvo acceso al equipo de recepción de señales EEG que tiene la Universidad del Valle de Guatemala, por lo que fue imposible utilizar la *OpenBCI* y el *ElectroCap* para capturar las señales en tiempo real.

Respecto al segundo módulo, se tuvo inconvenientes debido a que no se contaba con señales propias. El rendimiento del clasificador dependía del tipo de datos con el que se le entrenará, por lo que al usar diferentes bases de datos se obtuvieron diferentes resultados. Mismo motivo por el cual la investigación parecía no avanzar durante varias semanas pues el clasificador no proporcionaba los resultados esperados, hasta que se optó por cambiar la base de datos y de esta manera ya se obtuvo mejores resultados.

El tercer módulo quizá fue el que menos se vio afectado de los tres, para este módulo se pensaba utilizar el robot R17 de la UVG para fines demostrativos del correcto funcionamiento del proyecto. Sin embargo, debido a la misma situación de la pandemia fue imposible acceder al modelo físico por lo que se procedió a trabajar el modelo a nivel simulación al igual que dos manipuladores seriales más para sustituir lo que al principio se esperaba que fueran el R17 y los dispositivos animatrónicos de la universidad. El cuarto módulo tampoco se vio afectado de gran manera, los métodos para validar el correcto funcionamiento fueron diferentes a lo que se había pensado en un principio del proyecto, pero de igual manera se cumplió con los objetivos planteados.

CAPÍTULO 6

Marco teórico

6.1. Ondas Cerebrales

Las ondas cerebrales son la raíz de todos nuestros pensamientos, emociones y comportamientos. Son la comunicación entre las neuronas de nuestro cerebro. Se producen cuando grupos de neuronas se comunican con otros en forma de pequeños pulsos sincronizados. Las ondas cerebrales varían según la actividad que se esté relacionando, se mide en microvoltios y se clasifican según su frecuencia en (Hz). Las de frecuencias bajas se asocian al sueño y cansancio, mientras que las frecuencias altas con estrés y estado alerta [13]. En el Cuadro 1 se pueden apreciar las clasificaciones que comúnmente se le hacen y el nombre con el que se les conoce, sin embargo en la práctica no es tan sencillo diferenciarlas.

6.2. Electroencefalograma

El electroencefalograma es un estudio que monitorea la actividad electrofisiológica y registra la actividad eléctrica del cerebro. Este se relaciona con la actividad neuronal dentro del cerebro. Las señales que registra son conocidas como EEG y son fluctuaciones de voltaje que resultan de la corriente iónica entre las neuronas del cerebro en un periodo de tiempo [5].

Para captar las señales se posicionan electrodos alrededor del cerebro y se registran las diferencias de potenciales entre estos. La captación de estas señales se considera compleja, debido a que son señales estocásticas que presentan grandes variaciones dependiendo de diferentes factores como: la estructura craneal del paciente, el estado mental del mismo, niveles de concentración, movimientos musculares, movimientos involuntarios y del posicionamiento que se les dé a los electrodos [5].

Nombre de la Onda	Frecuencias (Hz)	Descripción
Delta	0.1-4	Son de baja frecuencia y altamente penetrantes. Se generan en alta meditación y etapas profundas del sueño. Tienen gran amplitud y se asocian con regeneración y sanación en el cuerpo. Su amplitud varía entre los 20 y 200 uV.
Theta	4-8	Se presenta en niños mayores a 13 años y en adultos que tienen complicaciones del sueño. También presente en desordenes hidrocefálicos y cuando hay altos niveles de estrés. Su amplitud varía entre los 20 y 100 uV.
Alfa	8-14	Aparece en los lados del cerebro, con una mayor amplitud en el lado dominante. Aparece al cerrar los ojos y relajarse, desaparece al abrirlos. Después de los 30 años aparece frecuentemente en la mayoría de personas. Su amplitud varía entre los 20 y 60 uV.
Beta	14-30	Aparece de manera simétrica en ambos lados del cerebro. Se acentúa con el consumo de drogas sedantes o hipnóticas. Se presenta comúnmente en pacientes con ansiedad. Su amplitud varía entre los 2 y 20 uV.
Gamma	>30	Esta frecuencia se pudo medir hasta que se implementó la tecnología digital, analógicamente era imposible. Esta frecuencia aparece en períodos de hiper alerta y altas percepciones de los sentidos.

Cuadro 1: Clasificación Ondas Cerebrales, [13] y [5].

6.3. Sistema de posicionamiento 10-20

El sistema internacional de posicionamiento es un estándar internacional definido por la Federación Internacional de Sociedades de Electroencefalografía (FISE) en 1958. Describe la ubicación del cráneo donde deben colocarse los electrodos para medir señales EEG. El estándar se estableció en base a la relación entre la localización del electrodo y el área de la corteza cerebral ubicada bajo el, los números 10 y 20 hacen referencia a que la distancia entre los electrodos vecinos puede ser 10 o 20 % de la distancia total entre la zona frontal y posterior, o de derecha a izquierda del cráneo, cada zona tiene asignada una letra para identificar el lóbulo y un número para identificar en qué hemisferio se encuentra, si el número es par pertenece al hemisferio derecho y si es impar al hemisferio izquierdo. Cada lóbulo del cerebro se puede identificar con una letra correspondiente. Para los lóbulos frontal, temporal, central, parietal y occipital sus identificadores son F, T, C, P, O respectivamente [5].

Por lo regular se utilizan cuatro referencias anatómicas para posicionar los electrodos. La primera es el nasión, este es el punto entre el hueso frontal y los dos huesos de la nariz del cráneo. El segundo es el inion, este se encuentra en el punto inferior trasero del cráneo y es la proyección más prominente del hueso occipital. Por último los puntos preauriculares detrás de los oídos [5]. En la Figura 1 se puede apreciar gráficamente la ubicación de los electrodos para un sistema 10-20. El estándar 10-20 es para una cantidad pequeña de electrodos, normalmente 21. Con la necesidad de más ondas y más canales se han establecido nuevos estándares como el 10-10 y 10-5 para 81 y 256 canales, respectivamente.

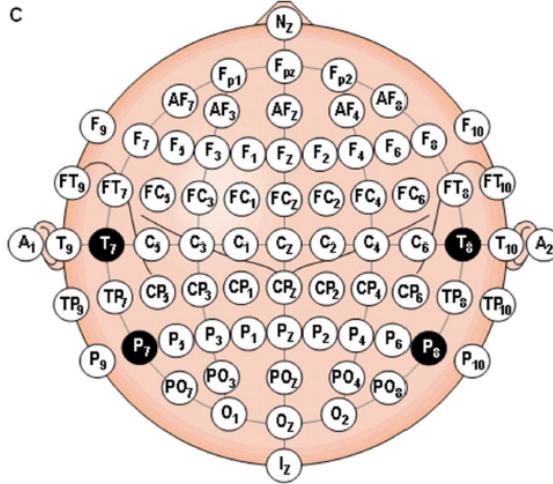


Figura 1: Ubicación electrodos según estándar 10-20 extendido [5].

6.4. Interfaz Cerebro Computador

La interfaz cerebro computador (BCI) por sus siglas en inglés Brain Computer Interface es capaz de medir la actividad cerebral y traducirla para poder procesarla con una computadora. El objetivo principal de estas es monitorear y estudiar la actividad cerebral ante diferentes escenarios. El estudio de la actividad cerebral se logra mediante las señales EEG y la calidad de estas depende del equipo de medición a utilizar. La diferencia en el tipo de señales a obtener depende de los equipos a utilizar. Existe una gran variedad de dispositivos para captar las señales, por un lado, están los dispositivos clínicos y médicos que entregan mejor calidad de señales, pero su costo es bastante elevado. Por otro lado, tenemos los equipos portátiles utilizados comúnmente para proyectos de investigación y desarrollo, pero nunca podrán sustituir a los dispositivos médicos [5].

Un sistema BCI esta conformado por varias etapas tal y como se observa en la Figura 2. La primera de ellas seria la adquisición de señales EEG en crudo, esto se puede realizar en tiempo real o guardar para un análisis posterior. La siguiente etapa sería el preprocesamiento donde se aplican filtros por lo regular para separar frecuencias y eliminar ruido o frecuencias no deseadas como la de la corriente eléctrica (60 Hz).

Luego está la extracción de características lo cual se considera el procesamiento en sí, estas características se obtienen mediante métodos matemáticos como la transformada rápida de Fourier (FFT), transformada de ondículas (Wavelets), transformada de Hilbert-Huang, número de cruces por cero (*Zero Crossing ZC*) o el promedio del valor absoluto (*Mean Absolute Value MAV*).

Siguiente a esto, tenemos la etapa de clasificación e interpretación de las características obtenidas mediante aprendizaje automático. Por último, tenemos la señal de salida al actuador o efecto, dependiendo de la clasificación realizada anteriormente [5].

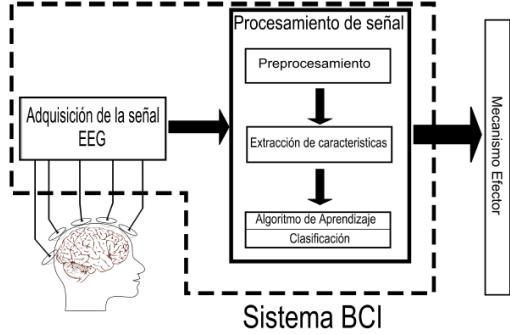


Figura 2: Proceso para BCI convencionales [3]

6.4.1. Tipos de BCI

Invasivo: Necesitan de neurocirugías para poder instalarlos en la materia gris. La señal es de muy buena calidad, pero con el tiempo la señal se va degradando porque el cuerpo lo rechaza y forma cicatrices [5].

Parcialmente invasivo: Se implantan dentro del cráneo, pero no en la materia gris mediante cirugías. Su calidad se considera intermedia, pero no se degradan con el paso del tiempo porque el cuerpo no los rechaza [5].

No invasivo: La mayoría de los trabajos de investigación y desarrollo en los últimos años se realizan con esta tecnología. Son sencillos de instalar, remover y no requieren cirugías. Sin embargo, la calidad de las señales no es tan buena como las mencionadas anteriormente y no se pueden detectar señales de alta frecuencia [5].

6.5. Aprendizaje automático

El aprendizaje automático, comúnmente conocido como *Machine Learning* es el estudio científico de algoritmos y modelos estadísticos que las computadoras utilizan para tomar decisiones sin ser programadas explícitamente. Según el doctor Adeel Najmi es cuando las máquinas pueden crear o refinar autónomamente modelos de decisión [14].

Esta nace como una subdisciplina de la Inteligencia Artificial (IA) que construye modelos matemáticos basado en los datos que se le proporcionan, conocidos como datos de entrenamiento para posteriormente realizar toma de decisiones o predicciones. Entre las aplicaciones principales se encuentran: reconocimiento de imágenes, sonido y texto, diagnósticos médicos, predicción de eventos, minería de datos, robótica, finanzas, telecomunicaciones y biomédica [14].

El proceso de Aprendizaje Automático se podría dividir en dos procesos. El primer proceso de decisión y ejecución. El segundo proceso de aprendizaje por medio de la retroalimentación. Ambas trabajan de la mano para lograr la mejora continua y poder tomar las mejores decisiones o predicciones sin haberlo programado explícitamente [14]. Existen diferentes tipos de algoritmos de aprendizaje automático que se utilizan como clasificadores,

a continuación se detallan dos de ellos.

6.5.1. Máquina de Soporte Vectorial

Una Máquina de Soporte Vectorial conocida con el acrónimo SVM por sus siglas en inglés (*Support Vector Machine*) es un algoritmo de aprendizaje automático supervisado que suele emplearse con fines de clasificación y regresión binaria tal y como se observa en la Figura 3. En la actualidad se implementa el algoritmo para resolver problemas de regresión, agrupamiento y multiclasiﬁcación. Por su alto rendimiento se ha implementado en visión artíﬁcial, reconocimiento de patrones y análisis de series temporales [10].

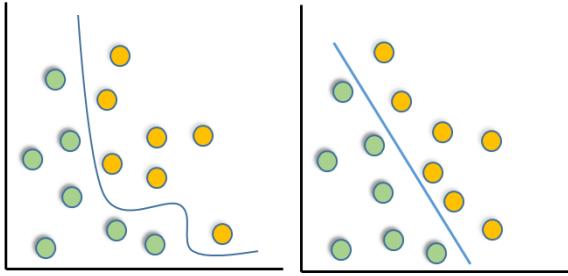


Figura 3: Representación datos de entrada en dos dimensiones y transformados [15].

El algoritmo construye un modelo a partir de ejemplos de entrenamiento ya clasificados, capaz de predecir a qué categoría pertenecerá un nuevo ejemplo. El funcionamiento del SVM se basa en encontrar la mejor línea recta, plano o hiperplano según la dimensión del problema que divide el conjunto de datos en dos clases, lo cual consigue aplicando los siguientes conceptos [15], los cuales se observan en la Figura 4.

- **Hiperplano:** Plano que separa y clasifica linealmente un conjunto de datos.
- **Vectores de apoyo:** Son los puntos críticos de datos más cercanos a la línea del hiperplano y que al eliminarse alterarían la posición del este.
- **Margen:** Distancia entre el hiperplano y el dato más cercano de cualquiera de los conjuntos.
- **Dimensión VC:** Cuyo nombre se deriva de quienes contribuyeron con grandes aportes para los SVM (Vapnik-Chervonekis). VC es el conjunto de funciones disponibles y la dimensión es la cantidad máxima de puntos que se pueden separar de todas las formas posibles mediante el conjunto VC. A mayor dimensión, menor error del conjunto de entrenamiento y mayor confianza.
- **Kernel:** También conocido como núcleo es el conjunto de funciones matemáticas utilizadas para transformar los datos de entrada en la forma deseada. Conectan la linealidad y no linealidad de las SVM, cuando los datos parecen no tener forma de separarlos.

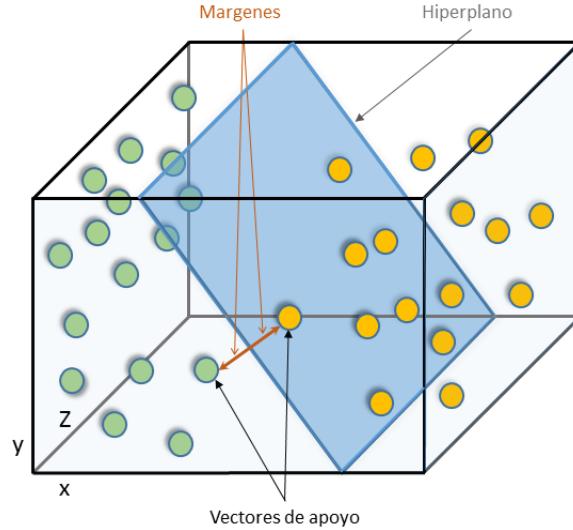


Figura 4: Hiperplano para datos en tres dimensiones [15].

6.5.2. Redes Neuronales Artificiales

Conocidas por el acrónimo RNA, el nombre proviene de sus similitudes en organización, comportamiento y estructura con su homólogo biológico. Por ser un tipo de aprendizaje automático supervisado, esta herramienta aprende y se forma a sí misma tratando de minimizar la función de pérdida que evalúa a la red en su totalidad mediante propagación hacia atrás. Tratando de emular el sistema neuronal biológico, las neuronas artificiales se organizan en capas y las capas en una estructura jerárquica similar a la del cerebro[3], tal y como se aprecia en la Figura 5.

