



**Centro de Investigación y de Estudios Avanzados del Instituto
Politécnico Nacional**

Unidad Zacatenco

**Decodificación continua de lateralidad durante la
intención de movimiento de miembros superiores
basada en el Potencial de Preparación (RP)**

Tesis que presenta

Alessio Daniel Hernández Rojas

**para obtener el grado de Maestro en Ciencias en la
especialidad de**

Control Automático

Director de la Tesis

Dr. Juan Manuel Ibarra Zannatha

Ciudad de México, México

Diciembre 2020

Prefacio

El presente trabajo fue realizado como parte de la tesis de maestría en Control Automático en el Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional, Unidad Zacatenco.

El contenido de este reporte esta dirigido a estudiantes e investigadores con interés en el campo de las interfaces cerebro-computadora, robótica, visión por computadora, rehabilitación y áreas relacionadas.

Resumen

Objetivos: Las Interfaces Cerebro-Computadora (BCI) permiten decodificar la actividad neuronal y pueden ser usadas para activar dispositivos de asistencia y/o rehabilitación de miembro superior. El objetivo de esta tesis es evaluar la clasificación de lateralidad, es decir, derecha o izquierda, durante la intención de movimientos ejecutados de miembros superiores, así como desarrollar un primer prototipo de manipulación guiado por visión para realizar una tarea *toma y coloca* a fin de trabajar en conjunto con una BCI con control-compartido.

Método: En este estudio se analizaron datos de electroencefalografía (EEG) de una base de datos pública (EEGMMIDB) y se llevó a cabo un análisis pseudo-en-línea (*pseudo-online*) entre-sujetos ($n=18$) que ejecutaron dos tareas: Abrir y cerrar muñeca **1**) derecha y **2**) izquierda, en un paradigma basado en estímulo. Por cada sujeto, cada tarea fue ejecutada al menos 20 veces. Se extrajeron independientemente características temporales del Potencial de Preparación (RP) tales como amplitudes de muestras en ventanas de tiempo de distintos tamaños y características morfológicas en una ventana fija previa al inicio del movimiento, ambas de canales que cubren la corteza motora. Finalmente se evaluó la clasificación mediante tres clasificadores: Análisis Discriminante Lineal (LDA), Máquina de Soporte Vectorial (SVM), y Regresión Logística (LR), usando una Validación Cruzada 5×5 pliegues en términos de la precisión. La Aplicación BCI propuesta consistió de un sistema conformado por un brazo robot de 5 GDL y un algoritmo de visión artificial, el cual puede ejecutar una tarea de *toma y coloca* dado un comando sintético que emula el estado mental proveído por una BCI.

Resultados: Se mostró que usando características de amplitudes de muestras del RP y usando un clasificador discreto SVM se puede clasificar la lateralidad con una precisión promedio máxima aceptable de 71.38 % alrededor del inicio de movimiento y que se pueden obtener precisiones por arriba del nivel de oportunidad empírico $\sim 53\%$ hasta ~ 0.3 s antes del inicio del movimiento. Con respecto al sistema de manipulación, este primer prototipo logró ejecutar la tarea demandada exitosamente bajo ciertas condiciones de calibración geométrica y de segmentación.

Significancia: Este estudio mostró que usando características (amplitudes de muestras) únicamente del RP y un clasificador SVM lineal es posible clasificar lateralidad durante la intención de movimientos ejecutados con precisiones $\sim < 75\%$ antes del inicio de movimiento. Esta relativamente baja precisión durante la intención de movimiento, la asociamos a la alta variabilidad ensayo-a-ensayo propia del RP, en particular en un paradigma basado en estímulo. Por lo anterior, se concuerda con la literatura de BCI actual en que es necesario desarrollar e implementar técnicas desde el punto de vista del decodificador que permitan mejorar la relación señal-ruido del RP a fin de predecir lateralidad con altas precisiones durante la intención de movimiento únicamente con características del RP.

Abstract

Objective: Brain-Computer Interfaces (BCI) allow decoding of neural activity and can be used to activate assistive devices and/or upper limb rehabilitation. The objective of this thesis is to evaluate the classification of laterality, i.e., right or left, during the intention of executed movements of the upper limbs, as well as to develop a first prototype of vision-guided manipulation to perform a *pick and place* task in order to work in conjunction with a BCI with shared-control.

Methods: In this study, electroencephalography (EEG) data from a public database (EEGMMIDB) was analyzed in a pseudo-online way between-subjects ($n = 18$) that performed two tasks: Open and close wrist **1)** right and **2)** left, in a cue-based paradigm. For each subject, each task was executed at least 20 times. Temporal characteristics of the Readiness Potential (RP) were independently extracted, such as sample amplitudes in time windows of different sizes and morphological characteristics in a fixed window prior to the onset of movement, both from channels that cover the motor cortex. Finally, the classification was evaluated using three classifiers: Linear Discriminant Analysis (LDA), Support Vector Machine (SVM), and Logistic Regression (LR), using a 5×5 -fold cross-validation in terms of the accuracy. The proposed BCI Application consisted of a system made up of a 5-DOF robot arm and an artificial vision algorithm, which can execute a *pick and place* task given a synthetic command that emulates the mental state provided by a BCI.

Results: It was shown that by using RP sample amplitude characteristics and using a discrete SVM classifier, laterality can be classified with a maximum acceptable average accuracy of 71.38 % around the onset of movement and that accuracies above the empirical chance level ~ 53 % can be obtained up to ~ 0.3 s before the onset of the movement. Regarding the manipulation system, this first prototype managed to execute the demanded task successfully under certain conditions of geometric calibration and segmentation.

Significance: This study showed that using characteristics (sample amplitudes) only from the RP and a linear SVM classifier it is possible to classify laterality during the intention of executed movements with accuracies $\sim < 75$ % before the onset of movement. This relatively low accuracy during the intention of movement, we associate it to the high trial-to-trial variability characteristic of the RP, in particular in a cue-based paradigm. Therefore, it is agreed with the current BCI literature that it is necessary to develop and implement techniques from the decoder point of view that allow to improve the signal-to-noise ratio of the RP in order to predict laterality with high accuracy during motor intention only with RP characteristics.

Agradecimientos

Primeramente, agradezco al DCA-CINVESTAV, por proporcionarme un espacio para trabajar y la oportunidad de desarrollarme profesionalmente, así como a CONACYT por proporcionarme el apoyo económico durante mis estudios de posgrado.

Gracias a mi asesor, el Dr. Juan M. Ibarra por su guía durante esta investigación, así como por su tiempo, paciencia e instrucción.

Gracias a mis compañeros de laboratorio: Óscar, Jorge, Santos, Pablo, Alexis e Isaac, por hacer más ameno este camino, así como por su enseñanza.

Gracias a la Dra. Jessica Cantillo y al Dr. Rubén Cariño por el apoyo intelectual y material para hacer posible este trabajo.

Gracias a mi Jurado: la Dra. Petra Wiederhold, Dra. Jessica Cantillo, el Dr. Moisés Bonilla y el Dr. Isaac A. Pérez, por su valiosa dedicación a la revisión de este trabajo.

Por último, pero no menos importante gracias a mis padres, hermanos, amigos y doctores de vida, por todo su apoyo y comprensión durante mi vida académica, sin los cuales este trabajo no hubiera sido posible.

Índice general

Prefacio	I
Resumen	II
Agradecimientos	v
1. Introducción	1
1.1. Enfermedad Vasculat Cerebral (EVC)	1
1.1.1. Plasticidad neuronal	2
1.2. Planteamiento del problema	3
1.3. Trabajo relacionado	4
1.4. Objetivos	7
1.5. Descripción del trabajo	7
2. Interfaz Cerebro-Computadora (BCI)	9
2.1. Definición de BCI	9
2.2. Esquema general de un sistema BCI	10
2.3. Adquisición de la señal	12
2.3.1. Electroencefalografía (EEG)	13
2.3.2. Señales de control en BCIs	14
2.4. Preprocesamiento	16
2.4.1. Filtrado temporal	17
2.4.2. Submuestreo y normalización	18
2.4.3. Filtrado espacial	18
2.5. Extracción de características	24
2.6. Clasificación	25

2.6.1.	Análisis Discriminante Lineal (LDA)	26
2.6.2.	Máquina de Soporte Vectorial (SVM)	29
2.6.3.	Regresión Logística	31
2.7.	Desempeño de BCI	32
2.7.1.	Medida de evaluación de clasificador	32
2.7.2.	Evaluación del clasificador	33
2.7.3.	Significancia estadística para precisión de clasificación	34
3.	Potencial Cortical Relacionado a Movimiento (MRCP)	36
3.0.1.	Definición y componentes del MRCP	36
3.0.2.	Registro	38
3.0.3.	Procesamiento de señal de MRCP para detección de la intención de movimiento	39
3.0.4.	Factores que afectan el MRCP	41
4.	Visión Artificial	42
4.1.	Sistema de Visión Artificial	42
4.2.	Técnicas de Procesamiento de Imagen	45
4.2.1.	Espacio de color	45
4.2.2.	Segmentación	45
4.2.3.	Filtrado espacial	46
4.2.4.	Filtro Mediana	47
4.2.5.	Filtro Bilateral	48
4.3.	Análisis de imagen	48
4.3.1.	Homografía	48
5.	Métodos	52
5.1.	Base de datos	52
5.1.1.	Paradigma	52
5.1.2.	Registro	54
5.1.3.	Estructura de datos	54
5.2.	Procesamiento de datos	55
5.2.1.	Selección de canales	55
5.2.2.	Software	55
5.2.3.	Preprocesamiento	57

5.2.4.	Análisis neurofisiológico	59
5.2.5.	Extracción de características	59
5.2.6.	Selección de características	61
5.2.7.	Selección de modelo	61
5.3.	Aplicación BCI	65
5.3.1.	Sistema de Visión	65
5.3.2.	Sistema del Robot	69
5.3.3.	Base de conocimiento: ROS	76
6.	Resultados	77
6.0.1.	Análisis Neurofisiológico	77
6.0.2.	Selección de características	78
6.0.3.	Clasificación Binaria de ensayo simple	79
6.0.4.	Aplicación BCI	84
7.	Discusión	85
7.0.1.	Análisis Neurofisiológico	85
7.0.2.	Clasificación Binaria de ensayo simple	86
7.0.3.	Limitaciones de este estudio	87
7.0.4.	Transferencia a control en línea	90
7.0.5.	Aplicación BCI	92
8.	Conclusión y Trabajo a futuro	94
8.1.	Contribuciones	95
	Bibliografía	95
	Anexos	116
A.	Artículo de conferencia COMROB 2019	116

Índice de tablas

1.1. Tratamientos convencionales de rehabilitación para recuperar funciones motoras	3
5.1. Parámetros D-H del robot	71
6.1. Mejores resultados de cada clasificador para las características propuestas obtenidas del MRCP	83
7.1. Descripción de algunos estudios EEG reportados en la literatura de clasificación (binaria) de lateralidad de miembros superiores	88