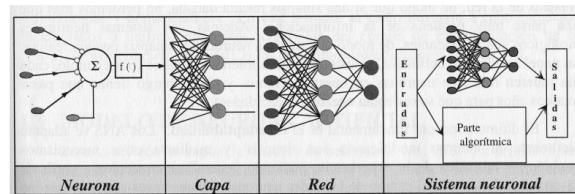


Figura 5: Proceso de RNA [3]

Las redes neuronales artificiales emulan los siguientes conceptos clave:

- **Procesamiento Paralelo:** Debido a la gran cantidad de neuronas que intervienen en un proceso, es posible efectuar procesos en paralelo.
- **Memoria Distribuida:** La información está distribuida por la sinapsis de la red, existiendo redundancia en caso falle o se dañe alguna sinapsis.
- **Adaptabilidad al entorno:** Es posible conceptualizar a partir de casos previos y aprender en base a la experiencia proporcionada por la información de las sinapsis.

Los conceptos claves en los que se basa la red neuronal artificial hace que sea ampliamente utilizada para implementar clasificadores con aprendizaje automático [3].

6.6. Extracción de características

La extracción de características es un proceso por el cual se busca reducir la dimensionalidad de un largo número de datos a grupos más pequeños cuyo procesamiento sea más sencillo, logrando expresar sus características más significativas con una menor cantidad de datos [16]. Para señales bioeléctricas se utilizan diversos tipos de características, a continuación, una breve explicación de las utilizadas en esta investigación.

6.6.1. Valor Absoluto Promedio

El valor absoluto promedio conocido comúnmente como MAV por el acrónimo de sus siglas en inglés *Mean Absolute Value* es el promedio del valor absoluto de la amplitud de la señal [17] y se define como:

$$\text{MAV} = \frac{1}{N} \sum_{n=1}^N |x_n| \quad (1)$$

6.6.2. Número de Cruces por Cero

El número de cruces por cero conocido como ZC por sus siglas en inglés *Zero Crossing* es tal y como su nombre lo indica, la cantidad de veces que la amplitud de una señal cruza el 0 del eje X. En señales bioeléctricas se acostumbra a usar un umbral para evitar que se cuenten cruces de valores menores a dicho umbral los cuales por lo regular representan ruido [17]. El ZC se utiliza como estimador de las propiedades en el dominio de la frecuencia y se define como:

$$\begin{aligned} \text{ZC} &= \sum_{n=1}^{N-1} [\operatorname{sgn}(x_n \times x_{n+1}) \cap |x_n - x_{n+1}| \geq \text{Umbral}]; \\ \operatorname{sgn}(x) &= \begin{cases} 1 & \text{si } x \geq \text{Umbral} \\ 0 & \text{si } x < \text{Umbral} \end{cases} \end{aligned} \quad (2)$$

6.6.3. Varianza

La varianza denotada por la letra V indica que tan cerca o lejos están los datos de la media, lo hace calculando el promedio de las diferencias al cuadrado entre los datos y su media, razón por la cual es muy sensible a datos atípicos [18].

$$V = \frac{1}{N-1} \sum_{n=1}^N |x_n - \bar{x}|^2 \quad (3)$$

6.6.4. Curtosis

La curtosis denotada por la letra k es también conocida como el cuarto momento de una distribución, esta medida está relacionada con los extremos o "colas" de la distribución y tambien es muy sensible a datos atípicos. La curtosis indica la cercanía entre los datos y los valores de tendencia central, una curtosis de 3 indica que la distribución es normal [19]. Se calcula con la siguiente expresión:

$$k = \frac{\frac{1}{N} \sum_{n=1}^N (x_n - \bar{x})^4}{\left(\frac{1}{N} \sum_{n=1}^N (x_n - \bar{x})^2 \right)^2} \quad (4)$$

6.6.5. Potencia Media

La potencia media de una señal es la suma de las muestras absolutas al cuadrado en el dominio del tiempo dividido entre el largo de la señal, equivalente al cuadrado de su nivel RMS [20], es conocida como bp por el acrónimo de sus siglas en inglés.

$$P = \frac{1}{N} \sum_{n=1}^N |x_n|^2 \quad (5)$$

6.6.6. Distancia Máxima Mínima

La distancia máxima-mínima es comúnmente conocida como MMD por sus siglas en inglés *Maximum Minimum Distance* y se basa en la fórmula de distancia derivada del teorema de Pitágoras para encontrar la distancia entre el punto máximo y el mínimo de una ventana determinada [21], se define como:

$$MMD = \sum_{n=1}^N | \sqrt{\Delta x_n^2 + \Delta y_n^2} | \quad (6)$$

6.6.7. Ondículas

Conocidas como *Wavelets*, es una herramienta poderosa en el procesamiento de señales debido a la capacidad de realizar análisis tanto en el tiempo como en la frecuencia de fenómenos estacionarios y no estacionarios. Este principio se basa en la aproximación de señales usando la superposición de señales. La energía de estas ondas se encuentra concentrada en el

tiempo y sirve como herramienta para el análisis de fenómenos transientes, no estacionarios y variantes en el tiempo [22].

6.7. Manipulador Serial

Los manipuladores seriales están formados por una serie de eslabones rígidos y juntas. Cada junta tiene un grado de libertad que puede ser traslacional en el caso que se trate de una junta prismática o rotacional si es una junta revoluta tal y como se observa en la Figura 6. El movimiento de las juntas cambia la pose relativa de los eslabones. Por lo regular un extremo del manipulador serial se encuentra aterrizado y se conoce como bancada, el otro extremo que se encuentra libre se conoce como efecto final [23].

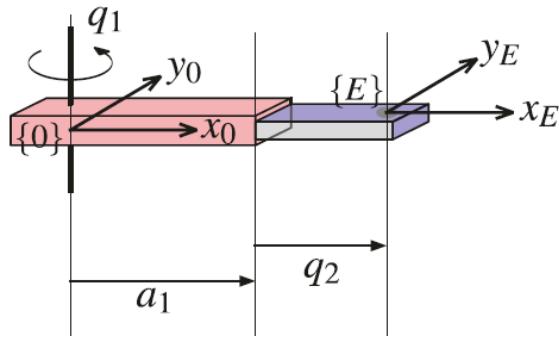


Figura 6: Tipos de juntas, q_1 revoluta y q_2 prismática [23]

Una manera sistemática de describir la geometría de un manipulador serial es con la notación de Denavit-Hartenberg. En los parámetros de Denavit-Hartenberg un eslabón define la relación espacial entre dos juntas vecinas. El eslabón se describe con cuatro parámetros los cuales son:

- θ_j ángulo entre ejes X.
- d_j el largo del eslabón en el eje Z.
- a_j el largo del eslabón en el eje X.
- α_j el ángulo entre ejes Z.

Siempre se deben respetar las dos restricciones siguientes: x_j intersecta a z_{j-1} y x_j es perpendicular a z_{j-1} [23].

6.8. Procesamiento de Señales

Una señal es una descripción formal de un fenómeno evolucionando en el tiempo o espacio. El procesamiento de señales hace referencia a cualquier operación manual o mecánica

que modifique, analice o manipule la información contenida en la señal. El procesamiento de señales se puede hacer de manera analógica o digital, cada una presenta sus ventajas y desventajas. En la actualidad se ha hecho énfasis en el procesamiento digital de señales por su inherente universalidad para diversos tipos de aplicaciones [24].

El procesamiento digital de señales tiene el inconveniente que no se pueden representar las señales como funciones continuas. Esto debido a que las computadoras y máquinas de medición no captan datos todo el tiempo, su funcionamiento está regido por algún reloj o tiempo establecido y la información que almacenan se encuentra en arreglos finitos de memoria. Por lo que se discretizan las señales, lo cual consiste en una secuencia de números ordenados [24], tal y como se observa en la Figura 7.

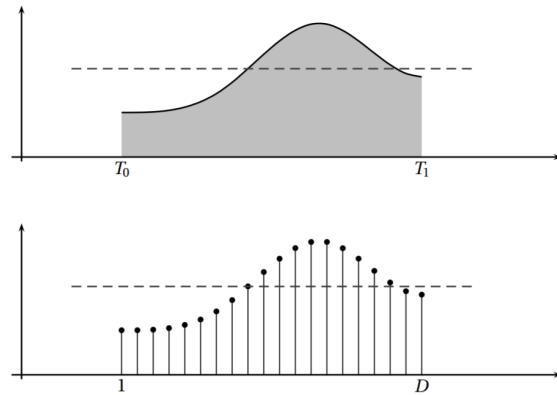


Figura 7: Señal continua y discreta [24]

Para llevar a cabo el procesamiento digital de señales es necesario filtrar la señal analógica, convertir la señal analógica a digital con un ADC, procesarla, convertir la señal digital en analógica con un DAC y filtrar de nuevo la señal analógica [25]. Dicho proceso se puede apreciar en el diagrama de la Figura 8.

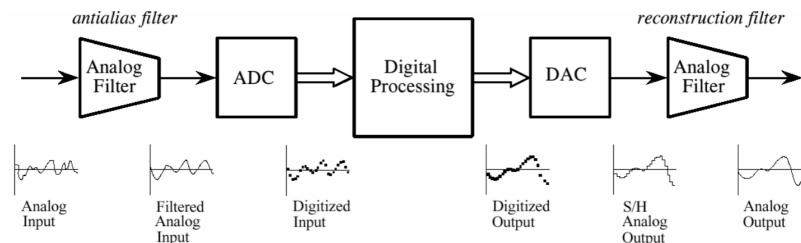


Figura 8: Diagrama de bloques Procesamiento Digital de Señales [25]

6.9. Filtrado de Señales

El proceso de filtrado de señales consiste en recibir una señal de entrada, manipularla y devolver una señal modificada como salida. Existen diversos tipos de filtros, unos de los más utilizados son los filtros selectores de frecuencia. Este tipo de filtros se caracteriza por dejar pasar las señales que estén dentro de la banda de paso y atenuar las señales que estén fuera

de la banda de paso, la banda atenuada se conoce como banda de rechazo. La ubicación de esta banda de paso es lo que diferencia las cuatro principales categorías de filtros selectores de frecuencia, las cuales se detallan a continuación. La diferencia entre la banda de paso y banda de rechazo se da en la frecuencia de corte, a partir de dicha frecuencia la señal se atenuara 70.7% lo cual es equivalente a -3dB de amplitud [26].

Los filtros pasa altas y pasa bajas se caracterizan por tener una banda de rechazo y una banda de paso, las cuales están separadas por la frecuencia de corte. Los nombres describen la funcionalidad del filtro, dejando pasar frecuencias mayores y menores a la frecuencia de corte para el pasa altas y pasa bajas, respectivamente. Las dos categorías restantes son el filtro pasabanda y rechaza banda, los cuales tienen dos frecuencias de corte. El filtro pasa banda deja pasar la señal a la salida solamente cuando la frecuencia de entrada está dentro del intervalo comprendido entre las dos frecuencias de corte. A diferencia del filtro rechaza banda que no deja pasar las señales con frecuencias que se encuentran dentro del intervalo comprendido por las frecuencias de corte y solo deja pasar las que están fuera de este intervalo [26]. Las cuatro categorías antes mencionadas se muestran gráficamente en la Figura 9.

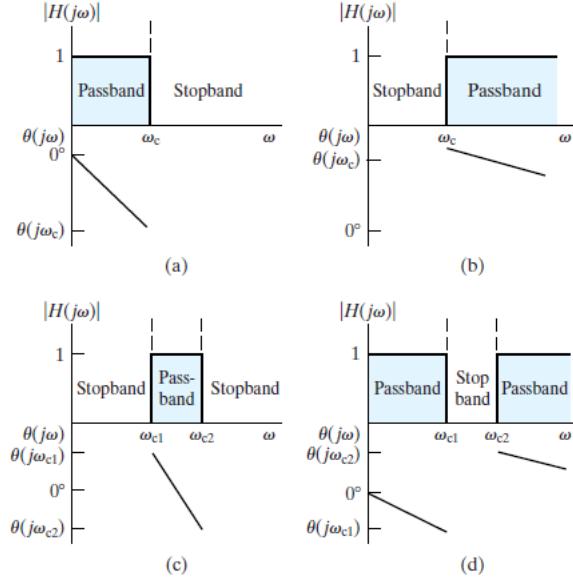


Figura 9: Respuesta ideal cuatro tipo de filtros selectores de frecuencia. (a) Filtro pasa bajas (b) Filtro pasa altas (c) Filtro pasa banda (d) Filtro rechaza banda [26].

Sin embargo, los filtros mostrados en la Figura 9 muestran filtros ideales. Los filtros reales a diferencia de los ideales tienen una banda de transición la cual se empieza a dar a partir de la frecuencia de corte y la velocidad de esta rampa de transición junto a otros parámetros varía según el tipo de filtro que se aplique. El tipo de filtro debe elegirse según la aplicación y la importancia de la respuesta en el dominio del tiempo o en el dominio de la frecuencia [26]. A continuación, se detallan cuatro tipos filtros.

El filtro de Butterworth es de los más utilizados debido a su respuesta maximamente plana en la banda de transición, es el que más se asemeja a la respuesta ideal. Luego tenemos el filtro de Chebyshev, el cual consigue caída más abrupta a frecuencias bajas y un rizado

de la respuesta en frecuencia en alguna de las bandas. El filtro de Bessel a diferencia de los anteriores no se ha degradado progresivamente, sin embargo, no pretende optimizar la anchura de la zona de transición sino asegurar fase lineal en toda la banda. El filtro elíptico, estrecha la zona de transmisión permitiendo un rizado constante en ambas bandas [27]. La respuesta en amplitud para los filtros mencionados se muestra en la Figura 10 comparada con la respuesta del filtro ideal en líneas punteadas.

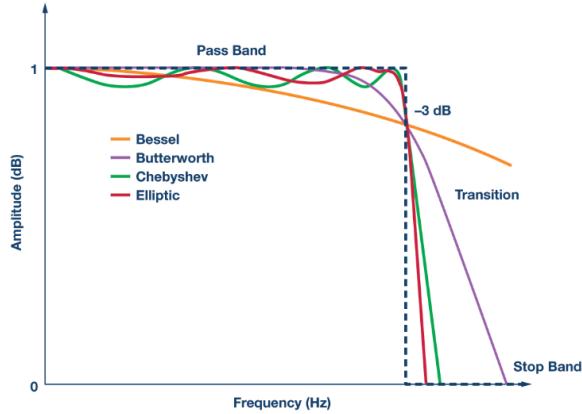


Figura 10: Respuesta en amplitud para diferentes tipos de filtros [28]

Como se mencionó anteriormente, para procesar las señales digitalmente estas se suelen discretizar. Por lo que el proceso de filtrado suele hacerse con filtros discretos. Se considera a un filtro discreto como un procedimiento que toma de la entrada las muestras numéricas, las almacena y manipula matemáticamente y las devuelve en la salida. Los filtros anteriormente mencionados en el dominio de la frecuencia pueden implementarse como filtros discretos. Así mismo también se puede implementar el filtrado discreto en el dominio del tiempo [24], existen 2 tipos de filtros los cuales se detallan a continuación:

Los filtros de respuesta finita al impulso, conocidos como FIR por sus siglas en inglés. Se implementan mediante una suma de convolución. La calidad del filtrado depende del número de muestras que se tomen. Estos filtros son conocidos como filtros de memoria finita. Una de las ventajas de este tipo de filtrado es que no depende de los valores de entrada, solamente del número de muestras. Sin embargo, uno de los problemas de este es que introduce un retraso, debido a que mientras más muestras N tenga el filtro, mejor se comportara, pero esto será costoso computacionalmente y por ende introducirá mayor cantidad de retraso equivalente a $N/2$ entre la entrada y la salida [24].