Índice de figuras

2.1. Protocolos de operación de una BCI	10
2.2. Esquema general de la operación de un sistema BCI	11
2.3. Características de métodos de neuroimagen.	12
2.4. Sistema internacional 10/20 (izquierda) y 10/10 (derecha) para registro EEG clínico	14
2.5. Principales señales de control en BCIs	15
2.6. Filtros espaciales independientes de datos	21
2.7. Análisis Discriminante Lineal (LDA)	28
2.8. SVM de margen suave	29
2.9. Matriz de confusión de un problema de clasificación binaria	32
2.10. Esquema típico de clasificación	33
3.1. MRCPs de pre-movimiento	37
3.2. Procesamiento para detección de MRCP	39
4.1. Diagrama de un Sistema de Visión Artificial para robots	43
4.2. Espacio de color HSV	45
4.3. Modelo de proyección en perspectiva	49
5.1. Protocolo experimental de base de datos EEGMMIDB	53
5.2. Montaje de EEG de la base de datos EEGMMIDB	54
5.3. Canalización del procesamiento de datos	56
5.4. Ejemplo de las características morfológicas consideradas en un MRCP de ensayo simple	60
5.5. Perspectiva general de nuestra Aplicación BCI	65
5.6. Sistema de manipulación propuesto	66

5.7.	Principales etapas del algoritmo de visión artificial propuesto	67
5.8.	Brazo robótico de 5 GDL	70
5.9.	Geometrías para resolver la cinemática inversa del robot	73
5.10.	Arquitectura de control del robot manipulador	75
6.1.	Análisis neurofisiológico de movimiento de mano derecha/izquierda . . .	78
6.2.	Mapa de calor de <i>puntaje-z</i> de las pruebas estadísticas <i>Wilcoxon Rank-Sum</i>	79
6.3.	<i>coeficiente-r²</i> para las características morfológicas propuestas del MRCP	79
6.4.	Precisión de clasificación de gran promedio usando características de muestras	81
6.5.	Precisión de clasificación de gran promedio usando características mor- fológicas	82
6.6.	Comparación de la precisión de gran promedio entre clasificadores y características propuestas	83
6.7.	Comparación de coordenadas cartesianas durante tarea <i>toma y coloca</i> .	84

Capítulo 1

Introducción

El Banco Mundial y la Organización Mundial de la Salud (OMS), estiman que en 2015 había cerca de un billón de personas, equivalentes a 15 % de la población mundial con algún tipo de discapacidad, teniendo mayor prevalencia en países en desarrollo. De éstas, un quinto, es decir, cerca de 190 millones de personas, experimentan discapacidades significativas.

1.1. Enfermedad Vascul ar Cerebral (EVC)

La Enfermedad Vascul ar Cerebral (EVC) o “ataque cerebral ” hace referencia a cualquier trastorno caracterizado por alguna falla de circulación sanguínea al cerebro, generalmente de comienzo brusco (de ahí su nombre *ictus* en latín o *stroke* en inglés). Existen dos principales tipos de EVC, el evento vascul ar cerebral isquémico (eVC isquémico) provocado por la obstrucción del flujo sanguíneo y el evento vascul ar cerebral hemorrágico (eVC hemorrágico) en donde se presenta la rotura de un vaso sanguíneo, causando un sangrado dentro del cerebro. La EVC es la segunda causa de muerte y la primera causa neurológica de discapacidad en el mundo [Choreño-Parra et al., 2019] y se espera que la prevalencia de su carga relacionada aumente en las próximas dos décadas [Langhorne et al., 2011].

La recuperación por EVC es heterogénea en su naturaleza, pues algunos eVCs pueden ser leves y tener efectos temporales, sin embargo otros pueden ser bastante graves y ocasionar discapacidades permanentes, según el tiempo durante el cual el cerebro careció de flujo sanguíneo y cual fue el área afectada. Tales discapacidades pueden ser motoras,

sensoriales, visuales, cognitivas u orgánicas [Gutiérrez and González, 2014], siendo la motora la más frecuente, considerada como la pérdida o limitación de funciones en el control muscular. Las funciones afectadas por tal déficit motor que experimentan alrededor del 80% de pacientes son el control del movimiento de la cara, brazos y piernas de un lado del cuerpo, conocido como *hemiparesia* [Langhorne et al., 2009a].

A pesar de avances impresionantes en el manejo médico de la EVC, sin un tratamiento eficaz, la mayoría de los cuidados posteriores a un eVC continuarán dependiendo de las intervenciones de rehabilitación [Langhorne et al., 2009b] con el objetivo principal de recuperar una función motora normal [Thickbroom, 2007].

1.1.1. Plasticidad neuronal

La *neuro-plasticidad* es el mecanismo básico que subyace a la recuperación de las funciones motoras después de un eVC [Chang and Kim, 2013]. La plasticidad neuronal se define como la capacidad del sistema nervioso para reorganizar su estructura, función y conexiones [Cramer et al., 2011]. Es importante resaltar que la inducción de plasticidad puede ocurrir de tres maneras diferentes: por aprendizaje, por lesión o inducida artificialmente [Thickbroom, 2007].

Algunos tratamientos convencionales para la inducción de plasticidad a fin de recuperar funciones motoras específicas se describen en la Tabla 1.1. La mayoría de estos métodos se basan en la teoría del aprendizaje motor que supone que *a)* el re-aprendizaje motor es comparable como el aprendizaje motor y *b)* pacientes pueden aprender [van Dokkum et al., 2015]. Adicionalmente, la mayoría de las herramientas de rehabilitación requieren un nivel residual del control motor para llevar a cabo las tareas terapéuticas requeridas a fin de tener algo para proveer realimentación.

Para pacientes con deficiencias severas (es decir, poco o ningún control del movimiento), las estrategias convencionales pueden no ser suficientes. Es en estas situaciones donde nuevas técnicas son requeridas, y una de ellas son las interfaces cerebro-computadora. Una *interfaz cerebro-computadora* (BCI) involucra el registro y decodificación de señales cerebrales mientras se intentan/ejecutan/imaginan tareas motoras y/o cognitivas. Al proveer una realimentación útil, que puede ser ya sea visual, auditiva o táctil, del movimiento deseado y por lo tanto restaurando el *acoplamiento acción-percepción*, la BCI ha mostrado inducir plasticidad neuronal [van Dokkum et al., 2015].

Fuente: [Langhorne et al., 2009a]

Tratamiento	Descripción	Objetivo
Terapia de Movimiento Inducida por restricción (CIMT)	Terapia en la cual se restringe el movimiento del miembro no afectado, realizando una práctica intensiva de una tarea específica en la extremidad afectada.	Miembro superior
Entrenamiento de tarea repetitiva	Tratamiento en el cual simplemente se practica la tarea por si misma.	Miembro superior Miembro inferior
Práctica mental con imaginación motora	Proceso a través del cual el paciente ensaya mentalmente repetidamente una acción o tarea sin realmente realizarla físicamente.	Miembro superior Miembro inferior
Bio-realimentación mediante electromiografía(EMG)	Técnica en la cual se registra una diferencia de potencial a lo largo de un músculo usando electrodos sobre la superficie de la piel. Posteriormente, la señal normalmente se muestra en un formato simplificado al paciente, p. ej. un tono auditivo o una respuesta en pantalla.	Miembro superior Miembro inferior
Estimulación eléctrica	Técnica en la cual se administra estimulación eléctrica al sistema neuromuscular periférico por medio de electrodos externos o internos, diseñada para mejorar el control voluntario del movimiento.	Miembro superior Miembro inferior
Robótica	Hace referencia a los tratamientos que usan robots para la rehabilitación. Los dispositivos pueden ser de dos clases principales: <i>robots terapeuticos</i> , desarrollados para entrenar la función motora perdida (p. ej. robots basados en exoesqueletos o en efector final) y <i>robots de asistencia</i> , diseñados para compensar las habilidades perdidas (p. ej. exoesqueletos, prótesis, robots para tareas de manipulación, sillas de ruedas, vehículos, etc) [Klamroth-Marganska, 2018].	Miembro superior Miembro inferior

Tabla 1.1: Tratamientos convencionales de rehabilitación para recuperar funciones motoras

1.2. Planteamiento del problema

La hemiparesia producida por un eVC afecta las extremidades superiores, y menos de la mitad de los pacientes pueden recuperar la función adecuada del brazo 6 meses posterior al eVC [Wang et al., 2019].

En casos severos, la aplicabilidad de los métodos tradicionales puede verse fuertemente reducida, lo cual conlleva a una incapacidad para usar las extremidades superiores, afectando seriamente la independencia funcional de los pacientes para llevar a cabo actividades básicas de la vida diaria, tales como usar un tenedor, abrocharse la camisa, abrir una puerta, etc. De esta forma, asistir a un paciente en tareas de manipulación de objetos puede mejorar su calidad de vida substancialmente.

Como parte de las líneas de investigación del Laboratorio de Robótica y Visión Artificial del DCA-Cinvestav, se pretende desarrollar a futuro un sistema para asisten-

cia y/o rehabilitación para pacientes con déficit motor usando una Interfaz Cerebro-Computadora (BCI) con control-compartido. Es en este contexto, en el cual esta tesis propone investigar la decodificación fuera de línea y desarrollar un primer prototipo de manipulación para estos fines, los cuales serán una base para la futura implementación de una BCI motora no invasiva.

1.3. Trabajo relacionado

A continuación se presenta el estado del arte para la decodificación de señales EEG correspondientes de miembros superiores, así como de sistemas de manipulación para BCI.

Hasta ahora, para el desarrollo de BCIs no hay un estándar universal, por lo que es necesario identificar que señales de control reportadas en la literatura permiten resolver un problema particular. En nuestro caso, para emitir salidas que corresponden a DERECHA/IZQUIERDA es posible usar tanto potenciales evocados como potenciales espontáneos (relacionados-a-movimiento), sin embargo se prefieren estos últimos principalmente debido a que no requieren de estímulos externos. Las dos principales señales de control relacionadas con la intención de movimiento son la *desincronización relacionada a eventos* (ERD), un tipo de los ritmos sensoriomotores (SMR), y los *potenciales corticales relacionados a movimiento* (MRCP). En [Pfurtscheller et al., 2000] se propuso un sistema BCI basado en SMR para controlar una ortesis de mano para -agarre- en un paciente con lesión de la médula espinal de nivel alto. En este estudio de 160 ensayos se le solicitó al paciente imaginar movimiento de manos o pies derecha/izquierda. Se obtuvo una precisión del 65% en la clasificación `mano-DER. vs mano-IZQ.` para abrir y cerrar la ortesis de mano, respectivamente. Además, esta precisión fue mejorada al evaluar la precisión de combinaciones de clases, llegando a obtener precisiones hasta del 95% para `ambos-pies vs mano-DER.` Adicionalmente, una medición topográfica mostró cambios en el área de la corteza motora, indicando neuroplasticidad inducida a través de las sesiones de entrenamiento. Una desventaja de los SMR, es que se requiere de mucho entrenamiento antes de ver cambios funcionales [Han Yuan and Bin He, 2014]. Las BCIs basadas en MRCP ofrecen una potencial alternativa a las basadas en SMR, ya que no requieren de un período sustancial de entrenamiento. En [Bai et al., 2007] se realizó un estudio de intención de movimiento fuera de línea usando MRCP y ERD en 12 sujetos sanos (11 diestros y 1 zurdo), donde se les pidió a cada uno que

ejecutaran a ritmo propio movimientos para una secuencia de 3 teclas sobre un teclado de computadora, ya sea con mano derecha/izquierda. El análisis promedio de estas señales, mostró que los MRCP fueron lateralizados al hemisferio izquierdo contralateral antes del movimiento de mano derecha y los MRCP fueron bilaterales (contralateral+ipsilateral) antes del movimiento de mano izquierda, mientras que para ERD, fue lateralizado al hemisferio izquierdo contralateral al movimiento de mano derecha, pero ERD fue bilateral para movimiento de mano izquierda. Además, mediante un *análisis de separabilidad de características* usando como criterio la *distancia de Bhattacharyya*, se mostró que este criterio es pequeño en la banda α (8-12 Hz) y cercana a corriente directa (CD) (en la cual reside el MRCP), posiblemente debido a la varianza entre los ensayos, y largo en la corteza sensoriomotora derecha de la banda β (12-26 Hz) (indicando alta separabilidad), siendo C4 el canal de máxima separabilidad. Esto concuerda con [Kawashima et al., 1993], [Volkman et al., 1998], [Jung et al., 2003] de que el hemisferio contralateral está predominantemente involucrado con el movimiento de mano dominante, mientras que los hemisferios contralateral e ipsilateral están involucrados con el movimiento de mano no dominante. De esta manera, usando la ERD de banda β y canales de la corteza sensoriomotora derecha, [Bai et al., 2007] obtuvo la mejor precisión de hasta 75 % para épocas de 1s (-1s a 0s con respecto al inicio de electromiografía (EMG)) para **mano derecha vs mano izquierda**. En [Alomari et al., 2013] se realizó una clasificación fuera de línea de ejecución de movimiento **mano derecha vs mano izquierda** de 6 sujetos de la base de datos *Physionet* [Goldberger et al., 2000]. Aquí se analizó solamente canales fijos (C3, C4 y Cz) y se obtuvieron características (P:potencia, M:promedio, E:energía y X:tipo-de-muestra (ERS/ERD/MRCP)) de los componentes independientes (ICs), obteniendo una precisión máxima de 97.1 % usando las características P, E y X. Tal precisión es alta, sin embargo puede deberse al pequeño número de sujetos de estudio. Es importante notar que en este estudio se incluyó una característica posterior al inicio del movimiento, i.e., ERS. En otros trabajos con la misma base de datos: *i*) [Huong and Linh, 2017] usando el mismo preprocesamiento y selección de características anterior, pero 109 sujetos, se obtuvo una precisión de 67 %, *ii*) [Alomari et al., 2014] usó el mismo preprocesamiento pero 20 sujetos y características de la transformada discreta de Wavelet consiguiendo la mejor precisión de 84.5 %, *iii*) [Roots, 2020] usando 103 sujetos y un clasificador basado en redes neuronales convolucionales (CNN) obtuvo una precisión de clasificación **mano derecha vs izquierda** de 84.1 % para movimientos ejecutados y 83.8 % para movimientos imagi-

nados.

En [Blankertz et al., 2002] se desarrolló un *interruptor-cerebral* basado en EEG usando movimiento de dedo índice derecho/izquierdo, analizando 1 sujeto y extrayendo como característica el *potencial de preparación* (RP) generado entre -400 y 0 ms antes de los movimientos reales (un componente de MRCP). Aquí, dos clasificadores fueron entrenados; el primero para *predecir movimiento*, conocido como *detector*, es decir, *derecha&izquierda vs descanso*, y el segundo para *predecir lateralidad* (*derecha vs izquierda*) del movimiento de dedo. En una evaluación pseudo-en-línea (*pseudo-online*) esta BCI mostró una precisión de clasificación alrededor de 90 %. Para nuestro conocimiento, los trabajos de [Blankertz et al., 2002, 2003, 2006], [Lew, 2012] son de los pocos que han investigado lateralidad de extremidades superiores usando puramente MRCP, pues los trabajos en MRCP durante los últimos años se han centrado en detección y clasificación de: *tipo de movimiento* como flexión, extensión, etc. y *características del movimiento*, tales como fuerza, velocidad, etc. Es importante resaltar que el diseño de una BCI basada en MRCP, es principalmente útil durante la intención de movimiento (es decir, antes del inicio del movimiento (RP)), pues incluso en movimientos repetitivos la discriminación decae alrededor de 1 segundo después [Blankertz et al., 2006]. Por otra parte, los SMRs ofrecen un control más continuo y hasta hoy han permanecido liderando el desarrollo de BCIs para tareas de movimiento [Al-Quraishi et al., 2018][Han et al., 2020].

En [Meng et al., 2016] se desarrolló una BCI para asistir una tarea *alcanzar y agarrar* con un brazo robot, en donde se estudiaron 13 sujetos y extrajeron características ERS/ERD de movimientos imaginados (mano derecha/izquierda, ambas manos, relajación de ambas manos). Esta tarea fue dividida en dos etapas: 1) el sujeto guiaba al brazo robótico en un plano 2D por encima del objeto objetivo y 2) si el sujeto seleccionaba el objeto correcto debía entonces guiar el brazo robótico hacia abajo en el eje z para agarrar el objeto. Este diseño secuencial simplifica el proceso de agarrar un objeto en el espacio 3D, sin embargo aumenta ligeramente el tiempo necesario para realizar la tarea completa. Finalmente, [Meng et al., 2016] obtuvo una precisión de hasta 80 % para tareas de agarre de cuatro o cinco objetivos.

Algunos trabajos que usan otras alternativas para BCIs e involucran *mano derecha vs mano izquierda* son: *i*) [Xu et al., 2015] en donde al evaluar 30 sujetos y extrayendo características basadas en la transformada de Wavelet de movimientos ejecutados, se logró controlar un grado de libertad (GDL) de un brazo robot y se obtuvo una precisión

de 91.5 %, *ii*) en [Tayeb et al., 2019] analizando 20 sujetos, movimientos imaginados, y usando modelos de *aprendizaje profundo*, los cuales prescindien de la ingeniería de características, se logró controlar un GDL de un brazo robot con una precisión de hasta 84 %.

Con respecto a BCIs de *control compartido*, es decir BCI + sensores-externos, relacionadas a *mano derecha vs mano izquierda*, recientemente en [Xu et al., 2019] se implementó una estrategia de este tipo, *BCI + visión por computadora*, para asistir una tarea *alcanzar y agarrar* con un brazo robótico de 6 GDL en un espacio 3D, analizando 5 sujetos, extrayendo características ERS/ERD y obteniendo una precisión fuera de línea mayor a 80 % y en línea mayor a 70 %. Aquí, durante el control *guiado-BCI*, los sujetos podían mover el brazo robot en el área circundante al objeto objetivo mediante la imaginación de movimiento, pero una vez que el efector final del robot llegará a la región guiada-por-visión, el robot agarraba el objetivo autónomamente. Otros trabajos pueden revisarse en [Tonin and Millán, 2021].

1.4. Objetivos

Así, el objetivo de este proyecto es:

- Evaluar la precisión de clasificación de lateralidad de intenciones de movimiento de extremidades superiores basados en características extraídas del Potencial de Preparación (RP), en un estudio fuera de línea.
- Desarrollar un primer prototipo de sistema de manipulación para realizar una tarea *toma y coloca*, dirigida para aplicación BCI con control-compartido.

1.5. Descripción del trabajo

Este trabajo se divide como se indica a continuación. Primeramente, en el [Capítulo 2](#) se describe cada uno de los componentes de una BCI, haciendo énfasis en aquellas basadas en electroencefalografía (EEG). A continuación, en el [Capítulo 3](#) se presentan las principales características del Potencial Cortical Relacionado a Movimiento (MRCP), así como su modo de registro y factores que lo afectan. Luego, en [Capítulo 4](#), se presenta la teoría de visión artificial requerida para el algoritmo de visión del sistema de manipulación propuesto. En [Capítulo 5](#) se describen la base de datos, la canalización

propuesta para la decodificación de lateralidad, así como cada subsistema del sistema de manipulación. Por otra parte, en [Capítulo 6](#) se presentan los resultados de decodificación de lateralidad basados en la precisión del clasificador, así como los resultados del seguimiento de posición de los actuadores del brazo manipulador durante la tarea *toma y coloca*. En [Capítulo 7](#) se discuten los resultados y limitaciones de la decodificación de lateralidad y de este primer prototipo para manipulación, así como consideraciones para transferencia a estudios en línea. Finalmente, en [Capítulo 8](#) se indica la conclusión y trabajo a futuro, resaltando las contribuciones de este estudio.

Capítulo 2

Interfaz Cerebro-Computadora (BCI)

El presente capítulo describe los principales elementos de una BCI. Primeramente se define y resalta la importancia de una BCI. Posteriormente, se discuten: la adquisición de señal, preprocesamiento, extracción de características, clasificación, así como medidas de desempeño de una BCI.

2.1. Definición de BCI

- “Una *interfaz cerebro-computadora* (Brain Computer Interface o BCI), también denominada *interfaz cerebro-máquina* (Brain Machine Interface o BMI), es un sistema de comunicación y/o control que permite interacción en tiempo real entre el cerebro humano y dispositivos externos” [Mak and Wolpaw, 2009].

De acuerdo a su protocolo de operación, una BCI se puede clasificar en (véase la [Figura 2.1](#)):

- *BCI síncrona (basada en estímulo)*. Actúa con base en un estímulo. Las BCIs de este tipo identifican los patrones del cerebro únicamente en ciertas ventanas de tiempo guiadas por un estímulo (relacionado con el comando BCI) que se le presenta al usuario [Wolpaw and Wolpaw, 2012].
- *BCI asíncrona (ritmo propio)*. No actúa con base en un estímulo. Aquí las BCIs pueden identificar patrones cerebrales sin restricciones de tiempo, de modo que el

usuario puede iniciar voluntariamente el control en cualquier momento [Wolpaw and Wolpaw, 2012].

Los protocolos de *ritmo propio* pueden producir interacciones más naturales y prácticas entre el usuario y la BCI, sin embargo, su desarrollo suele ser más difícil [Wolpaw and Wolpaw, 2012].

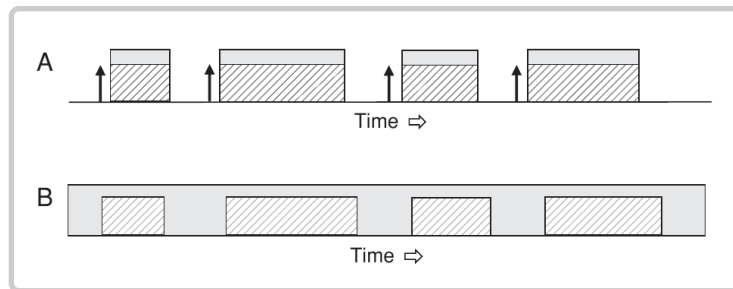


Figura 2.1: **Protocolos de operación de una BCI.** (A) Protocolo de operación síncrono (ritmo por estímulo). El área sombreada indica los tiempos en los que la BCI está disponible para control y la eclosión indica los tiempos cuando la BCI espera que el usuario provea los comandos de control. Las flechas indican las pistas dadas por la BCI previo al inicio de los períodos de uso de la BCI. (B) Protocolo de operación asíncrono (ritmo propio). El continuo sombreado indica que la BCI está siempre disponible al usuario, mientras que la eclosión indica períodos de control intencional. Los períodos de tiempo entre las eclosiones son períodos de no control.