Los filtros de respuesta infinita al impulso, conocidos como IIR por sus siglas en inglés. Estos a diferencia de los FIR, utilizan ecuaciones de diferencias con coeficientes constantes en lugar de la suma de convolución. El problema de este tipo de filtro es que cada valor nuevo depende del previo por lo que es necesario definir condiciones iniciales de operación. Estos filtros también inducen un retraso entre la entrada y salida. Los filtros IIR son más complejos de diseñar y ofrecen menos flexibilidad, es por esto por lo que en la práctica se suelen preferir los filtros FIR. Cuando el coste computacional es un requerimiento de diseño si se suele preferir los IIR [24].

CAPÍTULO 7

Lectura de señales

7.1. Planteamiento del Proyecto

El primer paso para iniciar este proyecto de investigación consistió en seleccionar la línea de investigación de Biomédica de la Universidad del Valle de Guatemala, dirigida por el Dr. Luis Alberto Rivera. Una vez aceptado en esta línea de investigación se procedió a leer trabajos de graduación de proyectos anteriores relacionados con el tema para poder elegir el proyecto cuyos objetivos coincidieran con mis intereses.

En este caso se eligió el tema de interfaz cerebro computador debido al auge que ha tenido el análisis de este tipo de señales e implementarles inteligencia artificial para diferentes tipos de aplicaciones. Este tema se encuentra en constante desarrollo y prueba de ello es que desde 2016 han surgido diferentes emprendimientos que buscan desarrollar esta tecnología como *Neuralink*, *Kernel*, *Paradromics*, entre otros. Otro motivo por el cual se eligió este tema de investigación es por la experiencia que tiene el Dr. Rivera, líder de la línea de investigación cuyo trabajo doctoral consistió en la clasificación de señales bioeléctricas aplicando inteligencia artificial [29].

Luego de haber definido el tema específico de la investigación, se procedió a buscar literatura sobre el tema para definir como se aplicaría el concepto utilizando los recursos de la Universidad del Valle de Guatemala. Debido a que el tema sigue siendo investigado, aún no hay una forma definida de lograr conexión cerebro computador mejor que las otras, por lo tanto, se procedió a analizar los recursos con los que se contaba y se sesgo la investigación para utilizar el equipo ya disponible en la UVG con inteligencia artificial, debido a la experiencia del Dr. Rivera como se explicó anteriormente.

Se decidió realizar una primera fase de interfaz cerebro computador utilizando señales EEG captadas por los sensores de recepción que tiene el departamento de Electrónica, Me-

catrónica y Biomédica de la UVG. Clasificarlas utilizando inteligencia artificial y demostrar su correcto funcionamiento controlando el R17 y los dispositivos animatrónicos que tiene el departamento.

Sin embargo, debido a la pandemia del COVID-19 ya no se tuvo acceso a ningún equipo de los antes mencionados. Por lo que se trabajaría todo a nivel simulación, siempre respetando el sesgo antes mencionado para que futuras fases de esta investigación puedan tomar como base este trabajo. Por lo que en lugar de familiarizarse con el equipo de recepción que posee la UVG, se procedió a buscar bases de datos cuyas grabaciones se pudieran tomar como grabaciones hechas con el equipo de la UVG realizando diferentes gestos o actividades.

7.2. Búsqueda Base de Datos

Por recomendación del asesor y en base al trabajo en [10] se procedió a buscar bases de datos en PhysioNet. Debido a que estas bases de datos son públicas y son utilizadas por la academia para investigación. Se inició buscando bases de datos de señales EEG que usaran el mismo equipo de recepción de señales que posee la UVG, sin embargo, no se pudo conseguir una base de datos pública cuyas grabaciones hayan sido hechas con dichos dispositivos.

7.2.1. Base de datos EEG Motor Movement/Imagery

Se procedió a buscar bases de datos cuyas grabaciones estuvieran debidamente identificadas y fueran diferentes individuos realizando gestos o actividades. Luego de una búsqueda exhaustiva se encontró la base datos presentada en [30]. La base de datos *EEG Motor Movement/Imagery* tiene más de 1500 grabaciones de 64 canales de señales EEG de 109 voluntarios, cada grabación dura 1 o 2 minutos con una frecuencia de muestreo de 160 muestras por segundo. Sin embargo, esta base de datos presenta ciertos inconvenientes los cuales se detallan en los capítulos posteriores y se tuvo que buscar otra base de datos para verificar el funcionamiento del clasificador automático.

La base de datos contiene archivos en formato .edf y .edf.event. EDF es el formato de datos europeo, comúnmente utilizado por su flexibilidad y simplicidad para intercambiar y almacenar señales biológicas o físicas multicanal. Para poder leer dichos datos fue necesario descargar las funciones `edfread()`, `edfreadUntilDone()` y `Eventread()` las cuales se encuentran en [31] y [32].

7.2.2. Base de datos Sleep-EDF Database

Debido a los inconvenientes que se tuvo con la base de datos anterior, se procedió a utilizar la base de datos utilizada en [10] entre otros trabajos de investigación, como prueba de concepto y demostrar que el problema no se encontraba en el clasificador sino en el tipo de datos que se tenían. La base de datos [33] es propiedad de PhysioNet al igual que la anterior y en ella se pueden encontrar 197 polisomnografías completas con sus respectivos hipnogramas anotados por especialistas en base a las guías de Rechtschaffen y Kales (R&K) de 1968. De

las 197 polisomnografías solamente se utilizaron las primeras 153, debido a que las restantes eran bajo efectos de Temazepam. Las grabaciones fueron de alrededor de 20 horas para cada individuo y contiene 4 señales, las cuales son: Dos canales EEG siendo Fpz-Cz y Pz-Oz, un canal electromiográfico (EMG) ubicado en el mentón y un canal electrooculográfico (EOG). La frecuencia de muestreo es de 100 muestras por segundo.

La base de datos contiene archivos en formato .edf. Sin embargo, se utilizaron los archivos previamente convertidos por [10] a formato .mat para probar los clasificadores en MATLAB.

7.3. Procesamiento de señales

Luego de poder visualizar los datos en MATLAB y almacenarlos como formato .mat se procedió a seleccionar los canales que se trabajarían. En el caso de la base de datos de movimientos se tenían 64 canales, por lo que se hicieron diversas pruebas con diferentes combinaciones de canales las cuales se detallan en los capítulos posteriores. En el caso de la base de datos de sueño se utilizaron 4 canales los cuales son: 2 EEG, 1 EMG y 1 EOG.

7.3.1. Adquisición de señales

Con los canales ya seleccionados se procedió a realizar la simulación de adquisición de señales en tiempo real, para simular correctamente la frecuencia de muestreo se utilizó la interrupción del Timer 1 del Arduino Mega, esto para asegurarnos que entraba un nuevo dato tal y como fue grabado en la base de datos. El baudaje seleccionado fue de 115,200 y a cada interrupción el Arduino envía una bandera por comunicación serial a MATLAB para leer un dato nuevo cada vez que se presente esta bandera. La interrupción se configuró diferente para cada base de datos debido a que el periodo de muestreo era distinto para cada una, siendo 6.25 milisegundos y 10 milisegundos para la base de datos de movimientos y de sueño, respectivamente.

7.3.2. Filtrado de señales

Para la parte de filtrado se decidió implementar un filtro rechaza banda, cuyas frecuencias de corte sean 59 y 61 Hz, esto para evitar el posible ruido generado por la red de distribución eléctrica que en Guatemala es de 60 Hz.

El filtro implementado se decidió que sería Butterworth, grado 2 e IIR por su respuesta plana y bajo coste computacional, según lo investigado en el capítulo 6. Se implementó en MATLAB utilizando la librería de procesamiento de señales. Específicamente la parte de diseño de filtros, tal y como se muestra en la Figura 11. Como se puede observar, la aplicación de MATLAB es bastante intuitiva por lo que se simplifica el diseño de filtros.

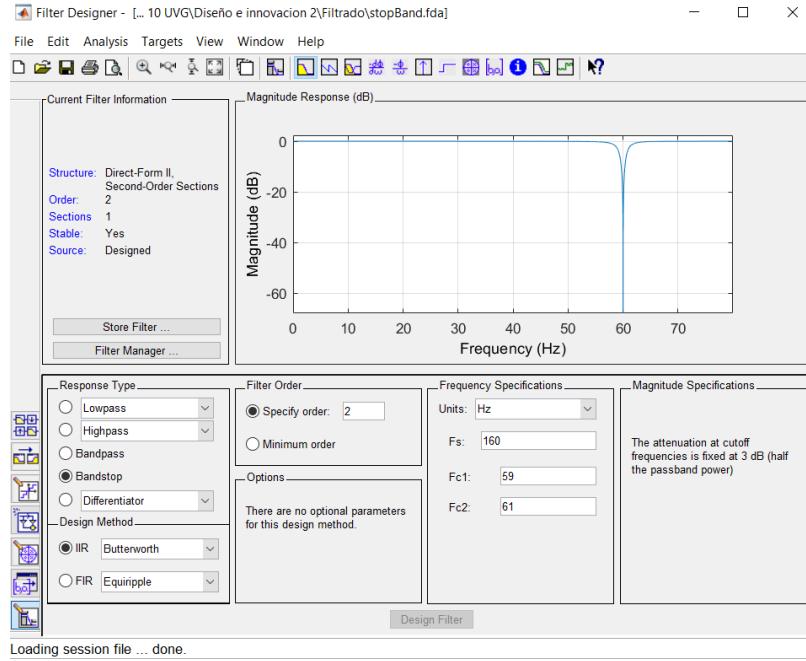


Figura 11: Diseño de filtro rechaza banda en el *Filter Designer* de MATLAB

Una vez diseñado el filtro se procedió a verificar su correcto funcionamiento creando una señal artificial de la forma:

$$A = \sin(2\pi * 60t) + \sin(2\pi * 5t) \quad (7)$$

Con el fin de verificar el correcto funcionamiento del filtro rechaza banda, se esperaba ver a la salida la banda de 60 Hz atenuada, por lo que se realizaron las gráficas de amplitud respecto al tiempo y espectro de frecuencia para observar que efectivamente se removiera la banda indicada. Al observar en la Figura 12 se puede observar que efectivamente el filtro remueve la frecuencia de 60 Hz.

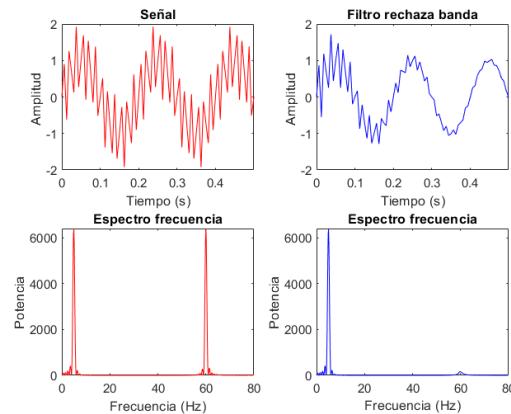


Figura 12: Gráfica amplitud respecto al tiempo y espectro de frecuencia antes y después del filtrado.

CAPÍTULO 8

Clasificadores

8.1. Características en el Dominio del Tiempo

Después de haber extraído la información de los 64 canales de la base de datos, se procedió a seleccionar solamente los canales de interés para el trabajo. Los cuales son C3, C1, Cz, C2 y C4 según la Figura 1. Luego de extraer los datos de estos cinco canales, se procedió a extraer las características. Al inicio se trajeron solamente el número de cruce por ceros (ZC) y el valor absoluto promedio (MAV). Sin embargo, como se puede observar en las Figuras 13 y 14 al aplicar SVM y RNA respectivamente, ninguno presenta resultados aceptables. Importante mencionar que para todos los resultados con SVM presentados a continuación se utilizó el 90 % de las muestras para entrenar y el 10 % para evaluar, de las 45 muestras obtenidas. Los resultados obtenidos con RNA se obtuvieron utilizando un tamaño de capa oculta de 10, la función de entrenamiento SCG debido al balance de consumo de memoria y calidad de resultados que presenta. Se utilizó un 80 % de las muestras para entrenar, 10 % para validar y 10 % para probar. En cuanto a las etiquetas utilizadas 1 y 2 se refieren a abrir y cerrar el puño izquierdo y derecho, respectivamente. Para SVM se tuvo que cambiar el 2 por el 0 para poder obtener de manera automática la matriz de confusión.

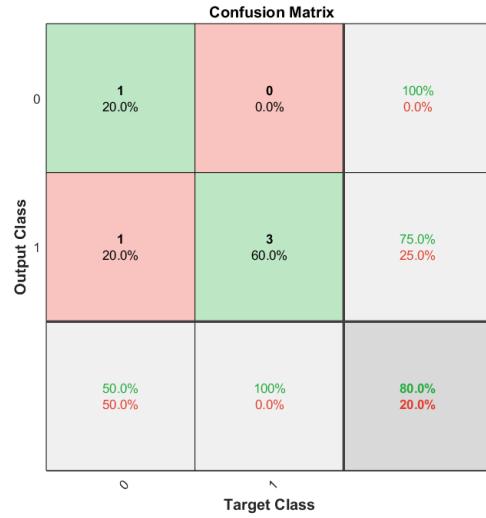


Figura 13: Matriz de confusión para SVM con kernel de función de base radial con características ZC y MAV.



Figura 14: Matriz de confusión para RNA con características ZC y MAV.

Debido a que los resultados de los clasificadores no eran los esperados, se procedió a calcular 2 características más las cuales son varianza y curtosis. Con estas dos características se obtuvieron mejores resultados para ambos clasificadores, los cuales se pueden observar en las Figuras 15 y 16 para SVM y RNA, respectivamente.



Figura 15: Matriz de confusión para SVM con kernel de polinomio grado tres con características varianza y curtosis.

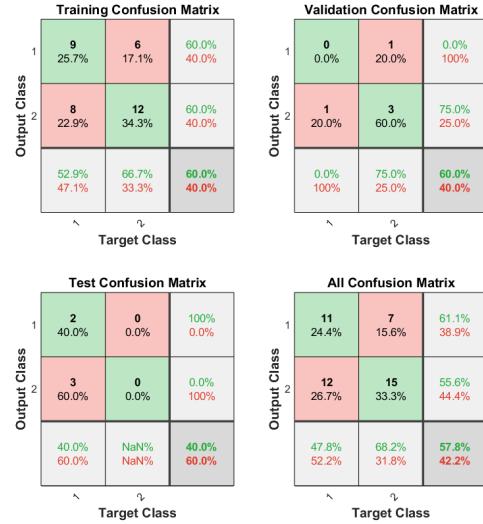


Figura 16: Matriz de confusión para RNA con características varianza y curtosis.

A pesar que en la Figura 15 se observa un rendimiento del 100 % en la matriz de confusión, cabe mencionar que este resultado se obtuvo porque el clasificador sufrió una sobrecarga de datos, el clasificador tenía más características que muestras. Esto se reflejaba en los experimentos como que el clasificador memorizara los datos de entrenamiento y al ser evaluado con datos nuevos, este ya no presentaba el mismo rendimiento. En busca de mejorar el rendimiento de los clasificadores, se procedió a utilizar las cuatro características a la vez. Sin embargo, los resultados obtenidos son muy similares a los obtenidos solamente con varianza y curtosis. Los resultados se pueden apreciar en las Figuras 17 y 18.



Figura 17: Matriz de confusión para SVM con kernel de polinomio grado tres con las cuatro características (ZC, MAV, k y V).