Fuente: [Wolpaw and Wolpaw, 2012]

Algunos conceptos clave en investigación de BCIs y de comprensión para el presente trabajo son:

- *Muestras (samples)*. Valores adquiridos por el sistema en cada paso de tiempo
- *Época (epoch)*. Grupo de muestras sucesivas de tamaño predeterminado, usado para simplificar la extracción de características
- *Ensayo (trial)*. Secuencia en el plan experimental donde se le solicita al usuario ejecutar una tarea específica
- *Sesión (session)*. Grupo de ensayos realizados *sucesivamente* por el usuario

2.2. Esquema general de un sistema BCI

Para la operación de una BCI generalmente se requieren las siguientes etapas (véase la Figura 2.2):

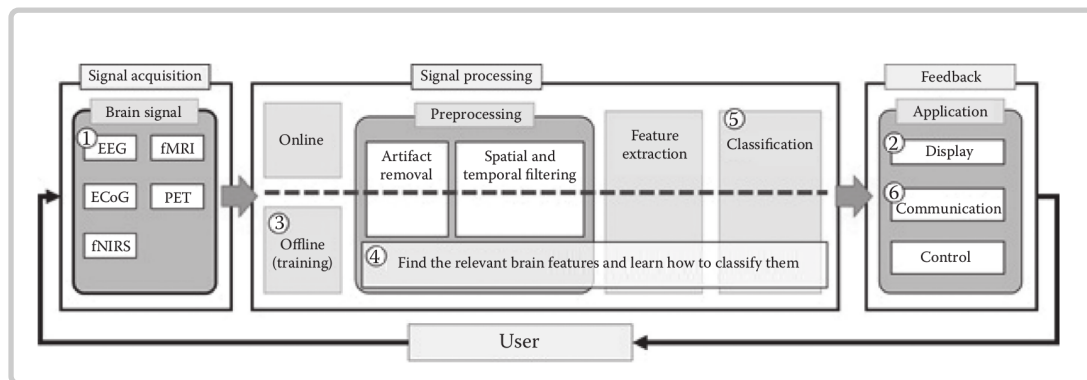


Figura 2.2: **Esquema general de la operación de un sistema BCI.** Un ejemplo práctico puede ser el siguiente. Alex, quien está paralizado pero cognitivamente intacto se comunica mediante una computadora. (1) Primeramente, en el laboratorio BCI se le adapta un sistema de adquisición por ejemplo EEG. (2) En seguida, mediante una pantalla se le presenta una matriz alfanumérica parpadeando al azar. (3) Se procede al entrenamiento del sistema para que responda apropiadamente a sus señales cerebrales. (4) Alex cuenta mentalmente cuantas veces parpadea el caracter que quiere escribir. El sistema BCI procesa los patrones cerebrales asociados al conteo mental para extraer las características relevantes. (5) Una vez que el entrenamiento está completo, Alex se enfoca otra vez en las letras que desea escribir, y el sistema BCI aplica el clasificador construido en la etapa de entrenamiento para detectar su intención. (6) Ahora, Alex puede escribir mensajes, usando únicamente sus ondas cerebrales [Nam et al., 2018].

Fuente: [Nam et al., 2018]

1. **Adquisición de la señal.** Etapa que consiste en el registro de la actividad cerebral, ya sea electrofisiológica, magnética o metabólica [Mak and Wolpaw, 2009].
2. **Procesamiento de señales.** Etapa necesaria para la interpretación de las señales cerebrales. Generalmente se aplican subetapas como *preprocesamiento*, *extracción de características*, y una de *clasificación* que genera la señal de control relacionada con los patrones mentales de interés, normalmente usando técnicas de *Aprendizaje Automático* (*Machine Learning* en inglés). Al conjunto dado por las etapas de adquisición y procesamiento se le conoce como *sistema BCI* [Graimann et al., 2010].
3. **Realimentación Sensorial.** Etapa que cierra el lazo de control, en la cual el usuario recibe algún tipo de realimentación, ya sea por medio de sus sentidos (visual, auditiva, táctil) o directamente a su cerebro mediante estimulación cerebral [Rao, 2013].

2.3. Adquisición de la señal

De acuerdo al método empleado para la adquisición de señales cerebrales, las BCIs se dividen en dos categorías: *Invasivas* y *No Invasivas*. Los métodos de registro invasivos son técnicas de neuroimagen en las que matrices de microelectrodos se implantan directamente en el tejido cerebral, por ejemplo la *electrocorticografía* (ECoG) y el *registro neuronal intracortical* (INR). Las técnicas no invasivas son aquellas que no requieren cirugía o implantación de químicos internos o máquinas, implicando menores riesgos para la salud, y generalmente se dividen en dos tipos: (1) *mediciones directas* que detectan la actividad electrofisiológica (como la *electroencefalografía* (EEG) y *magnetoencefalografía* (MEG)) y (2) *mediciones indirectas*, es decir, no relacionadas directamente con la actividad neuronal, como la respuesta hemodinámica [Nicolas-Alonso and Gomez-Gil, 2012], por ejemplo la *resonancia magnética funcional* (fMRI), *espectroscopía funcional de infrarrojo cercano* (fNIRS) y la *tomografía por emisión de positrones* (PET) [Nam et al., 2018].

Las técnicas menos usadas en sistemas BCI son la MEG, fMRI y PET debido a sus complejos requerimientos técnicos, costos, y capacidades limitadas para tiempo real [Daly and Wolpaw, 2008], mientras que la EEG, ECoG y INR lideran en uso, siendo la EEG la más común debido a su compromiso entre resolución temporal, facilidad de uso, seguridad y costos [Ramadan and Vasilakos, 2017], véase la Figura 2.3.

Neuroimaging method	Activity Measured	Direct/Indirect Measurement	Temporal resolution	Spatial resolution	Risk	Portability
EEG	Electrical	Direct	~0.05 s	~10 mm	Non-invasive	Portable
MEG	Magnetic	Direct	~0.05 s	~5 mm	Non-invasive	Non-portable
ECoG	Electrical	Direct	~0.003 s	~1 mm	Invasive	Portable
Intracortical neuron recording	Electrical	Direct	~0.003 s	~0.5 mm (LFP) ~0.1mm (MUA) ~0.05mm (SUA)	Invasive	Portable
fMRI	Metabolic	Indirect	~1 s	~1 mm	Non-invasive	Non-portable
NIRS	Metabolic	Indirect	~1 s	~5 mm	Non-invasive	Portable

Figura 2.3: Características de métodos de neuroimagen.

Fuente: [Zadeh et al., 2017]

2.3.1. Electroencefalografía (EEG)

La EEG registra la actividad eléctrica del cerebro mediante electrodos de superficie colocados en el cuero cabelludo, en donde las señales EEG reflejan la suma de los potenciales post-sinápticos de muchos miles de neuronas que están orientadas radialmente al cuero cabelludo [Rao, 2013]. La baja resolución espacial de EEG (en el rango de centímetros cuadrados) es causada principalmente por las diferentes capas de tejido (líquido cefalorraquídeo, cráneo y el cuero cabelludo [Nunez and Srinivasan, 2006]), que se encuentran entre los electrodos y las fuentes de corriente en el cerebro, pues estas capas actúan como un *conductor de volumen* y un filtro pasa bajas que difumina la señal original [Rao, 2013].

Las señales EEG tienen una duración de 0.01 a 2s, amplitudes alrededor de $\pm 100 \mu V$ y componentes de frecuencia (denominadas *ondas cerebrales*) que se dividen en las siguientes bandas: δ (hasta 4Hz), θ (4-8Hz), α (8-12Hz), β (12-26Hz) y γ (26-100Hz) [Subasi, 2019], en donde cada onda tiene su propio conjunto de características, representando un nivel específico de actividad cerebral y estados mentales [Mühl et al., 2014].

Un sistema básico de EEG consiste principalmente de (1) al menos dos electrodos de registro (activo y referencia) y uno de tierra, (2) amplificador, (3) convertidor A/D y (4) un sistema de registro (normalmente una computadora) [Nicolas-Alonso and Gomez-Gil, 2012]. Los electrodos activo y de referencia funcionan como receptores de señal para comparaciones de diferencia de potencial, mientras que el electrodo de tierra sirve como la línea base (*baseline*) de las señales EEG, que ayuda a eliminar datos irrelevantes de los electrodos activo y de referencia [Nam et al., 2018].

En pruebas clínicas, los electrodos EEG se posicionan sobre el cuero cabelludo usando estándares internacionales, tal como los sistemas 10/20 o 10/10 (véase la Figura 2.4). Los electrodos EEG pueden ser *pasivos* (conectados a un amplificador) o *activos* (con preamplificador integrado y menos sensibles al ruido); ambos requieren el uso de gel o líquido salino para reducir la impedancia de contacto entre el cuero cabelludo y el electrodo (típicamente mantenida entre $1k\Omega$ y $10k\Omega$ [Nicolas-Alonso and Gomez-Gil, 2012]). Debido a esto, alternativas como los electrodos secos están siendo cada vez más usados en BCIs [Lotte et al., 2015].

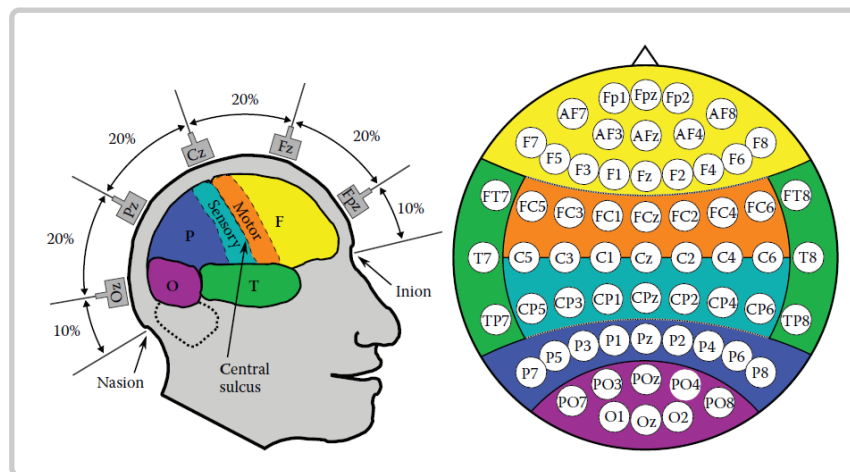


Figura 2.4: **Sistema internacional 10/20 (izquierda) y 10/10 (derecha) para registro EEG clínico.** En el sistema 10/20 los electrodos se localizan marcando los planos transversal y medio a intervalos de 10% y 20% entre dos puntos de referencia: *nasion* (parte superior de la nariz) e *inion* (parte posterior del cráneo). Las letras corresponden a las estructuras anatómicas que subyacen, lóbulo **F**rontal, **P**arietal, **O**ccipital, **T**emporal y surco **C**entral (**F**rontal polar, **A**nterior **F**rontal, etc), y los números indican el hemisferio cerebral, siendo pares para derecho e impares para izquierdo. **z** (*zero*) identifica la línea media (vértice craneal).

Fuente: [Nam et al., 2018]

2.3.2. Señales de control en BCIs

Para controlar un sistema BCI, existen diferentes tipos de señales de control, que se clasifican principalmente en: *i) exógenas/evocadas*, *ii) endógenas/espontáneas* y *iii) híbridas*, véase la Figura 2.5. Las señales exógenas son señales generadas *inconscientemente* por el usuario cuando recibe un *estímulo externo*, que puede ser visual, auditivo o táctil, por ejemplo SSEP y P300. Por otra parte, las señales endógenas son señales generadas *voluntariamente* por el usuario *sin estimulaciones externas*, por ejemplo los ritmos sensoriomotores (SMR), los potenciales corticales lentos (SCP) y las tareas cognitivas no motoras. Por último, las señales híbridas son aquellas que usan una combinación de señales cerebrales y su principal propósito es aumentar la confiabilidad y evitar las desventajas de cada tipo de señales [Ramadan and Vasilakos, 2017].

Para detectar la intención de movimiento en BCIs, dos de las señales de control más usadas provienen de SMR y SCP: ERD/ERS [Han Yuan and Bin He, 2014] y MRCP [Xu et al., 2013] respectivamente.

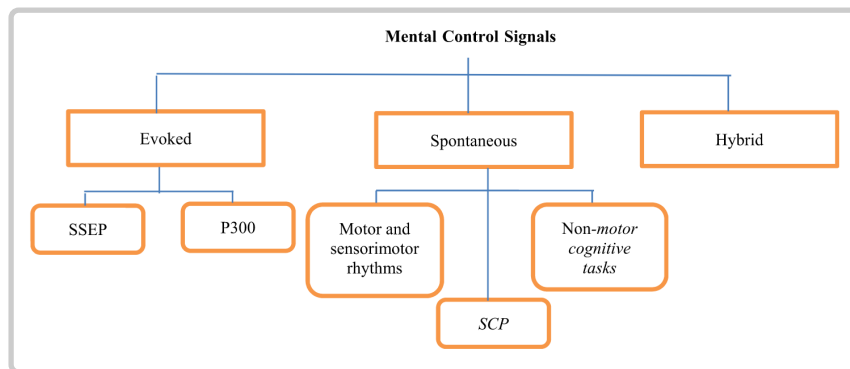


Figura 2.5: Principales señales de control en BCIs

Fuente: [Ramadan and Vasilakos, 2017]

Ritmos sensoriomotores (SMR)

Los *ritmos sensoriomotores* (SMR) son componentes de baja frecuencia, específicamente en las bandas μ (8-12 Hz) y β (12-26 Hz), que pueden modularse durante movimientos reales/ imaginados, ya sea en paradigmas basados en estímulos o a ritmo propio y pueden detectarse tanto en personas sanas como en personas con algún déficit motor [Han Yuan and Bin He, 2014]. Una *desincronización/sincronización relacionadas a evento* (ERD/ERS) se refiere al decremento (en el ritmo μ) e incremento (en el ritmo β) de la amplitud o potencia de SMRs y corresponde a una disminución/aumento de la sincronización de las poblaciones neuronales subyacentes, respectivamente [Pfurtscheller and Lopes da Silva, 1999]. Debido a que ambas ERD/ERS ocurren alrededor del inicio de movimiento, sobre la corteza sensoriomotora contralateral [Derambure et al., 1999], han sido un referente como señal de control para BMIs [Venkatakrisnan et al., 2014].

Potenciales corticales lentos (SCP)

El *potencial cortical lento* (SCP) es una deriva de voltaje negativo o positivo en la actividad eléctrica cortical que puede durar desde milisegundos hasta varios segundos [Birbaumer et al., 1990]. Una deriva negativa refleja un incremento en la actividad cerebral y típicamente ocurre durante la preparación cognitiva y conductual, mientras que una deriva positiva está relacionada con la disminución de excitabilidad cortical y es observada durante la inhibición del comportamiento [Studer et al., 2014]. Al igual que SMRs, los SCPs pueden generarse por movimientos reales/imaginados, con

sujetos/pacientes, y en paradigmas a ritmo propio/estimulados, lo cual los hace una potencial alternativa de señal de control BCI [Birbaumer et al., 1990].

El *potencial cortical relacionado a movimiento* (MRCP), un tipo de SCP, es un *potencial relacionado a evento* (respuesta promediada de muchos ejemplos) bloqueado al *onset* de movimiento [Shibasaki and Hallett, 2006]. Se caracteriza por una lenta deflexión negativa del componente cercano a DC (0.05-3 Hz) en la señal EEG, que aparece hasta 2s antes del *onset* de movimiento, alcanzando su pico de negatividad cerca del *onset* de movimiento, y seguido por un rebote positivo antes de que la señal regrese a su nivel de referencia [Jahanshahi and Hallett, 2003].

A pesar de que las señales ERD/ERS y MRCP provienen de distintos dominios (temporal en MRCP y frecuencial en ERD/ERS) la precisión de detección de intención de movimiento de ambas señales es comparable [Rodpongpun et al., 2020]. Sin embargo, investigaciones recientes han mostrado que MRCP (\sim -400 ms) [Lew, 2012][Lew et al., 2014] [Ren Xu et al., 2014] [Xu et al., 2016] es superior en términos de latencia de detección que ERD (\sim -600 ms) [Bai et al., 2011] [Wairagkar et al., 2018]. Adicionalmente, se ha mostrado que la combinación de características de ERD/ERS y MRCP mejora la detección del *onset* de movimiento en miembros superiores [Ibáñez et al., 2014], [López-Larraz et al., 2014]. Más adelante se presentan las principales características del MRCP objeto de estudio de este trabajo.

2.4. Preprocesamiento

El *preprocesamiento* (también conocido como *acondicionamiento de señal*) tiene como objetivo mejorar la calidad de la señal al mejorar la *relación señal/ruido* (SNR por sus siglas en inglés) [Grimann et al., 2010]. Para mejorar la SNR existen diferentes técnicas, la selección adecuada de éstas depende de múltiples factores, tales como la tecnología de registro, el número de electrodos y el neuro-mecanismo de la BCI [Bashashati et al., 2007]. Entre las principales técnicas de preprocesamiento para BCI basadas en EEG se encuentran: *filtrado temporal*, *submuestreo*, *normalización*, y *filtrado espacial* [Wolpaw and Wolpaw, 2012].

2.4.1. Filtrado temporal

El filtrado temporal o *filtrado de frecuencia* se refiere a la atenuación de los componentes de señal de una frecuencia (banda) particular. Es una técnica útil para atenuar ruido/artefactos, mientras se preserva la señal de interés [Widmann et al., 2015].

En general, el análisis EEG hace uso de *filtros digitales*. De acuerdo a su función algunos tipos de filtros son: *pasa-bajas* (atenúa bandas de altas frecuencias), *pasa-altas* (atenúa bandas de bajas frecuencias), *pasa-bandas* (atenúa bandas de bajas y altas frecuencias) y el *elimina-banda* (atenúa bandas de frecuencia específica). Algunos usos frecuentes son [Cohen, 2014]: filtro pasa-bajas para remover artefactos de alta frecuencia, tales como *electromiografía* (EMG); filtro pasa-altas con frecuencia de corte de 0.1 o 0.5 Hz para minimizar derivas lentas (*slow drifts* en inglés) que pueden distorsionar la visualización de la señal [Wolpaw and Wolpaw, 2012]; filtro elimina-banda (conocido como *filtro notch*) de 59-61 Hz que atenúa el ruido de la línea de alimentación de 60 Hz (por ejemplo en México y E.U.A.); filtro elimina-banda de 1-4 Hz para reducir artefactos de baja frecuencia, tales como *electrooculografía* (EOG) [Rao, 2013], entre otros. Es importante notar que el filtrado funciona únicamente cuando la señal cerebral de interés no cae en el rango de frecuencia de los artefactos [Rao, 2013]. En general, el ancho del filtro debe ser establecido de forma conservadora para evitar la pérdida innecesaria de información [Wolpaw and Wolpaw, 2012].

De acuerdo al tipo de retardo de la salida del filtro respecto de su entrada hay dos clases de filtros: de *fase lineal* que introduce un retraso igual en todas las frecuencias de banda y de *fase no-lineal* que genera diferentes retardos en diferentes bandas de frecuencia. El retraso de filtros de fase lineal se puede corregir cambiando la salida del filtro en el tiempo, lo que resulta en un *filtro de fase cero* que no tiene retraso. Debido a que los filtros de fase cero usan muestras precedentes (pasadas) y consecuentes (futuras) de la señal de entrada no filtrada; el *filtro* es clasificado como *no-causal*. Por otra parte, un *filtro causal* calcula la salida únicamente con muestras de entrada precedentes (pasadas). Es importante destacar que los filtros de fase cero (no causales) preservan latencias pico, mientras que los filtros causales necesariamente cambian la señal en el tiempo. Si es necesario un filtro causal, se debe considerar un filtro de *fase mínima* no lineal, ya que introduce solamente el mínimo posible retraso en cada frecuencia para una respuesta de magnitud dada. El retraso de fase cero también se puede conseguir con filtros no lineales al filtrar la salida del filtro por segunda vez en dirección

inversa (“filtrado de dos pasos”) para compensar el retraso del filtro (p. ej. la función `filtfilt()` de MATLAB/Octave), resultando en un filtro no-causal [Widmann et al., 2015] [Sani et al., 2016].

Finalmente de acuerdo al algoritmo del filtro digital, hay filtros de: *respuesta finita al impulso* (FIR) y *respuesta infinita al impulso* (IIR) o *filtros recursivos*. La salida de los filtros FIR se calcula solamente con los valores de *entrada* actual y previos, mientras que la de un filtro recursivo, además de los valores de entrada, también utiliza valores de *salida anteriores* [Kropotov, 2009]. Los filtros FIR introducen un retraso largo (varios segundos), haciendo a los filtros IIR (p. ej. *Butterworth* o *Chevyshev*), la única opción para una aplicación BCI en línea [Bibian et al., 2017]. A diferencia de los otros filtros IIR, los filtros Butterworth presentan una respuesta de frecuencia máxima plana, permitiendo una adquisición de la señal precisa en todo el ancho de banda especificado y una respuesta de fase aproximadamente lineal en la banda de paso del filtro [Bibian et al., 2017], lo cual implica que la forma en el dominio del tiempo de una señal con componentes espectrales dentro de la banda de paso del filtro no se modifica mediante el filtrado [Widmann et al., 2015]. De esta manera, en el presente trabajo usamos el filtro Butterworth en el preprocesamiento de la señal MRCP.