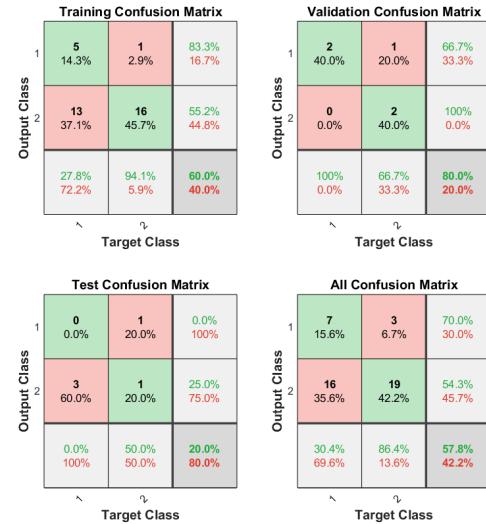


Figura 18: Matriz de confusión para RNA con las cuatro características (ZC, MAV, k y V).

Como se puede observar los resultados aun no son los deseados, a pesar de que para el caso de SVM se eligió el núcleo que mejor resultado presentó siendo RBF para ZC, MAV y polinomio grado 3 para varianza, curtosis y las cuatro características juntas. Los resultados fueron mejorando, sin embargo, aún no son los esperados. Aún que la matriz de confusión de la Figura 17 parezca tener un rendimiento perfecto, cabe mencionar que este rendimiento se debe a la sobrecarga de datos antes mencionada.

8.2. Características en el Dominio de tiempo frecuencia

Por instrucciones del asesor se implementó el clasificador extrayendo características en el dominio del tiempo-frecuencia, aplicando ondículas a la misma base de datos en lugar de características en el dominio del tiempo como se había estado trabajando anteriormente.

8.2.1. Clasificador 3 clases

Se inicio trabajando con los canales C3, C1, C6, Cp5, T9, T10, Poz y Po4. Luego se separaron los canales por sus bandas de frecuencia con el uso de ondículas para observar su dispersión en el tiempo se procedió a utilizar la aplicación de *Classification Learner* de MATLAB para observar que tipo de clasificador presentaba mejor rendimiento. Se realizaron pruebas para los primeros 2 canales, los primeros 4 canales y los 8 canales mencionados anteriormente obteniendo el mejor rendimiento para cada prueba con los métodos de clasificación de K-vecinos cercanos, SVM kernel cúbico y discriminante lineal, tal y como se observa en las Figuras 19, 20, 21, respectivamente.

Como se puede observar en las Figuras 19, 20, 21 los resultados del clasificador no son los esperados, independientemente del clasificador porque al utilizar redes neuronales tampoco se logró el rendimiento deseado tal y como se muestra en las Figuras 22 y 23.

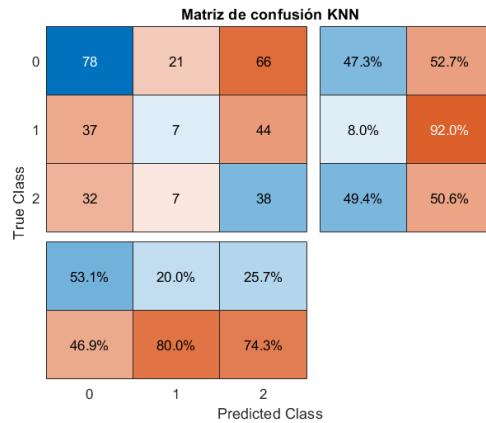


Figura 19: Matriz de confusión para KNN con ondículas para canales C3 y C1.

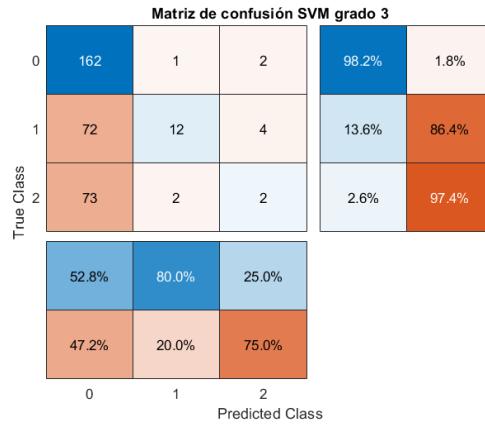


Figura 20: Matriz de confusión para SVM grado 3 con ondículas 4 canales (C3, C1, C6 y Cp5).



Figura 21: Matriz de confusión para discriminante lineal con ondículas 8 canales (C3, C1, C6, Cp5, T9, T10, Poz y Po4).

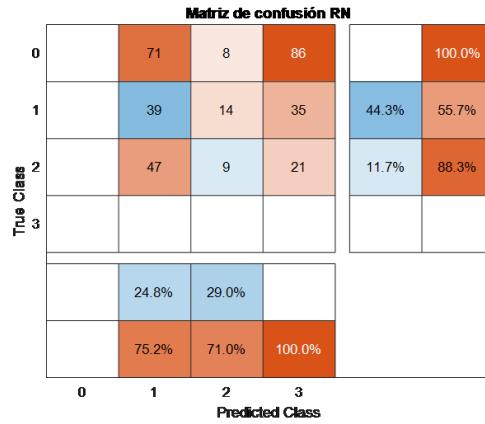


Figura 22: Matriz de confusión para RNA con ondículas para canales C3 y C1.

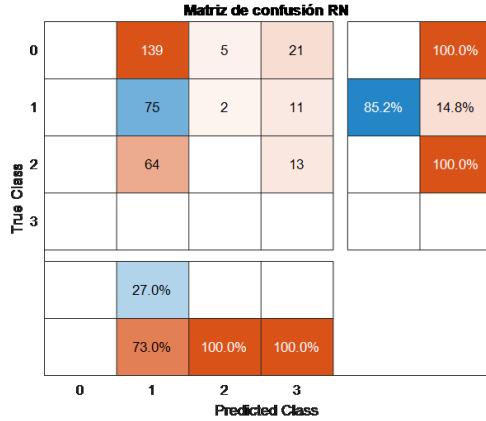


Figura 23: Matriz de confusión para RNA con ondículas 8 canales (C3, C1, C6, Cp5, T9, T10, Poz y Po4).

Los motivos por los cuales pudo no funcionar fue que con 2 canales existían muy pocos datos para clasificar correctamente. Al utilizar 4 y 8 canales se tuvo el problema que, al concatenar las características de todos los canales, el número de características era mayor al número de muestras, un problema conocido en Aprendizaje Automático como sobrecarga de datos. Por lo que se debía de tener cuidado que no existiese sobrecarga de datos para poder esperar buenos resultados.

Con estos resultados se observó que el clasificador SVM grado 3 presentaba siempre los mejores resultados o por lo menos estaba entre los primeros 3 al utilizar la aplicación de MATLAB por lo que se decidió utilizar solamente ese clasificador para las siguientes pruebas. Acompañado de redes neuronales como se había hecho desde un principio.

8.2.2. Clasificador 2 clases

Debido a los malos resultados obtenidos anteriormente se procedió a implementar un clasificador binario que, en lugar de decidir entre 3 clases como las pruebas anteriores, solo tuviera que decidir en 2 clases con 2 canales para no incurrir en la sobrecarga de datos de nuevo esperando obtener mejores resultados.

Se probaron las 3 diferentes combinaciones de clases y los mejores resultados se obtuvieron al clasificar las clases 0 y 1, cuya matriz de confusión se muestra en la figura 24. Como se puede observar los resultados aún no son los esperados, el clasificador en todas las pruebas muestra estar sesgado hacia la clase 0 esto se puede deber a que el número de muestras es el doble en comparación a las otras clases.

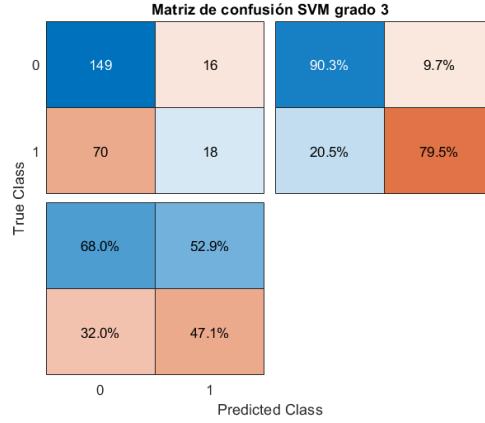


Figura 24: Matriz de confusión para SVM grado 3 con ondículas para clases 0 y 1.

8.2.3. Diferentes canales 3 clases

Se finalizaron las pruebas y no se obtuvo el resultado esperado por parte de los clasificadores, por lo que se optó por utilizar diferentes canales, en esta ocasión el O1 y el P3. Se obtuvieron los resultados que se muestran en las figuras 25 y 26, para SVM y redes neuronales, respectivamente.

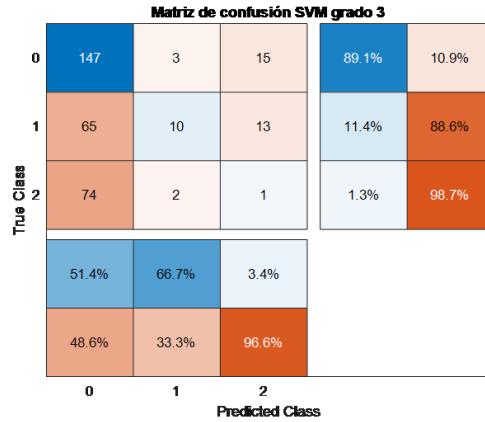


Figura 25: Matriz de confusión para SVM grado 3 con ondículas O1 y P3.

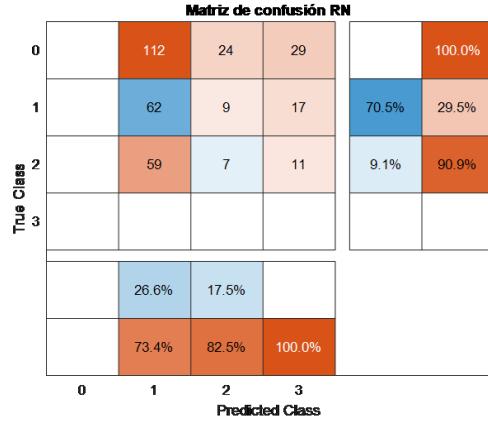


Figura 26: Matriz de confusión para RNA con ondículas para canales O1 y P3.

Luego de extraer características por medio de ondículas y aplicar los clasificadores de SVM kernel cúbico y Redes Neuronales, aun no se obtienen los resultados esperados. Por lo que se opta, por elegir diferentes canales, siendo estos el Pz y el Fpz. Los resultados se pueden observar en las figuras 27 y 28, para SVM y redes neuronales, respectivamente.

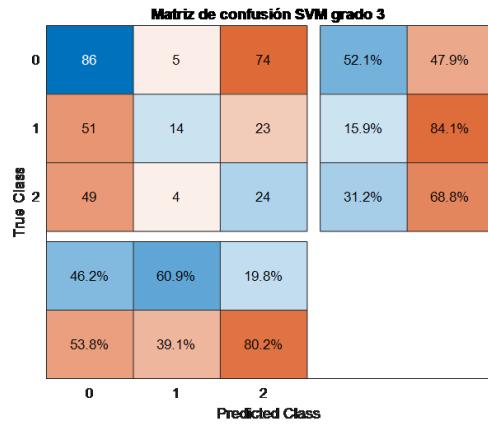


Figura 27: Matriz de confusión para SVM grado 3 con ondículas para canales Pz y Fpz.

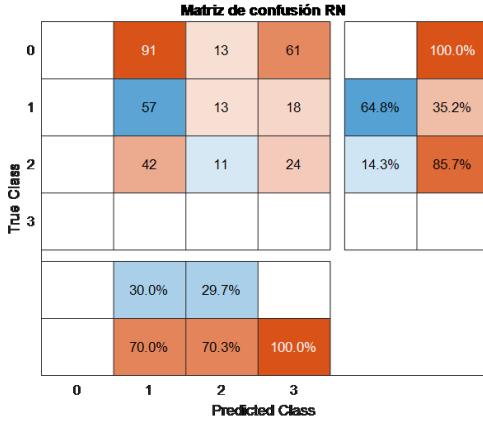


Figura 28: Matriz de confusión para RNA con ondículas para canales Pz y Fpz.

Al finalizar estas pruebas se puede ver que los clasificadores no presentan buen rendimiento, esto se puede deber a la poca cantidad de muestras que se tiene en la base de datos. A pesar que la base de datos tiene más de 1500 grabaciones de 109 voluntarios, los datos recopilados en una sesión no fueron suficientes para entrenar y poner a prueba el clasificador. Al querer utilizar datos de diferentes sesiones de grabación del mismo individuo, las características calculadas eran completamente diferentes, lo cual se puede deber al estado alerta y de animo del voluntario, así como la posición de los sensores. Estos problemas se podrían evitar en una próxima fase teniendo equipo propio de medición para entrenar y validar el clasificador en la misma sesión. Debido al inconveniente antes mencionado se optó por buscar otra base de datos para validar el funcionamiento de los clasificadores.

8.3. Base de datos de sueño

Debido a los malos resultados obtenidos con la base de datos anterior, se procedió a buscar otra base de datos con la cual se pudieran validar los clasificadores. Se eligió la base de datos de sueño [33] de PhysioNet utilizada por [10], debido a que tiene gran cantidad de señales EEG y están correctamente identificadas según sus clases, por lo que el tamaño de las señales no sería una limitante para el rendimiento de los clasificadores. Este cambio se hizo para demostrar que el concepto de clasificadores esta aplicado de la manera correcta y el problema presentado era por el tipo de datos de la base de datos anterior.

8.3.1. Varianza

Se inició analizando la base de datos en el dominio del tiempo. Las características que se extrajeron fueron 5, las cuales son: valor absoluto promedio, número de cruces por cero, varianza, potencia media y curtosis. Para las 4 señales proporcionadas cuyos nombres son fpz, pz, EOG y EMG. Se extrajeron épocas equivalentes a 30 segundos, tomando en cuenta que las grabaciones fueron hechas a con una frecuencia de 100 Hz.

Luego de extraer las características se procedió a implementar los clasificadores, los

resultados que se obtuvieron se muestran en las figuras 29 y 30 , para SVM y redes neuronales, respectivamente.

Matriz de confusión SVM grado 3					
True Class	1	2	3	4	5
	35	12	1	2	
1	35	12	1	2	
2	6	21	18	2	3
3			41	9	
4			4	46	
5			1		49
	85.4%	63.6%	63.1%	78.0%	94.2%
	14.6%	36.4%	36.9%	22.0%	5.8%
Predicted Class					

Figura 29: Matriz de confusión para SVM grado 3 con 5 características (MAV, ZC, bp, k y v).

Matriz de confusión RN					
True Class	1	2	3	4	5
	29	1		1	
1	29	1		1	
2	7	3	1		6
3		6	62		7
4			9	57	
5		1			36
	80.6%	27.3%	86.1%	98.3%	73.5%
	19.4%	72.7%	13.9%	1.7%	26.5%
Predicted Class					

Figura 30: Matriz de confusión para RNA con 5 características (MAV, ZC, bp, k y v).

8.3.2. Distancia Máxima Mínima

A pesar de obtener buenos resultados, se buscó mejorar utilizando otro tipo de características en el dominio del tiempo. Se utilizó la mayoría de las características anteriores, la diferencia consistió en que se sustituyó la varianza por la distancia máxima-mínima de la señal y se obtuvieron los siguientes resultados que se muestran en las figuras 31 y 32 , para SVM y redes neuronales, respectivamente.