2.4.2. Submuestreo y normalización

Cuando la señal EEG se digitaliza a una tasa mayor a la tasa de Nyquist, el submuestreo puede ser útil para un eficiente procesamiento y almacenamiento. La normalización se emplea para convertir diferentes señales a la misma escala permitiendo simplificar potencialmente el análisis y la interpretación de señales, así como los pasos de procesamiento subsecuentes. Sin embargo, debe usarse con cuidado, ya que elimina las diferencias de amplitud potencialmente útiles dentro de un conjunto de señales [Wolpaw and Wolpaw, 2012].

2.4.3. Filtrado espacial

El filtrado espacial mejora la sensibilidad para una actividad de interés y/o suprime actividad indeseada [de Cheveigné and Simon, 2008]. El objetivo de los *filtros espaciales* es combinar señales de sensores a fin de obtener *canales virtuales* (*canales sustitutos*, *componentes*) con un mayor SNR [Wolpaw and Wolpaw, 2012]. Las señales EEG filtradas espacialmente pueden considerarse como una combinación lineal de los canales

EEG monopolares originales, es decir:

$$\mathbf{y}(t) = \mathbf{W}\mathbf{x}(t), \quad y_i(t) = \sum_{j=1}^N w_{ij}x_j(t) \quad (2.1)$$

donde $\mathbf{x}(t) \in \mathbb{R}^{N \times 1}$ representa las señales EEG monopolares originales con N canales en el instante t , $\mathbf{y}(t) \in \mathbb{R}^{M \times 1}$ las M señales filtradas, $\mathbf{W} \in \mathbb{R}^{M \times N}$ la matriz de filtrado y w_{ij} los pesos del filtro i . Los algoritmos de análisis de componentes normalmente asumen $M = N$. Los métodos de filtrado espacial difieren en la forma de calcular los pesos bajo un cierto criterio [Graumann et al., 2010]. Usualmente los coeficientes del filtro permanecen constantes sobre el conjunto de datos, es decir, *invariante en el tiempo* [de Cheveigné and Arzounian, 2018]. Los pesos pueden ser *independientes de los datos* (por ejemplo CAR, bipolar o laplaciano de superficie), o *dependientes de los datos* (p. ej. PCA, ICA, CSP).

Los filtros espaciales independientes de datos (también conocidos como *métodos de referencia* [Nam et al., 2018]) usan normalmente relaciones geométricas para determinar los pesos del filtro espacial. Estos filtros tienen características globales (toman información de todos los canales) o locales (usan solo ciertos canales). Normalmente es difícil conocer las características exactas de una actividad cerebral relevante [Wolpaw and Wolpaw, 2012]. Una alternativa a esto pueden ser los filtros espaciales que se derivan de los datos del usuario BCI. A continuación, se describen algunos de los filtros espaciales más comunes.

Referencia bipolar

La *referencia bipolar* calcula la primera derivada espacial y enfatiza la diferencia en el gradiente de voltaje en una dirección particular actuando como filtro pasa-altas espacial (es decir, acentúa los componentes del EEG con distribuciones espaciales altamente enfocadas) [He et al., 2013]. Una ventaja de la referencia bipolar es que es fácil remover artefactos que ocurren al mismo tiempo en la misma posición, sin embargo, su desventaja es que la medición de una posición específica no es posible debido a que la información del electrodo podría cancelarse cuando las actividades cerebrales en dos posiciones son temporal y espacialmente similares entre sí [Nam et al., 2018]. La

referencia bipolar se calcula como sigue:

$$V_{i,j}^{bipolar}(t) = V_i^{CR}(t) - V_j^{CR}(t) \quad (2.2)$$

donde $V_{i,j}^{bipolar}(t)$ es la diferencia de potencial entre el electrodo j e i , colocados anteriormente y posteriormente (o de izquierda y derecha) de la posición objetivo.

Laplaciano de superficie (LAP)

El *laplaciano de superficie* (LAP) [McFarland et al., 1997] (originalmente conocido como *Referencia Promedio Local* (LAR) [Pfurtscheller et al., 1994]) calcula para cada electrodo la segunda derivada de la distribución de voltaje espacial instantáneo y, por lo tanto, enfatiza actividad que se origina en fuentes radiales inmediatamente debajo del electrodo [Lu et al., 2013] actuando como filtro pasa-altas espacial. En la práctica, una forma típica de aproximar el LAP es usando el método de diferencias finitas, en donde se asume que (1) la superficie del cuero cabelludo es localmente aproximadamente plana y que (2) los electrodos son equidistantes, está demostrado que para la configuración ‘más (+)’ alrededor de un electrodo i [Carvalhoes and de Barros, 2014]:

$$V_i^{LAP}(t) = V_i^{CR}(t) - \frac{1}{K} \sum_{j=1}^K V_j^{CR}(t) \quad (2.3)$$

donde $V_i^{CR}(t)$ es el potencial entre el electrodo i y el electrodo de referencia y $K = 4$ el número de electrodos alrededor del electrodo i (anterior, posterior, izquierda y derecha). Hay dos tipos de filtros laplacianos: el *laplaciano pequeño*, el cual resta el promedio de los cuatro vecinos más cercanos y el *laplaciano grande*, que resta el promedio de los cuatro vecinos siguientes a los más cercanos [Grimann et al., 2010].

Referencia promedio común (CAR)

El filtro *Referencia Promedio Común* (CAR) es otro filtro pasa-altas espacial. Se implementa al substrair el valor promedio de todos los canales del montaje del canal de interés, permitiendo reducir el ruido global [Bertrand et al., 1985]. Sin embargo, CAR no es efectivo en reducir el ruido que no es común en todos los electrodos y que se presenta frecuentemente, tal como la *electrooculografía* (EOG) y la *electromiografía*

(EMG), entre otros [Graimann et al., 2010]. El filtro CAR se calcula como:

$$V_i^{CAR}(t) = V_i^{CR}(t) - \frac{1}{N} \sum_{j=1}^N V_j^{CR}(t) \quad (2.4)$$

donde $V_i^{CR}(t)$ es el potencial entre el electrodo i y el electrodo de referencia y N el número de electrodos en el montaje.

En la Figura 2.6 se muestran algunas representaciones gráficas de los filtros mencionados previamente.

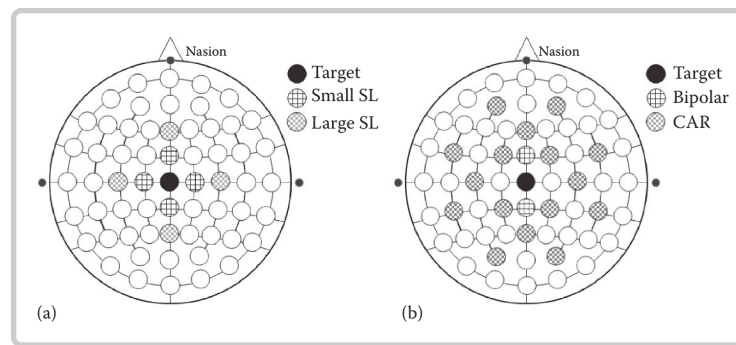


Figura 2.6: **Filtros espaciales independientes de datos.** (a) Laplaciano de superficie pequeño y grande (b) Bipolar y CAR

Fuente: [Nam et al., 2018]

Análisis de Componentes Principales (PCA)

En general, el *Análisis de Componentes Principales* (PCA) se conoce como una técnica para *reducción de dimensionalidad*, esto es, para reducir el número de variables en un conjunto de datos con la menor pérdida de información. Así, el objetivo del PCA es construir conjunto de pesos (denominados *componentes principales* (PC)) para transformar N variables *correlacionadas* en un nuevo conjunto de $M \leq N$ variables *no-correlacionadas*, que describen la mayor parte de la varianza entre-variables de entrada [Cohen, 2014].

Para el caso de análisis EEG, sea $\mathbf{X} \in \mathbb{R}^{N \times P}$ la matriz de señales EEG *canal-por-muestras de tiempo* (o más generalmente matriz *variables-por-instancias*), entonces los *cursos de tiempo* de los PC del espacio de entrada estandarizado pueden obtenerse

mediante [Graumann et al., 2010]:

$$\mathbf{Y} = \mathbf{W}^\top (\mathbf{X} - \bar{\mathbf{X}}) \quad (2.5)$$

donde $\bar{\mathbf{X}} \in \mathbb{R}^{N \times P}$ es una matriz apropiadamente dimensionada que contiene el promedio de cada canal a lo largo del tiempo, y $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_N] \in \mathbb{R}^{N \times N}$ es la matriz de transformación PCA, que puede obtenerse mediante una *descomposición de valores propios* de la *matriz de covarianza* [Cohen, 2014]:

$$\text{Cov}(\mathbf{X}) = \frac{1}{P-1} (\mathbf{X} - \bar{\mathbf{X}})(\mathbf{X} - \bar{\mathbf{X}})^\top \quad (2.6)$$

donde $\mathbf{w}_1, \dots, \mathbf{w}_N$ son los vectores propios de la matriz de covarianza, conocidos como los PC, que corresponden a los N diferentes valores propios $\lambda_1, \dots, \lambda_N$ en orden descendente, mientras que cada fila de \mathbf{W} almacena los pesos (también llamados *cargas de componentes*) de cada electrodo [Cohen, 2014]. Nótese que cada PC puede ser trazado como un *mapa topográfico* y que tiene asociado un curso de tiempo, y ese curso de tiempo es una suma ponderada de la actividad de todos los electrodos. Finalmente, para extraer aquellos PC que representan la mayoría de la varianza entre-canales de la señal de entrada, se seleccionan M vectores propios correspondientes a los M valores propios ‘más grandes’ [Cohen, 2014]. Existen varios métodos para determinar M , tales como el *gráfico de sedimentación*, retención de una proporción fija de varianza (usualmente 70-90%), retención por valores propios mayores a la unidad (*regla de Kaiser*), así como métodos estadísticos, por ejemplo la *prueba de Bartlett* o el remuestreo *bootstrap* para valores propios [Costa et al., 2014].

Es importante notar que a pesar de que PCA decorrelaciona y reduce la dimensionalidad de señales multivariadas, el vector resultante \mathbf{Y} no implica independencia estadística de orden superior (más allá de la varianza) tal como el sesgo y la curtosis [Cohen, 2014]. De esta manera, los PC no son usualmente variables significativas, pues aún son mezclas de las fuentes subyacentes [Semmlow and Griffel, 2014]. Esta debilidad de independencia estadística a través de la decorrelación nos motiva a usar métodos de separación de fuentes tales como ICA.