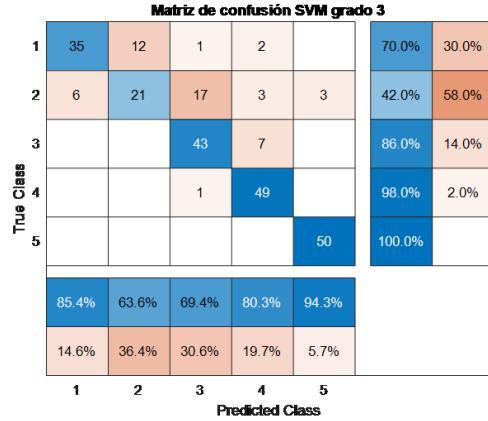


Figura 31: Matriz de confusión para SVM grado 3 con 5 características (MAV, ZC, bp, k y MMD).

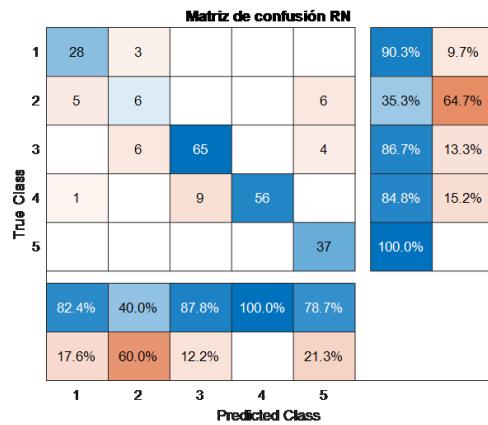


Figura 32: Matriz de confusión para RNA con (MAV, ZC, bp, k y MMD).

8.3.3. Ondículas

Por último, se probó clasificar las señales con característica en el dominio tiempo-frecuencia aplicando ondículas, en lugar de las 5 características extraídas anteriormente en el dominio del tiempo. Cuyos resultados se pueden observar en las Figuras 33 y 34 , para SVM y redes neuronales, respectivamente.

Matriz de confusión SVM grado 3						
True Class	0	1	2	3	4	
	336	2	1	32	1	90.3% 9.7%
0	336	2	1	32	1	90.3% 9.7%
1	4	736	90	32	38	81.8% 18.2%
2	2	95	693		2	87.5% 12.5%
3	39	32		112	21	54.9% 45.1%
4	1	6		16	421	94.8% 5.2%
	88.0%	84.5%	88.4%	58.3%	87.2%	
	12.0%	15.5%	11.6%	41.7%	12.8%	

Figura 33: Matriz de confusión para SVM grado 3 con ondículas.

Matriz de confusión RN						
True Class	1	2	3	4	5	
	327		2	43		87.9% 12.1%
1	327		2	43		87.9% 12.1%
2	4	682	124	32	58	75.8% 24.2%
3	3	81	706	2		89.1% 10.9%
4	59	32		89	24	43.6% 56.4%
5	5	7	3	15	414	93.2% 6.8%
	82.2%	85.0%	84.6%	49.2%	83.5%	
	17.8%	15.0%	15.4%	50.8%	16.5%	

Figura 34: Matriz de confusión para RNA con ondículas.

Luego de analizar los resultados de todas las pruebas, se procede a elegir Máquinas de Soporte Vectorial de la sección 8.3.2. Utilizando la distancia máxima-mínima de la señal, pues presentó los mejores resultados de clasificación a pesar de ser en el dominio del tiempo, lo cual se puede considerar como una ventaja por su fácil implementación en señales en tiempo real y bajo costo computacional.

Cabe mencionar que a pesar de obtener rendimiento mayor a 80 % el cual se considera un rendimiento aceptable para esta primera fase, siempre existe una de las cinco clases con bajo rendimiento esto se debe a la poca cantidad de datos que se tenían de esta clase para el entrenamiento del clasificador, por lo que al tener acceso al equipo de recepción de señales EEG de la universidad se espera no existan este tipo de errores que están fuera del alcance de esta primera fase.

CAPÍTULO 9

Visualización Gráfica

9.1. Robot

Esta sección se implementó para validar el correcto funcionamiento del clasificador automático antes mencionado. Tomando en cuenta que al inicio del proyecto se planificó comprobar el funcionamiento del mismo con el robot R17 del laboratorio de la universidad, pero por motivos fuera de nuestro alcance debido a la pandemia del COVID-19 no fue posible tener acceso a dicho robot, por lo que se procedió a implementar el modelo del mismo robot a nivel simulación para validar el funcionamiento del clasificador como si se tuviese acceso al modelo real de la universidad. Adicionalmente al modelo del R17, también se implementó la simulación de dos manipuladores seriales más para verificar el funcionamiento del clasificador sin importar el robot que debía controlar, estos manipuladores seriales son el UR5 y el Puma 560, dichos modelos y su implementación se detallan en las siguientes secciones.

Importante mencionar que la animación de los tres modelos se hizo con la librería de Robótica de Peter Corke [23]. Por lo que se propone desarrollar los diferentes modelos de robots en MATLAB y validar el correcto funcionamiento de la BCI a nivel simulación, con el fin de sustituir esta parte de simulación, por el robot de los laboratorios para próximas fases del proyecto.

Se decidió que cada robot tuviera dos rutinas de movimientos disponibles pensando en futuras fases del proyecto, para que la BCI ofrezca flexibilidad para adaptarse a diferentes tipos de controladores.

La primera rutina de movimientos consiste en generar trayectorias suaves para las juntas en el espacio tridimensional que lleven al efecto final a una posición final deseada, dicha rutina se explica en las siguientes secciones. Esta rutina de movimientos se pensó para aplicaciones de recoger y colocar comúnmente conocidas como *Pick & Place* con un manipulador

serial en ambientes hostiles que representen un riesgo para el ser humano y de esta manera evitar posibles accidentes. Esta rutina se piensa implementar en el R17 en futuras fases.

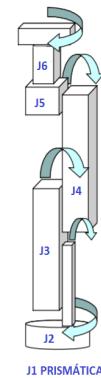
La segunda rutina de movimientos consiste en mover una junta a la vez 45° , esto con el fin de hacer movimientos redundantes en una junta a la vez. Según la clase que devuelva el clasificador, se moverá una junta específica. Esta rutina se pensó para los dispositivos animatrónicos de la universidad, con comandos no tan complejos se podría lograr abrir la boca o mover las cejas e incluso mover diferentes dedos, según el dispositivo animatrónico que se desee controlar.

9.1.1. R17

Como se mencionó anteriormente, se utilizó el manipulador R17 porque es el modelo que posee el Departamento de Ingeniería Electrónica, Mecatrónica y Biomédica de la UVG, el cual se muestra en la Figura 35a. El R17 es un manipulador serial de ST Robotics. Este tiene cinco grados de libertad más carrera, tal y como se observa en la Figura 35b tiene 1 junta prismática y 5 revolutas. Debido a que este modelo no lo incluye la toolbox de Peter Corke se tuvo que crear el mismo.



(a) Modelo Experimental



(b) Juntas

Figura 35: Manipulador Serial R17 de 5 GDL (más carrera) [34].

Como primer paso se establecieron los parámetros de Denavit-Hartenberg del R17, los cuales se muestran en el Cuadro 2, una vez ya definidos dichos parámetros se define el número de juntas, el tipo de junta **Prismatic()** o **Revolute()** y los parámetros para cada junta. Luego de tener establecidas las juntas se procede a definir el robot como un objeto **SerialLink()** y por último se establece la transformada de base para ajustar el modelo a las coordenadas globales utilizando **transl()** y **trotx()** tal y como se muestra en el código adjunto.

θ_j	d_j	a_j	α_j
0	q_1	0	$-\pi/2$
$q_2 + \pi/2$	-0.355	0	$-\pi/2$
$q_3 + \pi/2$	0	0.375	0
q_4	0	0.375	0
$q_5 - \pi/2$	0	0	$-\pi/2$
$q_6 + \pi/2$	0	0	0

Cuadro 2: Parámetros de Denavit-Hartenberg para manipulador R17 de 5 GDL (más carrera).

```

1 % Dimensiones del robot
2 a2 = 0; d2 = -0.355;
3 a3 = 0.375; d3 = 0;
4 a4 = 0.375; d4 = 0;
5 a5 = 0; d5 = 0;
6 a6 = 0; d6 = 0;
7
8 % Se definen 6 juntas, 1 prismatic y 5 revolutas
9 L1 = Prismatic('theta', 0, 'a', 0, 'alpha', -pi/2, 'offset', 0);
10 L1 qlim = [0,0.600];
11 L2 = Revolute('d', d2, 'a', a2, 'alpha', -pi/2, 'offset', pi/2);
12 L3 = Revolute('d', d3, 'a', a3, 'alpha', 0, 'offset', pi/2);
13 L4 = Revolute('d', d4, 'a', a4, 'alpha', 0, 'offset', 0);
14 L5 = Revolute('d', d5, 'a', a5, 'alpha', -pi/2, 'offset', -pi/2);
15 L6 = Revolute('d', d6, 'a', a6, 'alpha', 0, 'offset', pi/2);
16
17 % Se define el robot como un objeto Serial Link
18 r17 = SerialLink([L1, L2, L3, L4, L5, L6], 'name', 'r17');
19
20 % Se establece la transformada de base para ajustar
21 % el modelo a las coordenadas globales
22 r17.base = transl(0,0,-0.355)*trotx(-90);

```

Luego de definir el objeto serial se procedió a utilizar el comando `teach()` para definir cuáles serían las poses para cada comando. Las poses generadas según cada clase se presentan en el Cuadro 3. Cada pose representa un movimiento angular en el plano XY del robot, siendo 0° , 180° , 45° , 135° y 90° para las clases 1 a la 5, respectivamente. Dichas poses se logran con las configuraciones de las juntas presentadas en el Cuadro 3. En la Figura 36 se observan las poses para las configuraciones mostradas en el Cuadro 3.

Junta/Clase	0	1	2	3	4	5
J1 (m)	0	0	0	0.473	0.473	0.473
J2 (rad)	0	$\pi/2$	$-\pi/2$	$\pi/2$	$-\pi/2$	0
J2 (rad)	0	$\pi/4$	$\pi/4$	$\pi/4$	$\pi/4$	$\pi/4$
J4 (rad)	0	$\pi/4$	$\pi/4$	$\pi/4$	$\pi/4$	$\pi/4$
J5 (rad)	0	0	0	0	0	0
J6 (rad)	0	0	0	0	0	0

Cuadro 3: Configuración de juntas R17 para cada clase.

La primera configuración denominada Clase 0 es la pose que tiene el robot cuando no se detecta actividad. Para que el robot alcance la pose deseada sin exceder los límites de las juntas y realizando una trayectoria suave se utilizó la función `jtraj()` con un intervalo de tiempo de 1/4 de segundo para llegar a la trayectoria y el mismo tiempo para retornar a la configuración inicial. Por lo que efectuar cada trayectoria completa le tomaría medio segundo al manipulador real, variable que es modificable según la aplicación del manipulador.

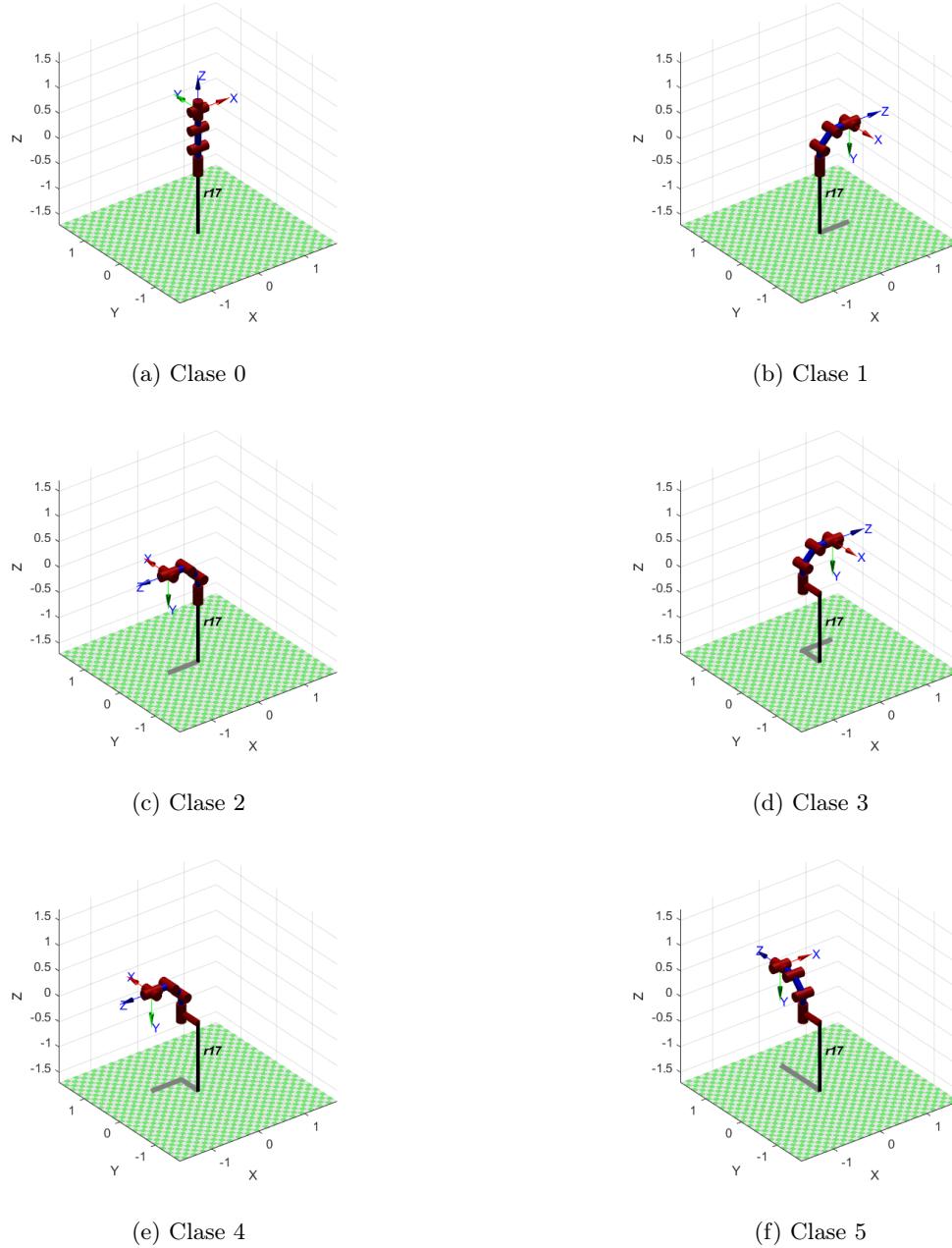


Figura 36: Configuración de juntas según clase para R17.

9.1.2. UR5

Para los siguientes dos modelos de manipuladores seriales, no hubo necesidad de crear el objeto serial como en el caso del R17, porque la librería de Peter Corke ya tenía definidos los diferentes modelos. El UR5 es un manipulador serial de Universal Robots con 6 juntas revolutas tal y como se observa en la Figura 37. Se utilizó este modelo por ser un manipulador serial popular en la industria debido a su versatilidad y la cantidad de accesorios disponibles para diversas aplicaciones. Para importar el modelo ya existente se utilizó el comando `mdl_ur5`. Luego, se procedió a establecer las configuraciones de las juntas deseadas para cada clase, las cuales se presentan en el Cuadro 4 y se pueden observar en la Figura 37.