Análisis de Componentes Independientes (ICA)

Uno de los métodos más populares para detección/remoción de artefactos en EEG está basado en la *separación de fuentes ciegas* (BSS) [Islam et al., 2016]. Teniendo una *matriz de observación* EEG $\mathbf{X} \in \mathbb{R}^{N \times P}$ con N canales y P muestras, el modelo lineal BSS para el mismo número de fuentes y sensores (es decir, observaciones) se formula como:

$$\mathbf{X} = \mathbf{A}\mathbf{S} \quad (2.7)$$

donde $\mathbf{A} \in \mathbb{R}^{N \times N}$ es la *matriz de mezcla* y $\mathbf{S} \in \mathbb{R}^{N \times P}$ la *matriz de fuentes*. La justificación para la descripción de esta técnica como *ciega* es que la información sobre la matriz de mezcla y las fuentes son desconocidas [Cohen, 2014]. De esta manera, el objetivo de diferentes algoritmos de BSS es calcular la *matriz de desmezcla* $\mathbf{W} \in \mathbb{R}^{N \times N}$ (idealmente la inversa de \mathbf{A}) para determinar un estimado de las fuentes originales y artefactos (que no son necesariamente independientes), $\mathbf{Y} \in \mathbb{R}^{N \times P}$, de solamente las observaciones \mathbf{X} , i.e.:

$$\mathbf{Y} = \mathbf{W}\mathbf{X} \quad (2.8)$$

donde cada fila de \mathbf{W} es un *filtro espacial* para estimar un componente. Una vez obtenida \mathbf{W} con el algoritmo BSS se puede reconstruir la señal EEG mediante:

$$\tilde{\mathbf{X}} = \mathbf{W}^{-1}\mathbf{Y} \quad (2.9)$$

donde $\tilde{\mathbf{X}}$ es la señal EEG reconstruida. Cada columna de \mathbf{W}^{-1} es un *patrón espacial* que refleja la distribución espacial de un componente en cada uno de los sensores del cuero cabelludo y en donde, mediante una interpolación, se puede generar el conocido *mapa de cuero cabelludo* (*scalp map*, en inglés) asociado con cada componente [Makeig et al., 1999].

El problema BSS puede resolverse usando varias suposiciones de separación. El *análisis de componentes independientes* (ICA) es un algoritmo no-supervisado de BSS en donde las condiciones necesarias para separabilidad de fuente son [Haykin, 2010]: *i*) las *fuentes* son *estadísticamente independientes* (conocidas como *componentes independientes*), *ii*) los componentes independientes tienen una *distribución no gaussiana* (en contraste con PCA que supone que las fuentes tienen una distribución gaussiana) y *iii*) la matriz de revoltura es invertible.

Existen dos principales familias de algoritmos ICA [Haykin, 2010], mientras algunos

se basan en la *minimización de la información mutua* entre las fuentes estimadas (por ejemplo *infomax* [Bell and Sejnowski, 1995], *extended-infomax* [Lee et al., 1999], etc.), otros se basan en la *maximización de la No-Gaussianidad* (por ejemplo *fastICA* [Hyvärinen and Oja, 1997]). En este trabajo, usamos *extended-infomax ICA* debido a sus aplicaciones exitosas en varios campos, particularmente en procesamiento de señales EEG [Brunner et al., 2007].

La corrección de artefactos en datos EEG se puede llevar a cabo de distintas maneras ya sea [Barthélemy et al., 2017]: *i*) manual, *ii*) fuera de línea, *iii*) en línea a bloques o *iv*) en línea. En general, hay tres pasos para la remoción de artefactos/ruido mediante técnicas BSS: **1**) hallar los estimados de fuentes con el algoritmo BSS, **2**) identificar y descartar los componentes que representan artefactos, manteniendo las fuentes donde la actividad cerebral domina y **3**) reconstruir el espacio de sensores [Zhang et al., 2015]. Algunas formas sencillas para identificar componentes que representan artefactos son: *i*) inspección visual de los cursos de tiempo y patrones espaciales de los componentes o *ii*) fuera de línea mediante un algoritmo que usa información de \mathbf{W}^{-1} y \mathbf{W} , como en [Zhou et al., 2014]. Finalmente, una vez identificado los componentes asociados a artefactos es posible obtener la señal EEG limpia, al establecer a cero el filtro espacial respectivo o equivalentemente mediante [Mantini et al., 2007]:

$$\widetilde{\mathbf{X}}_{clean} = \mathbf{X} - \mathbf{W}^{-1}\mathbf{Z}\mathbf{Y} \quad (2.10)$$

donde $\mathbf{Z} \in \mathbb{R}^{N \times N}$ es una matriz diagonal donde z_{ii} es igual a 1 si el i -ésimo componente fue un artefacto y 0 en caso contrario.

2.5. Extracción de características

La extracción de características tiene como objetivo representar las señales de EEG crudas mediante un número idealmente pequeño de valores relevantes llamados *características*, las cuales capturan la información relevante para describir los estados mentales a identificar [Lotte, 2014]. Las características de las BCIs basadas en EEG normalmente son ya sea *temporales*, *frecuenciales*, *espaciales*, o combinaciones de éstas. Entre las características más frecuentes se encuentran [Lotte, 2008]: *i*) *temporales*: amplitud de las señales EEG crudas, parámetros auto-regresivos y parámetros de Hjorth,

ii) frecuenciales: potencia de banda y la densidad espectral de potencia (también conocida como espectro), *iii) tiempo-frecuencia*: transformada de Fourier de tiempo reducido y transformada Wavelet, *iv) espacio-temporales*: filtro laplaciano y patrones espaciales comunes (CSP).

Adicionalmente, puede aplicarse un paso de selección de características después de la extracción de características a fin de seleccionar el subconjunto de características más informativas para la clasificación [Lotte et al., 2018]. Esto es especialmente útil para diseños de BCI con datos de entrada de alta dimensión, ya que reduce la dimensión del espacio de características. Hay 3 principales tipos de técnicas de selección de características [Adair, 2018]: métodos de *filtro*, *envoltura* e *integrados*. Los métodos de filtro seleccionan características independientemente del clasificador y se dividen en dos clases: *univariados* y *multivariados*. Las medidas de rendimiento asociadas a métodos de filtro pueden ser de: *similaridad*, *consistencia*, *distancia*, *información* o *estadísticas* [Jović et al., 2015]. Los métodos univariados evalúan la discriminabilidad de cada característica independientemente y entonces se seleccionan las K mejores características (K es seleccionado por el diseñador de la BCI), su ventaja es su rapidez computacional, sin embargo, ignoran la redundancia entre características [Lotte, 2014]. Los métodos multivariados evalúan subconjuntos de características y mantienen el mejor subconjunto con K características, su ventaja es que pueden lidiar con la redundancia pero son computacionalmente más costosos [Lotte, 2014]. Los métodos de envoltura seleccionan las mejores características en base al rendimiento de un clasificador, p. ej. la *búsqueda secuencial hacia adelante/atrás* (SFS/SBS) y los *algoritmos evolutivos* (EAs). Por último, los métodos integrados combinan la selección de características con el clasificador en un proceso único, p. ej. el *operador de mínima contracción y selección absoluta* (LASSO), la *eliminación de características recursiva* (RFE) y los *árboles de decisión*.

2.6. Clasificación

Clasificación es el problema de asignar una de M etiquetas a una nueva señal de entrada, dado datos los de entrenamiento etiquetados y sus correspondientes etiquetas de salida. Hay varias categorizaciones de clasificadores en BCIs, sin embargo se pueden resaltar principalmente los *métodos lineales* y *no lineales*. Los clasificadores lineales usan una función lineal para aproximar la frontera entre clases (por ejemplo LDA, SVM lineal, regresión logística, etc.), mientras que los no lineales (k -vecinos cercanos

(kNN), cuantización del vector de aprendizaje (LVQ), SVM con kernel, modelos de Markov ocultos (HMM), etc.) intentan aprender una frontera de decisión no lineal [Oweiss, 2010]. En este trabajo se hace uso de clasificadores lineales.

Se considera el problema de clasificación de dos clases. La frontera de decisión está dada por la ecuación del hiperplano [Rao, 2013]:

$$g(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + w_0 = 0 \quad (2.11)$$

La frontera entre las dos clases es así caracterizada por el vector normal al hiperplano, \mathbf{w} y el umbral w_0 , que se determinan a partir de los datos de entrenamiento etiquetados, i.e. $\mathbf{x}_i \in \mathbb{R}^p, i \in [1, \dots, N]$ y $y_i \in \{-1, 1\}$, donde $N = N_1 + N_2$ es el número total de ejemplos de entrenamiento, N_1 para $y_i = 1$ y N_2 para $y_i = -1$.

Dado un nuevo vector de entrada $\mathbf{x} \in \mathbb{R}^p$ la clasificación se logra al calcular el signo de la distancia perpendicular del vector de características al hiperplano, dada por $d(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + w_0$ (suponiendo $\|\mathbf{w}\| = 1$) [Rao, 2013], esto es:

$$y = \begin{cases} +1 & \text{si } \text{sign}(\mathbf{w}^\top \mathbf{x} + w_0) \text{ es positivo (o cero)} \\ -1 & \text{si } \text{sign}(\mathbf{w}^\top \mathbf{x} + w_0) \text{ es negativo} \end{cases} \quad (2.12)$$

Durante experimentos con una BCI en línea algunas veces, la distancia (con signo) $d(\mathbf{x})$, también se usa para proveer realimentación al usuario acerca de qué tan cerca de la frontera está el nuevo vector de características [Rao, 2013].

2.6.1. Análisis Discriminante Lineal (LDA)

El *Análisis Discriminante Lineal* (LDA; también conocido como *Discriminante Lineal de Fisher* (FLD)) es una técnica de clasificación lineal en términos de la reducción de dimensiones [Bishop, 2006]. Considérese el caso de dos clases y supóngase un vector de entrada $\mathbf{x} \in \mathbb{R}^p$ y proyectémoslo a dimensión uno usando:

$$y = \mathbf{w}^\top \mathbf{x}. \quad (2.13)$$

Si se coloca un umbral sobre y y se clasifica $y \geq -w_0$ como clase \mathcal{C}_1 y de otro modo como clase \mathcal{C}_2 , entonces obtenemos el clasificador lineal estándar de la Ecuación 2.11 [Bishop, 2006]. De esta manera, la idea básica del LDA es encontrar la mejor dirección

de proyección discriminante, \mathbf{w} , para maximizar la varianza *entre las clases* y minimizar la varianza *dentro de clases*, véase la [Figura 2.7](#). Para calcular \mathbf{w} , el LDA supone que las distribuciones condicionales de clase $P(\mathbf{x}|\mathcal{C}_1)$ y $P(\mathbf{x}|\mathcal{C}_2)$ son *distribuciones normales* con vector promedio $\boldsymbol{\mu}_i$ y *matrices de covarianza iguales* $\boldsymbol{\Sigma}_i, i \in \{1, 2\}$ [[Rao, 2013](#)] [[Nam et al., 2018](#)]. Se puede mostrar que el vector de proyección \mathbf{w} está definido por [[Blankertz et al., 2011](#)]:

$$\mathbf{w} = \boldsymbol{\Sigma}_c^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) \quad (2.14)$$

donde $\boldsymbol{\Sigma}_c = \frac{1}{2}(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)$ es la matriz de covarianza común estimada, es decir, el promedio de las matrices de covarianzas empíricas $\boldsymbol{\Sigma}_i$. Intuitivamente, la dirección que maximiza la separación entre las clases puede verse como una transformación lineal (por $\boldsymbol{\Sigma}_c^{-1}$) del vector que une los promedios de las dos clases $(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$ [[Zaki and Meira Jr, 2019](#)]. El vector \mathbf{w} es conocido como FLD, aunque estrictamente no es un discriminante, sino más bien una elección específica de la dirección de proyección de los datos a una dimensión. Sin embargo, los datos proyectados pueden subsecuentemente usarse para construir un discriminante, al seleccionar un umbral y_0 , a fin de clasificar un nuevo punto como perteneciente a \mathcal{C}_1 si $y(\mathbf{x}) \geq -y_0$ y clasificarlo como perteneciente a \mathcal{C}_2 de otro modo [[Bishop, 2006](#)]. El umbral y_0 se define frecuentemente a la mitad de la proyección de los promedios de las dos clases [[Rao, 2013](#)] [[Graumann et al., 2010](#)], i.e.:

$$y_0 = -\mathbf{w}^\top(\boldsymbol{\mu}_2 + \boldsymbol{\mu}_1)/2 \quad (2.15)$$