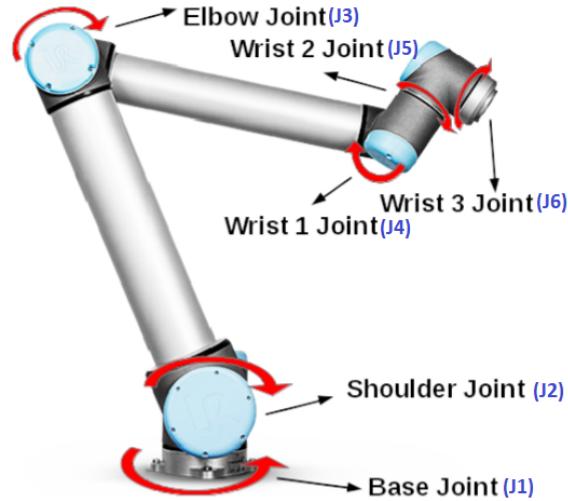


Figura 37: Juntas manipulador serial UR5 [35].

Junta/Clase	0	1	2	3	4	5
J1 (rad)	0	0	0	$\pi/4$	$-\pi/4$	$\pi/2$
J2 (rad)	$-\pi/2$	$-\pi$	0	$-\pi$	0	$-\pi$
J2 (rad)	0	0	0	0	0	0
J4 (rad)	0	0	0	0	0	0
J5 (rad)	0	0	0	0	0	0
J6 (rad)	0	0	0	0	0	0

Cuadro 4: Configuración de juntas UR5 para cada clase.

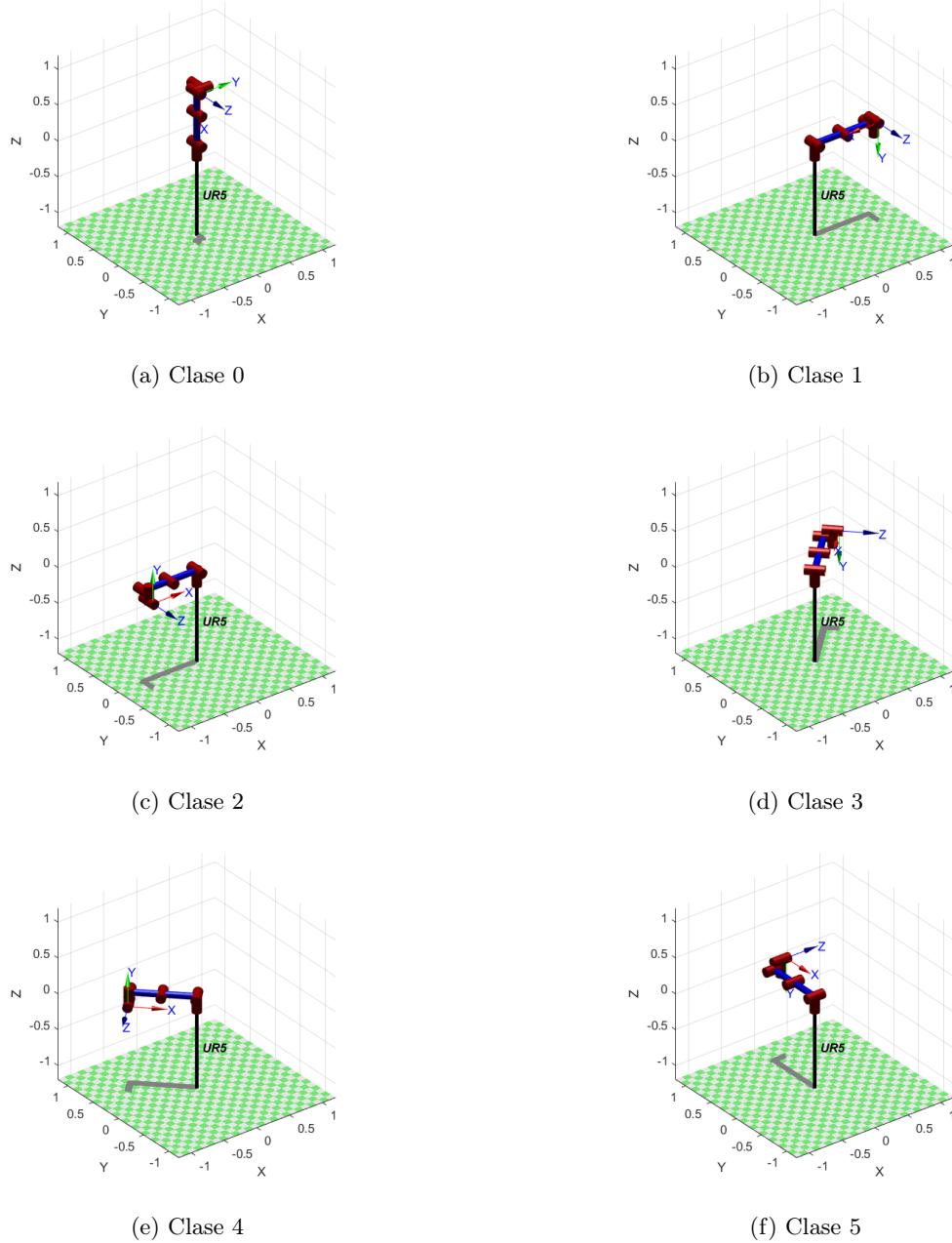


Figura 38: Configuración de juntas según clase para UR5.

9.1.3. Puma 560

El último manipulador serial seleccionado es el Puma 560 de Unimation. Se eligió este modelo por ser uno de los pioneros en la robótica y por el modelo renderizado que se puede utilizar en MATLAB para animaciones gracias a la librería ARTE para MATLAB de la Universidad Miguel Hernández [36]. Este manipulador serial cuenta con 6 juntas revolutas, por lo que luego de importar el modelo del robot se procedió a utilizar el comando `teach()` para definir las configuraciones deseadas para las juntas. En el Cuadro 5 se pueden apreciar

las configuraciones de las juntas para cada pose deseada según el comando entregado por el clasificador. En la Figura 37 se puede apreciar la pose del robot para cada una de las configuraciones antes mencionadas.

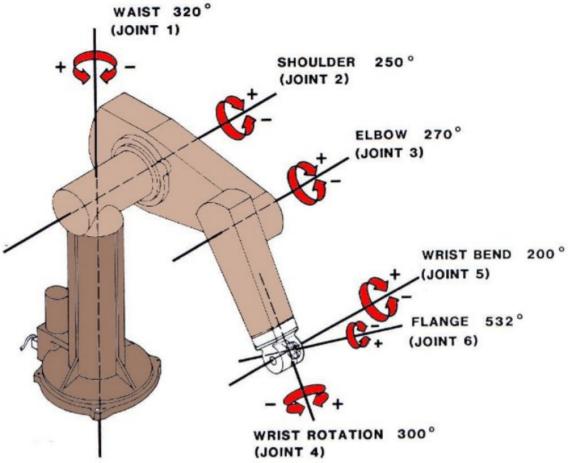


Figura 39: Juntas manipulador serial Puma 560 [37].

Junta/Clase	0	1	2	3	4	5
J1 (rad)	0	0	0	$\pi/4$	$3\pi/4$	$\pi/2$
J2 (rad)	$\pi/2$	0	π	0	0	0
J2 (rad)	$-\pi/2$	$-\pi/2$	$-\pi/2$	$-\pi/2$	$-\pi/2$	$-\pi/2$
J4 (rad)	0	0	0	0	0	0
J5 (rad)	0	0	0	0	0	0
J6 (rad)	0	0	0	0	0	0

Cuadro 5: Configuración de juntas Puma 560 para cada clase.

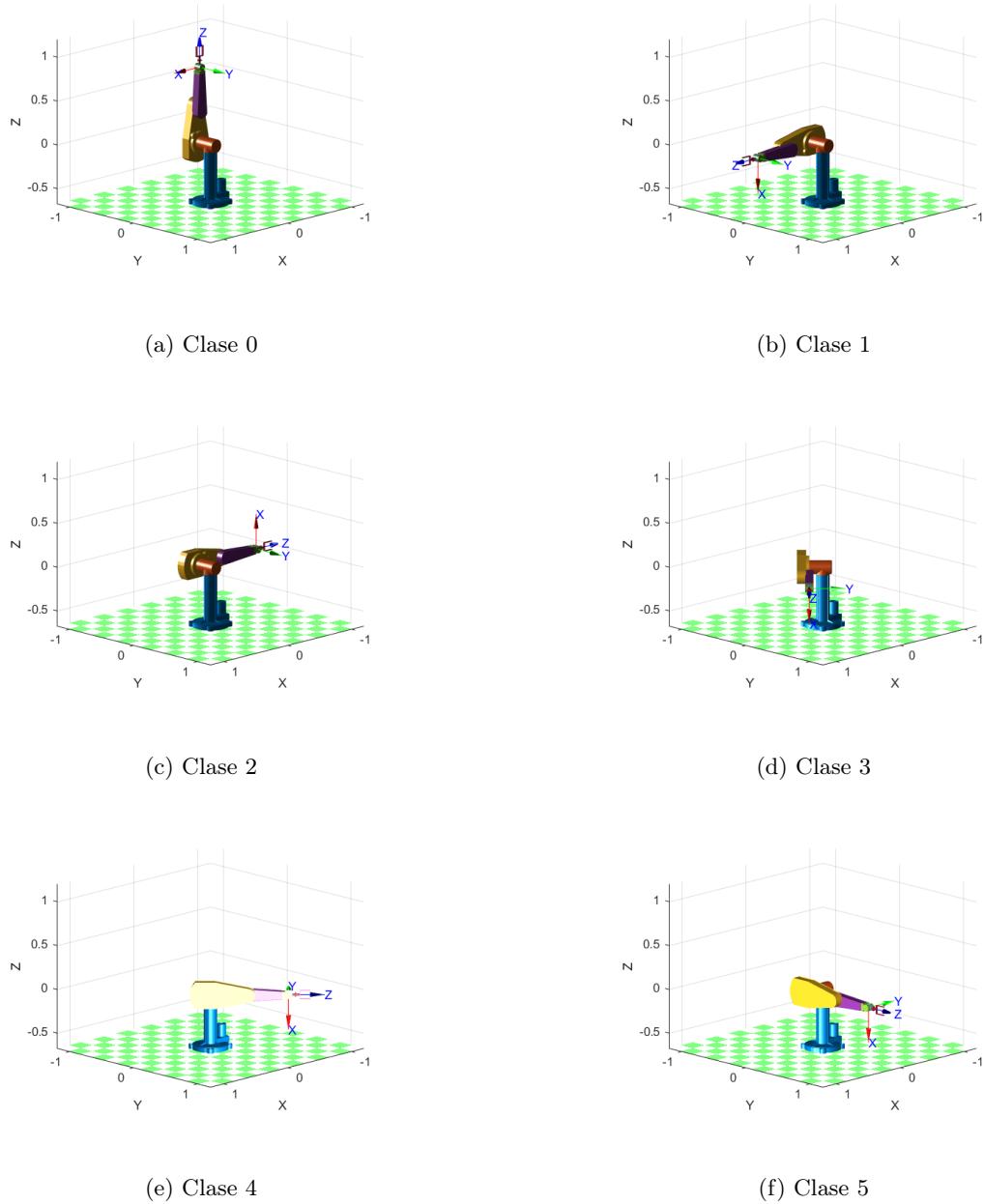


Figura 40: Configuración de juntas según clase para Puma 560.

9.2. Interfaz Gráfica

Para presentar los resultados obtenidos se diseñó una interfaz gráfica que facilitara la visualización de los resultados y permitiera la interacción del usuario con la BCI de manera amigable. El desarrollo de la interfaz gráfica se llevó a cabo con el *App Designer* de MATLAB. Se consideró importante que la interfaz a diseñar fuera intuitiva y fácil de usar para el usuario sin importar los conocimientos en robótica o programación que este tenga.

Para facilitar la interacción del usuario con la interfaz se ordenó la información a desplegar en dos pestañas las cuales son “Control Manual” y “Control Automático”. Adicional a estas dos pestañas, se tiene un panel al lado izquierdo que siempre está presente, donde se encuentran dos switches. El primer switch es para iniciar o detener la grabación de datos para el modo automático, dicho botón también sirve para seleccionar el tipo de control que manipulará al robot. El segundo switch es para seleccionar la rutina de movimientos que se desea que ejecute el robot, siendo junta por junta o trayectorias (*Pick & Place*).

9.2.1. Control Manual

Al inicializar la interfaz, se desplegará la pestaña de control manual, tal y como se muestra en la Figura 41. La idea de esta pestaña es permitir al usuario familiarizarse con los movimientos que es capaz de realizar el robot dependiendo el comando que seleccione. El botón de iniciar grabación se encuentra en off, lo cual indica que el modo de control automático no está activo y se puede controlar el robot de manera manual.

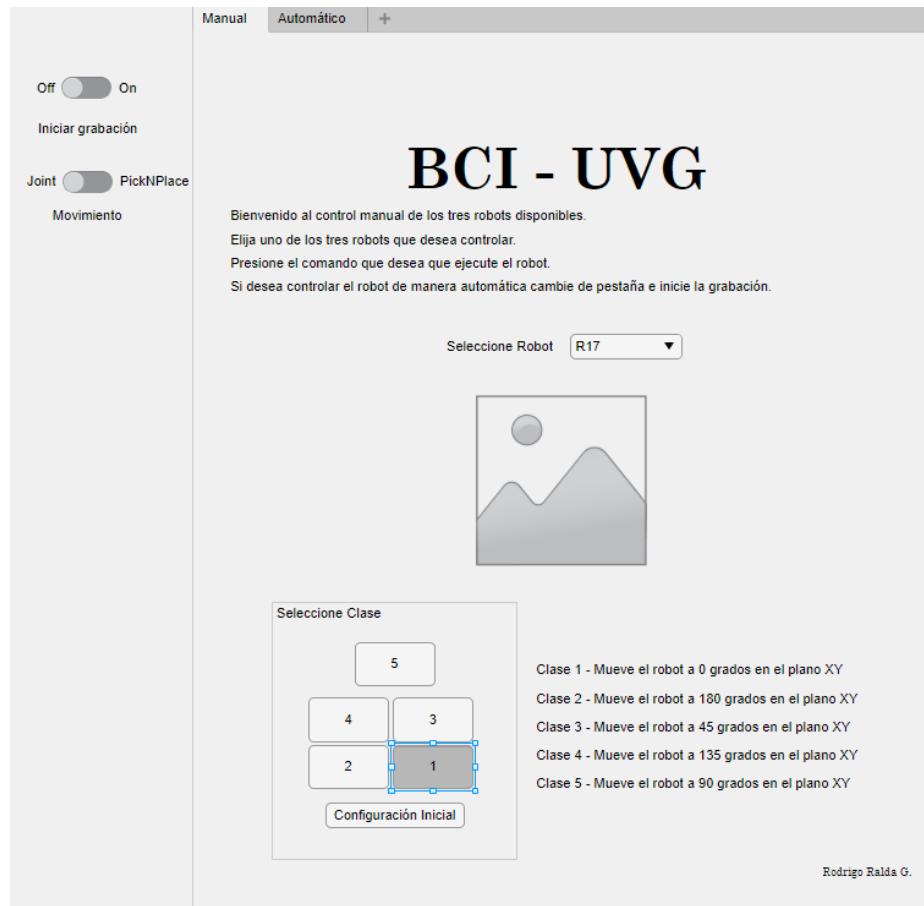


Figura 41: Pestaña modo manual.

En la parte superior de la pestaña aparece un cuadro de lista desplegable donde el usuario puede seleccionar el modelo del robot que desea controlar entre las tres opciones disponibles, las cuales son: R17, UR5 y Puma 560. Inmediatamente al seleccionar el robot, se desplegará

en la parte central de la pestaña la imagen del robot que se seleccionó las cuales se pueden observar en las Figuras 35, 37 y 39 para el R17, UR5 y Puma 560, respectivamente. Adicional a las imágenes también se despliega una figura de MATLAB cuyas opciones varían según la opción seleccionada y se presentaron en la sección anterior en las Figuras 36, 38 y 40 donde se visualizarán los movimientos que ejecute el robot acorde al comando seleccionado por el usuario en tiempo real.

Estos comandos aparecen en la parte inferior de la pestaña, al lado izquierdo se pueden observar los cinco botones para los diferentes tipos de comandos y al lado derecho una breve descripción de lo que ejecutará el robot al presionar el botón correspondiente.

Importante mencionar que utilizar el comando `SerialLink.plot()` de Peter Corke hace ineficiente el código, a manera que el funcionamiento de la interfaz se detiene mientras se está ejecutando el comando de graficar los manipuladores seriales. Para la rutina de movimiento individual de juntas este problema es imperceptible, sin embargo, para la rutina de movimiento por trayectorias se puede observar cómo se detiene la interfaz mientras se mueve el robot. Problema que no afectará en futuras fases del proyecto debido a que el modelo físico será ajeno al funcionamiento de la interfaz.

9.2.2. Control Automático

La segunda pestaña de la interfaz es para visualizar y activar todo lo relacionado al control automático de la interfaz. El modo automático permanecerá apagado hasta que se presione el botón de iniciar grabación en el panel izquierdo de la interfaz. Antes de presionar dicho botón se recomienda al usuario familiarizarse con los datos que desplegará la nueva pestaña que se muestra en la Figura 42.

En la parte superior izquierda se puede observar un deslizador el cual se recomienda configurar antes de iniciar la grabación, en este deslizador se puede seleccionar un valor entre 1 y 30 segundos. El significado de este tiempo es cuantos segundos se desea que dure la ventana para capturar datos y clasificar para luego enviar el comando al robot, por lo que el tiempo seleccionado se verá reflejado en el eje horizontal de cada uno de los cuatro gráficos de la parte central de la pestaña y también en el tiempo entre la ejecución de los comandos del robot seleccionado.

En la parte superior derecha se puede observar el resultado del clasificador, la etiqueta de "Clase actual" despliega el resultado del clasificador y este resultado se actualiza a cada intervalo de tiempo según lo seleccionado en el deslizador.

En el centro de la pestaña automático se pueden observar 4 gráficas, las cuales corresponden a las cuatro señales que se están recopilando en tiempo real para clasificarlas cada cierto tiempo definido por el deslizador antes mencionado. Las dos señales superiores corresponden a los canales EEG de Fpz y Pz, las dos señales inferiores corresponden a los canales de electrooculografía (EOG) y electromiografía (EMG), de izquierda a derecha respectivamente. Importante mencionar que después de varias pruebas se comprobó que es más eficiente computacionalmente graficar punto por punto, a manera de sobreescibir los puntos anteriores en lugar de graficar toda la ventana con el comando `plot()` de MATLAB. Por lo que al iniciar las gráficas se observara una línea horizontal en $y = 0$, para luego sobreescibir

estos valores.

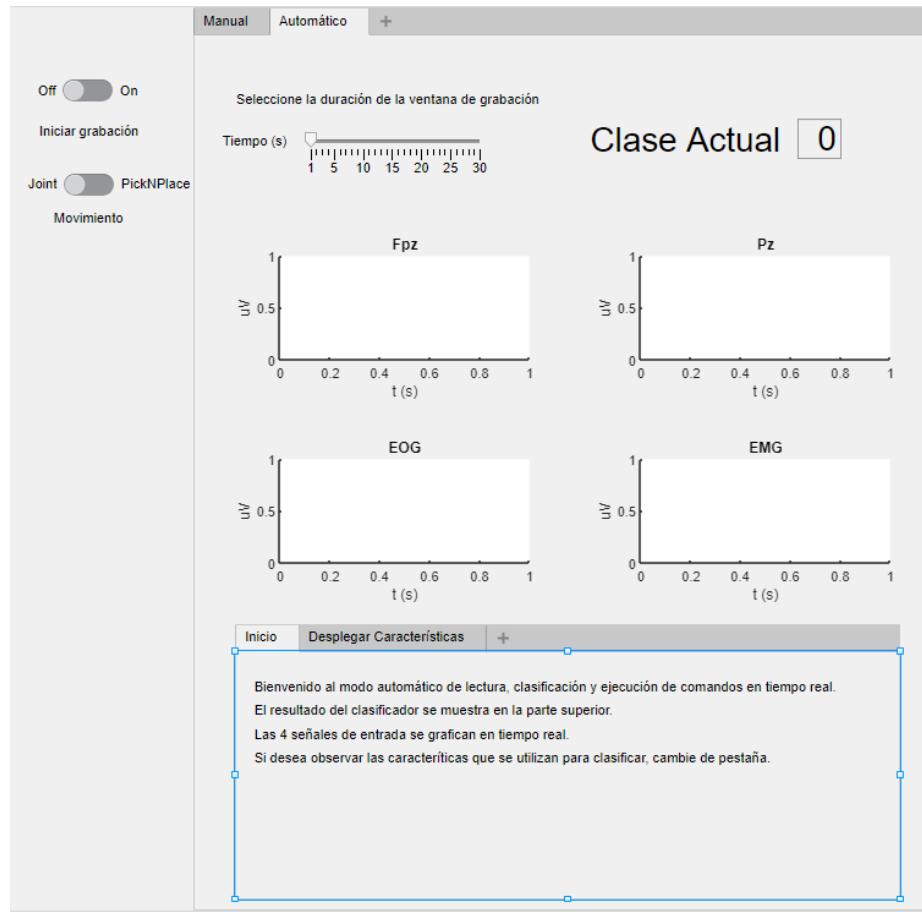


Figura 42: Pestaña modo automático.

En la parte inferior de esta pestaña se despliega una pestaña donde el usuario puede leer una breve descripción de lo que está viendo, también puede cambiar a la pestaña "Desplegar Características" donde observara las 16 características utilizadas para llevar a cabo la clasificación. Estas características se actualizan según el tiempo determinado en el deslizador.

CAPÍTULO 10

Validación de resultados

Este capítulo consiste en la integración de los diferentes módulos trabajados durante el proyecto. Para validar el correcto funcionamiento es necesario hacer varias pruebas.

Se empezó validando en la pestaña de control manual, la idea de esta pestaña es que el usuario pueda familiarizarse con los comandos que realizará el robot según el resultado del clasificador, además de controlar el robot de manera manual. Luego de seleccionar el modo manual, se procedió a seleccionar la rutina de movimientos de juntas individuales, la cual se detalló en capítulos anteriores. Se seleccionó el robot R17 y se verificó que efectivamente el control por juntas funcionaba pues al seleccionar el comando 2 se mueve la junta 3 del robot 45° , tal y como se muestra en la Figura 43. Además, que la selección del robot es correcta y despliega una imagen con fines ilustrativos para que el usuario se familiarice con el modelo real.

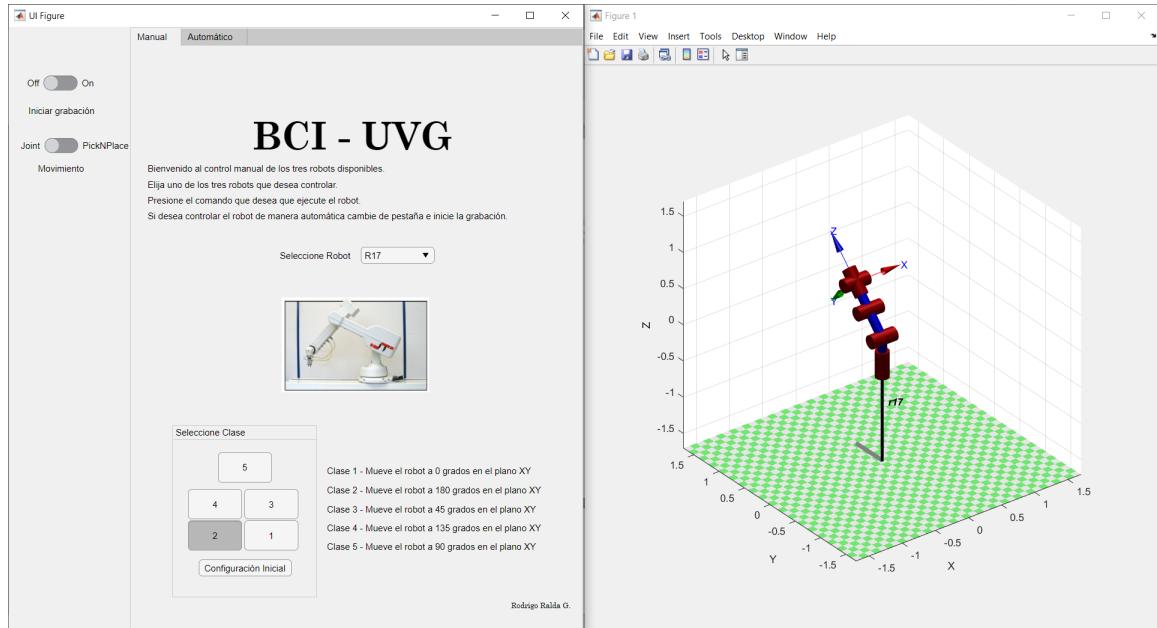


Figura 43: Control Manual por Juntas para R17, selección comando 2

Como segunda prueba se seleccionó el control manual con rutina de movimiento por trayectorias, en este caso se utilizó el robot UR5 y se seleccionó la clase 2 y se comprueba que el robot efectivamente se mueve hacia 180°en el plano XY con una trayectoria suave. El momento en que alcanza la posición deseada se muestra en la Figura 44.

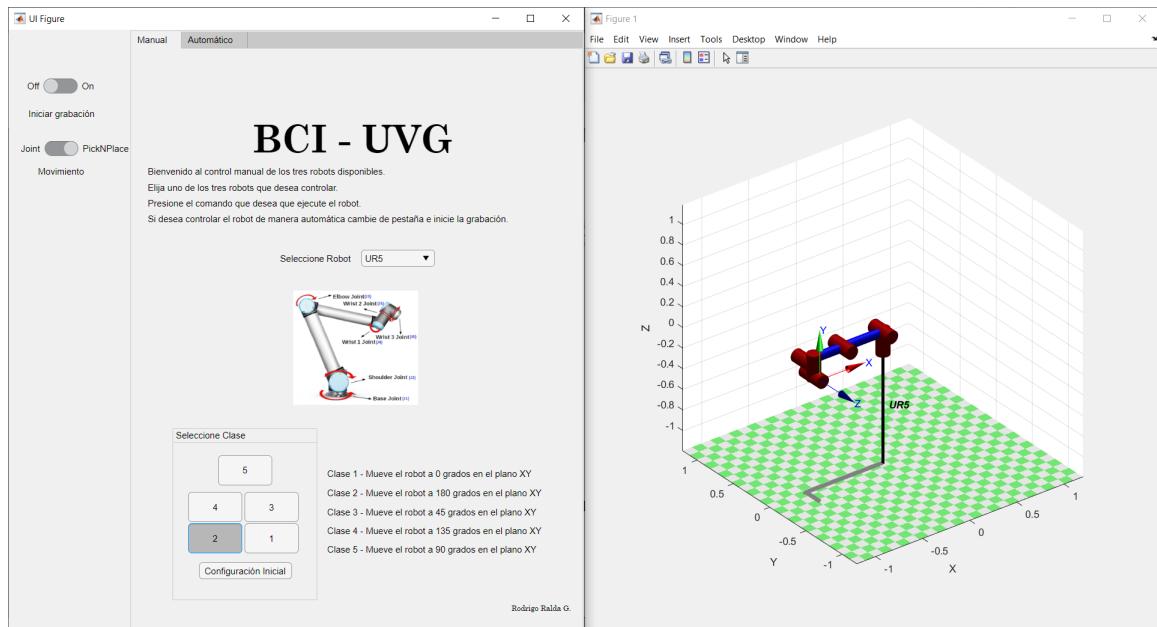


Figura 44: Control Manual por trayectorias para UR5, selección comando 3

Con las pruebas anteriores se pudo confirmar que la pestaña de control manual funciona de manera correcta, por lo que se procedió a probar la pestaña de control automático con

el robot Puma 560 que no se había probado.

Se selecciono el robot y luego se estableció un tiempo de captura de datos de 3 segundos, se verifico que al detectar ruido el clasificador fuera capaz de detectar la clase 0 y no realizar ningún movimiento, es decir mantenerlo en su configuración inicial. Tal y como se muestra en la Figura 45.

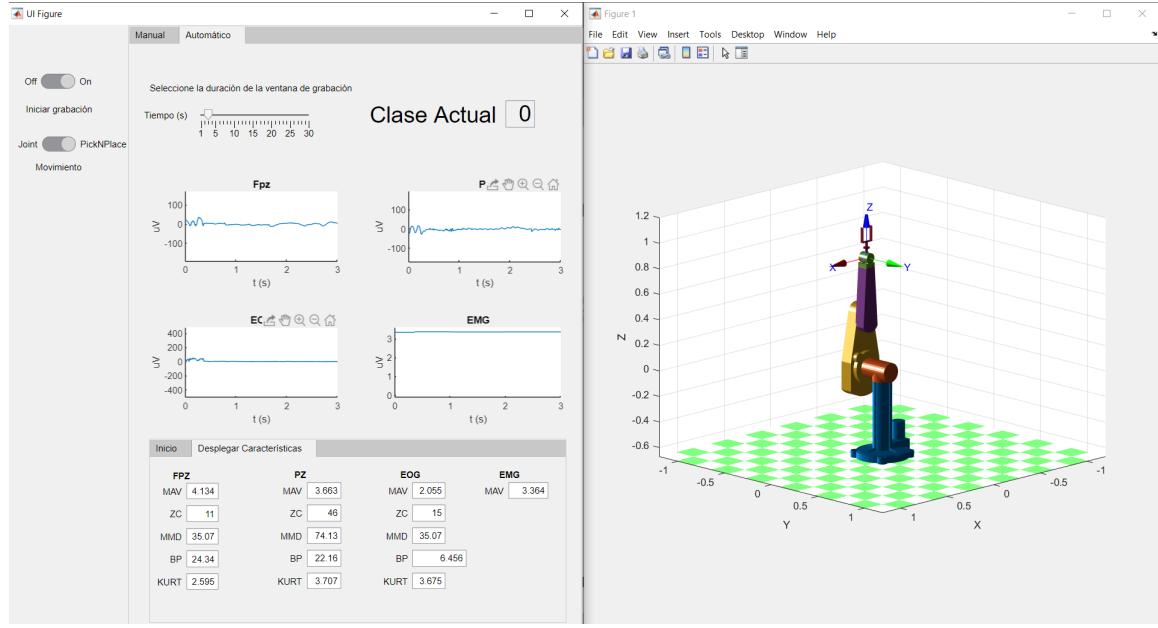


Figura 45: Control Automático de 3 segundos al detectar ruido.

La última prueba que se realizó fue cambiar la ventana de tiempo y se seleccionó de 5 segundos con el mismo robot. Se logra observar en la Figura 46 que la interfaz despliega correctamente las características calculadas en tiempo real en la parte inferior de la interfaz, el modelo del robot se mueve a 0 grados en el plano XY tal y como se esperaba y las señales son graficadas en tiempo real.

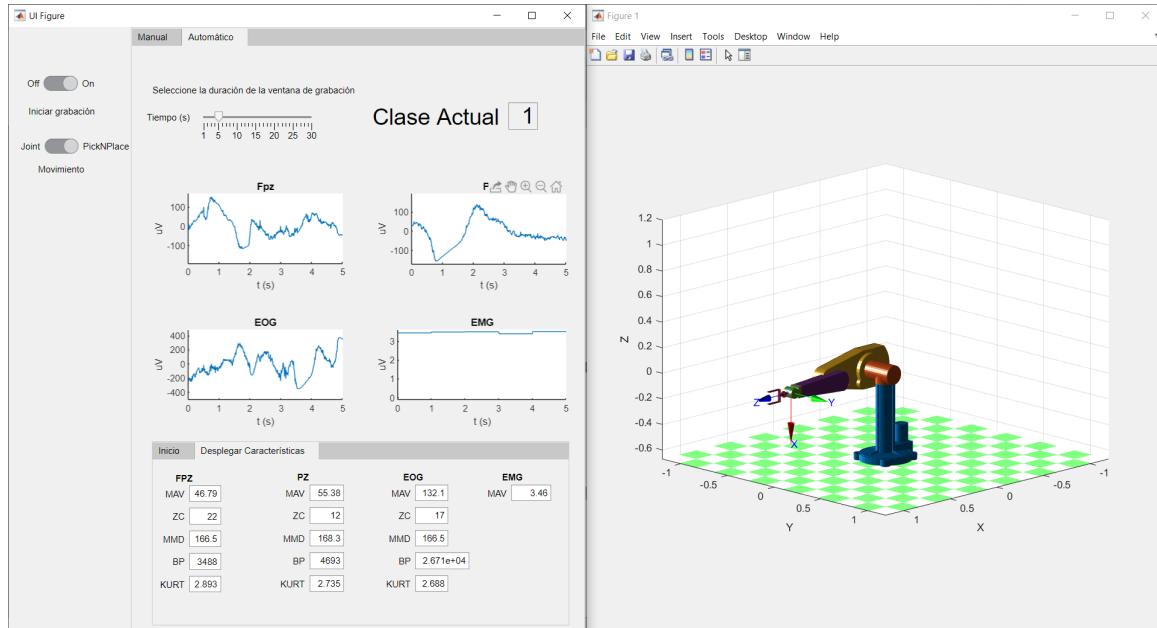


Figura 46: Control Automático de 5 segundos.

También se logra observar que las muestras se recopilan según el periodo de muestreo esperado, dictado por la interrupción del Arduino como se mencionó en las secciones anteriores. Por lo que se verifica el correcto funcionamiento de la BCI con señales EEG.

CAPÍTULO 11

Conclusiones

- Es posible simular un mecanismo de adquisición, filtrado y visualización en tiempo real de señales EEG a partir de una base de datos.
- Las características en el dominio del tiempo presentan rendimientos mayores a 80 % muy similares a las características en el dominio tiempo-frecuencia con un coste computacional menor.
- Las máquinas de soporte vectorial con kernel cúbico y las redes neuronales son los clasificadores que mejor rendimiento presentan para señales EEG.
- Es posible controlar robots a partir de señales EEG y graficar en tiempo real las señales recopiladas.
- Las características seleccionadas afectan el rendimiento del clasificador, deben ser significativas y no en exceso.
- El rendimiento de un clasificador automático depende de la calidad y cantidad de los datos de entrenamiento.
- Los filtros IIR Butterworth de segundo orden funcionan de la manera esperada y tienen un bajo coste computacional para filtrar señales EEG.
- El mejor rendimiento de clasificación se obtuvo utilizando ZC, MAV, MMD, Curtosis y Potencia Media para un SVM con kernel polinomial de grado 3.

CAPÍTULO 12

Recomendaciones

- Al implementar la BCI se debe calibrar momentos antes de su uso, los resultados del clasificador varían dependiendo las condiciones de la grabación.
- Al momento de calibrar la BCI, hacer énfasis en los datos de entrenamiento para asegurar un buen rendimiento por parte de la clasificación.
- Tener el equipo de recopilación de señales EEG para grabar gestos propios se considera fundamental para así probar el rendimiento de los clasificadores con diferentes tipos de gestos.
- Mejorar el rendimiento de los clasificadores probando diferentes combinaciones de características, como por ejemplo características en el dominio de la frecuencia.

CAPÍTULO 13

Bibliografía

- [1] P. Schembri, R. Anthony y M. Pelc, “Detection of Electroencephalography Artefacts using Low Fidelity Equipment”, ene. de 2017, págs. 65-75. DOI: 10 . 5220 / 0006398500650075.
- [2] K. Correa Arana y A. Vivas Alba, “Protesis de mano virtual movida por señales encefalográficas EEG”, es, *Prospectiva*, vol. 14, págs. 99-110, dic. de 2016, ISSN: 1692-8261. dirección: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S1692-82612016000200099&nrm=iso.
- [3] R. Sepúlveda, M. Oscar, G. Díaz, D. Gutierrez y O. Castillo, “Clasificación de señales encefalográficas mediante redes neuronales artificiales”, Español, *Computación y Sistemas*, 2015, ISSN: 1405-5546. dirección: <https://www.redalyc.org/articulo.oa?id=61536854006>.
- [4] J. Esqueda Elizondo, L. Hernández-Manzo y M. Pinto-Ramos, “Manipulación de un brazo robótico mediante señales electroencefalográficas”, *Revista de Tecnología e Innovación*, vol. 3, págs. 89-98, jun. de 2016.
- [5] L. J. Gómez Figueroa, “Análisis de señales EEG para detección de eventos oculares, musculares y cognitivos”, págs. 1-14, 2016.
- [6] S. Brewster, *Un robot dirigido con la mente ayuda con telepresencia a personas discapacitadas*, Visitado por última vez 04 de abril de 2020., 2015. dirección: <https://www.technologyreview.es/s/5373/un-robot-dirigido-con-la-mente-ayuda-con-telepresencia-personas-discapacitadas>.
- [7] R. Ceres, M. A. Mañanas y J. M. Azorín, “Interfaces y Sistemas en Rehabilitación y Compensación Funcional para la Autonomía Personal y la Terapia Clínica”, *Revisa Iberoamericana de Automática e Informática industrial*, vol. 8, n.º 2, págs. 5-15, 2011, ISSN: 1697-7920. DOI: 10 . 1016/S1697-7912(11)70021-8. dirección: <https://polipapers.upv.es/index.php/RIAI/article/view/8576>.
- [8] J. Aguirre, “Diseño, Análisis, y Desarrollo de un Sistema de Entrenamiento para Mejorar el Desempeño de los Atletas del Comité Olímpico Guatemalteco”, 2018.

- [9] M. Godoy, “Diseño e implementación de módulo de recopilación de señales y módulo de retroalimentación”, 2018.
- [10] J. P. Muñoz, “Diseño de un sistema inteligente de monitoreo de ondas EEG y generador de pulsos binaurales para combatir desórdenes de sueño en los atletas”, págs. 1-54, 2019.
- [11] D. L. Hora, *La probabilidad de sufrir una amputación de pie es 15 veces mayor en un diabético, según experto*, Visitado por última vez 04 de abril de 2020., 2018. dirección: <https://www.elpais.cr/2018/07/28/los-retos-de-vivir-con-una-protesis-en-guatemala/>.
- [12] P. Pernas, *Los retos de vivir con una prótesis en Guatemala*, Visitado por última vez 04 de abril de 2020., 2019. dirección: <https://lahora.gt/la-probabilidad-de-sufrir-una-amputacion-de-pie-es-15-veces-mayor-en-un-diabetico-segun-experto/>.
- [13] Brainworks, *WHAT ARE BRAINWAVES?*, Visitado por última vez 04 de abril de 2020. dirección: <https://brainworksneurotherapy.com/what-are-brainwaves>.
- [14] A. Najmi, *WHAT IS MACHINE LEARNING?*, Visitado por última vez 04 de abril de 2020., 2017. dirección: <https://supplychainbeyond.com/what-is-machine-learning/>.
- [15] Numerentur, *Máquina de Soporte Vectorial SVM*, Visitado por última vez 03 de junio de 2020., 2018. dirección: <http://numerentur.org/svm/>.
- [16] S. Khalid, T. Khalil y S. Nasreen, “A survey of feature selection and feature extraction techniques in machine learning”, *Proceedings of 2014 Science and Information Conference, SAI 2014*, págs. 372-378, oct. de 2014. DOI: 10.1109/SAI.2014.6918213.
- [17] A. Phinyomark, S. Hirunviriya, C. Limsakul y P. Phukpattaranont, “Evaluation of EMG Feature Extraction for Hand Movement Recognition Based on Euclidean Distance and Standard Deviation”, vol. 2010, mayo de 2010, págs. 856-860.
- [18] MATHWORKS, *Variance*, Visitado por última vez 12 de septiembre de 2020., 2020. dirección: <https://es.mathworks.com/help/matlab/ref/var.html#bundkwe-1>.
- [19] ——, *Kurtosis*, Visitado por última vez 11 de septiembre de 2020., 2020. dirección: <https://es.mathworks.com/help/stats/kurtosis.html>.
- [20] ——, *Measure the Power of a Signal*, Visitado por última vez 11 de septiembre de 2020., 2020. dirección: <https://es.mathworks.com/help/signal/ug/measure-the-power-of-a-signal.html?lang=en>.
- [21] K. Aboalayon, M. Faezipour, W. Almuhammadi y S. Moslehpoor, “Sleep Stage Classification Using EEG Signal Analysis: A Comprehensive Survey and New Investigation”, *Entropy*, vol. 18, n.º 9, pág. 272, ago. de 2016, ISSN: 1099-4300. DOI: 10.3390/e18090272. dirección: <http://dx.doi.org/10.3390/e18090272>.
- [22] R. González, *Algoritmo basado en Wavelets aplicado a la detección de invendios forestales*. Visitado por última vez 12 de septiembre de 2020., 2010. dirección: http://catarina.udlap.mx/u_dl_a/talles/documentos/mel/gonzalez_g_ra/capitulo3.pdf.
- [23] P. I. Corke, *Robotics, Vision & Control: Fundamental Algorithms in MATLAB*, Second. Springer, 2017, ISBN 978-3-319-54412-0.

- [24] P. P. y M. Vetterli, *Signal Processing for Communications*, EPFL Press. [Online], 2008. dirección: <https://doi.org/10.1201/9781439808009>.
- [25] e. a. M. Martínez L. Gómez, *Introducción al Procesado Digital de Señales*, Universidad de Valencia. [Online], 2009. dirección: <http://ocw.uv.es/ingenieria-y-arquitectura/1-1/tema1.pdf>.
- [26] J. W. Nilsson y S. Riedel, *Electric Circuits (8th Edition)*. USA: Prentice-Hall, Inc., 2007, ISBN: 0131989251.
- [27] e. a. M. Martínez L. Gómez, *Revisión de los tipos de filtros analógicos más comunes*, Universidad de Valencia. [Online], 2009. dirección: http://ocw.uv.es/ingenieria-y-arquitectura/filtros-digitales/tema_2._revision_de_los_tipos_de_filtros_analogicos_mas_comunes.pdf.
- [28] S. Xie, *Practical Filter Design Challenges and Considerations for Precision ADCs*. Visitado por última vez 18 de septiembre de 2020., 2016. dirección: <https://www.analog.com/en/analog-dialogue/articles/practical-filter-design-precision-adcs.html#>.
- [29] L. Rivera, N. Smith y G. Desouza, “High-accuracy recognition of muscle activation patterns using a hierarchical classifier”, ago. de 2014, págs. 579-584. DOI: 10.1109/BIOROB.2014.6913840.
- [30] D. M. G. Schalk, *BCI2000: A General-Purpose Brain-Computer Interface (BCI) System*. Visitado por última vez 19 de septiembre de 2020., 2004. dirección: <https://doi.org/10.13026/C28G6P>.
- [31] B. Shoelson, *edfRead - MathWorks*, Visitado por última vez 19 de septiembre de 2020., 2020. dirección: <https://es.mathworks.com/matlabcentral/fileexchange/31900-edfread>.
- [32] E. Rastgar, *EventRead - MathWorks*, Visitado por última vez 19 de septiembre de 2020., 2018. dirección: <https://es.mathworks.com/matlabcentral/answers/375362-how-can-i-read-edf-event-file-since-i-have-corresponded-edf-file-in-matlab>.
- [33] A. Z. B. Kemp, *Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG*. Visitado por última vez 19 de septiembre de 2020., 2000. dirección: <https://doi.org/10.13026/C2X676>.
- [34] *R17 Robot Manual*, ST Robotics, Cambridge, Reino Unido, 2017.
- [35] A. Topalidou-Kyniazopoulou, “Motion Planning Strategy For a 6-DOFs Robotic Arm In a Controlled Environment”, M.S. thesis, Dept. Computer Science. Ger., Bonn Univ., Bonn, Alemania, 2017. dirección: https://www.ais.uni-bonn.de/theses/Angeliki_Topalidou-Kyniazopoulou_Master_Thesis_07_2017.pdf.
- [36] A. Gil, *ARTE A Robotic Toolbox for Education in MATLAB*. Elche, España: Universidad Miguel Hernández, 2014. dirección: http://arvc.umh.es/arte/index_en.html.
- [37] *USING THE PUMA 560 ROBOT*, Jerry Rutherford, 2012. dirección: <http://rutherford-robotics.com/PUMA/%USING%5C%20THE%5C%20PUMA%5C%20ROBOT.pdf>.

CAPÍTULO 14

Glosario

Aprendizaje Automático Subdisciplina de la inteligencia artificial capaz de crear o refinar autónomamente modelos de decisión. 13, 32

Arduino Plataforma de desarrollo de *software* libre, basada en una placa de *hardware* libre que incorpora un microcontrolador reprogramable. 24

BCI Acrónimo de Interfaz Cerebro Computador, es un sistema capaz de medir la actividad cerebral, traducirla y ejecutar acciones a partir de las mismas. 1, 3, 7, 8, 12, 13, 39, 46

EEG Acrónimo de electroencefalograma, es un método que detecta la fluctuación de voltaje que resulta de la corriente iónica entre las neuronas del cerebro en un determinado período de tiempo. 1, 3, 4, 6–8, 10–12, 22–24, 35, 48, 54, 55

EMG Acrónimo de electromiograma, es un método que detecta la actividad muscular midiendo la diferencia de potencial entre dos o más músculos. 6, 24, 35, 48

EOG Acrónimo de electrooculograma, es un método que detecta la actividad ocular midiendo la diferencia de potencial entre la córnea y la retina. 24, 35, 48

hipnogramas Representaciones gráficas de las diferentes etapas del sueño a través del tiempo. 23

kernel Conjunto de funciones matemáticas utilizadas para transformar datos de entrada a la forma deseada. 30, 34, 54

manipulador serial Serie de eslabones unidos entre sí por juntas prismáticas o revolutas. 18, 39, 40, 43, 44

MATLAB Es un programa computacional numérico interactivo con el usuario que posee un lenguaje de programación propio. 24, 30, 32, 39, 44, 46, 48

polisomnografías Pruebas multiparamétricas que registran actividad cerebral, respiración, actividad muscular y niveles de oxígeno que sirve como diagnóstico y estudia el sueño mientras se duerme.. 23, 24

RNA Acrónimo de Redes Neuronales Artificiales, es un tipo de aprendizaje automático supervisado organizado por capas que respetan estructura jerárquica similar a la del cerebro. 26

SVM Acrónimo de Máquina de Soporte Vectorial, es un algoritmo de aprendizaje automático supervisado que se emplea para clasificación y regresión binaria. 14, 26, 29, 30, 32–34, 36, 37