En general, se ha encontrado que el LDA produce buenos resultados en investigación de BCIs debido a su fácil implementación y computo rápido para uso en línea [[Rao, 2013](#)]. La principal desventaja del LDA es su linealidad, la cual supone *datos linealmente separables*, lo cual normalmente no ocurre en experimentos EEG reales. Sin embargo, la simplicidad de un método lineal puede ser también una ventaja, ya que la frontera de decisión no será sobreajustada a las muestras de entrenamiento. Otra consideración es que cuando se construyen clasificadores no lineales como perceptrones multicapa o máquinas de soporte vectorial, se tienen que seleccionar diferentes parámetros libres cuidadosamente para que la clasificación sea eficiente [[Bashashati et al., 2007](#)]. Tomando en cuenta estas consideraciones, el LDA puede ser más robusto y producir mejores resultados de clasificación que al usar métodos no lineales [[Bashashati et al., 2007](#)]. Las técnicas de *regularización* se usan típicamente para promover la generalización y evitar el sobreajuste, especialmente cuando el número de características es grande y

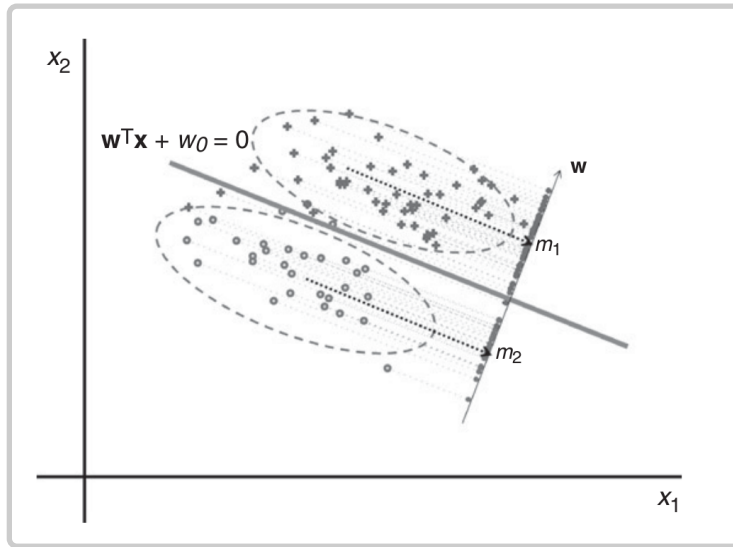


Figura 2.7: **Análisis Discriminante Lineal (LDA)**. Las cruces representan la clase 1 mientras que los círculos representan clase 2. Las proyecciones de estos puntos de datos sobre el vector w se muestran como cruces y círculos más pequeños.

Fuente: [Rao, 2013]

el número de observaciones disponibles es pequeño, conocido como *maldición de la dimensionalidad* [Rao, 2013]. Por ejemplo, en el caso del LDA se podrían tener datos insuficientes para estimar con precisión el promedio de clase u_i y la covarianza de clase Σ_i . En particular, Σ_c podría llegar a ser singular [Rao, 2013].

El Análisis Discriminante Lineal Regularizado (RLDA; también conocido como LDA regularizado por covarianza o LDA con contracción) es una variante del LDA que usa la *contracción* para compensar el sesgo sistemático de las matrices de covarianza estimadas, en donde Σ_i se reemplaza por [Blankertz et al., 2011]:

$$\tilde{\Sigma}_i = (1 - \gamma)\Sigma_i + \gamma\nu\mathbf{I} \quad (2.16)$$

donde \mathbf{I} es la matriz identidad, $\gamma \in [0, 1]$ es el *parámetro de regularización* que se puede sintonizar vía técnicas de selección de modelo (por ejemplo validación cruzada) [Rao, 2013] o métodos analíticos [Ledoit and Wolf, 2000] y ν es un *parámetro de escalado* definido como el valor propio promedio ($\text{traza}(\Sigma_i)/p$ de Σ_i en donde p es la dimensionalidad del espacio de características [Blankertz et al., 2011]).

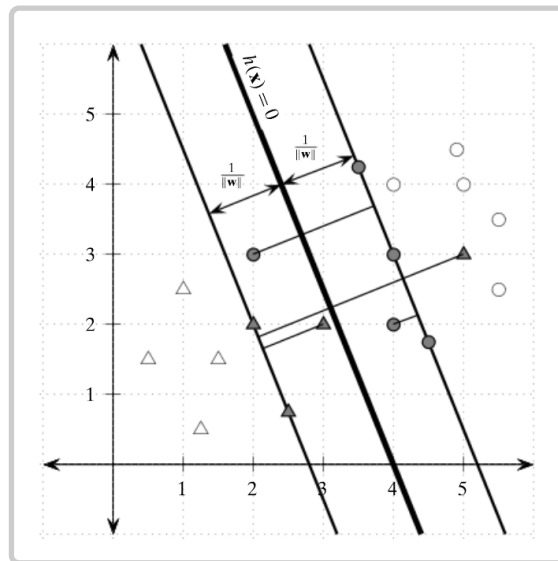


Figura 2.8: **SVM de margen suave.** La *SVM de margen suave* encuentra el hiperplano de margen $\frac{1}{\|w\|}$ máximo que separa los datos de entrenamiento con un número mínimo de clasificaciones erróneas. Los vectores de soporte se indican con puntos sombreados, mientras que puntos con valores flojos positivos (i.e., $\xi_i > 0$) con línea negra delgada.

Fuente: [Zaki and Meira Jr, 2019]

2.6.2. Máquina de Soporte Vectorial (SVM)

La *Máquina de Soporte Vectorial* (SVM) es un clasificador que encuentra el hiperplano de separación que maximiza el *margen* (es decir, la distancia mínima de un punto de datos desde el hiperplano de separación), $\frac{1}{\|w\|}$, entre las muestras de las clases. Todos los puntos (vectores) que logran esta distancia mínima son llamados *vectores de soporte* para el hiperplano [Bishop, 2006]. La solución estándar de SVM se deriva del siguiente problema de optimización [Cortes and Vapnik, 1995].

Sea $\mathbf{D} = (\mathbf{x}_i, y_i), i = 1, \dots, N$ el conjunto de entrenamiento en el *espacio de entrada*, donde $\mathbf{x}_i \in \mathbb{R}^p$ y $y_i \in \{+1, -1\}$. Aplicando $\Phi : \mathbb{R}^p \rightarrow \mathcal{H}$ a cada punto (generalmente $\dim(\mathcal{H}) \geq \dim(\mathbb{R}^p)$), obtenemos un nuevo conjunto de datos en el *espacio de características*, dado por $\mathbf{D}_\Phi = (\Phi(\mathbf{x}_i), y_i), i = 1, \dots, N$. La función de costo de la SVM en el

espacio de características (\mathbf{D}_Φ) es [Zaki and Meira Jr, 2019]:

$$\begin{aligned} & \min_{\mathbf{w}, b, \xi_i} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N (\xi_i)^k \right\} \\ \text{sujeto a } & y_i(\mathbf{w}^\top \Phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \quad \forall \mathbf{x}_i \in \mathbf{D} \end{aligned} \quad (2.17)$$

donde \mathbf{w} es el vector de pesos, b es el umbral y ξ_i las variables flojas, todas en el espacio de características.

Note que en lugar de maximizar el margen $\frac{1}{\|\mathbf{w}\|}$ se puede minimizar $\|\mathbf{w}\|$, que es equivalente a minimizar $\|\mathbf{w}\|^2$. Añadiendo el factor $\frac{1}{2}$ por conveniencia, obtenemos el primer factor de la función de costo [Bishop, 2006] [Zaki and Meira Jr, 2019]. En el caso de datos linealmente no separables, también llamado caso de *margen suave*, se añade el término $\sum_{i=1}^N (\xi_i)^k$, que representa la *perdida*, es decir, un estimado de la desviación del caso separable. Las *variables flojas* $\xi_i \geq 0$ miden la desviación de cada punto de entrenamiento desde de la frontera de margen correcta, mientras que el *parámetro de regularización* $C \geq 0$, seleccionado normalmente vía selección de modelo, controla el compromiso entre *maximizar el margen* (es decir, minimizar $\|\mathbf{w}\|^2/2$) o *minimizar la perdida* (es decir, minimizar $\sum_{i=1}^N (\xi_i)^k$). La constante $k > 0$ gobierna la forma de la pérdida. Típicamente k se fija en 1 (*pérdida de bisagra*) o en 2 (*pérdida cuadrática*) [Zaki and Meira Jr, 2019].

En la práctica, el problema de optimización de la Ecuación 2.17 se resuelve típicamente usando su *forma dual Wolfe* [Kowalczyk, 2017]:

$$\begin{aligned} & \max_{\alpha} \left\{ \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \right\} \\ \text{sujeto a } & 0 \leq \alpha_i \leq C, i = 1, \dots, N \\ & \sum_{i=1}^N \alpha_i y_i = 0 \end{aligned} \quad (2.18)$$

donde α_i son los *multiplicadores de Lagrange* y $K : \mathbb{R}^p \rightarrow \mathbb{R}$ definido por $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle_{\mathcal{H}}$ es una *función* conocida como *kernel* ($\langle \cdot, \cdot \rangle_{\mathcal{H}}$ denota un producto punto en \mathcal{H}), que devuelve el resultado de un producto punto realizado en otro espacio, es decir, en \mathcal{H} , donde los datos pueden ser linealmente separables. Para el caso lineal

$\Phi(\mathbf{x}) = \mathbf{x}$, el kernel es $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$, véase la [Figura 2.8](#). Esto significa que para resolver el caso lineal, no se necesita como tal el dato de entrenamiento \mathbf{x}_i , sino únicamente el producto punto $\mathbf{x}_i \cdot \mathbf{x}_j$ entre dos datos de entrenamiento. De esta manera, aplicar el *truco kernel* simplemente significa remplazar el producto punto de dos observaciones en \mathcal{H} por una función kernel [[Kowalczyk, 2017](#)]. Ahora, se tiene la capacidad de cambiar la función kernel a fin de clasificar datos no separables [[Kowalczyk, 2017](#)]. Uno de los kernel más usados en BCIs es el de *función de base radial* (RBF) o Gaussiano, que devuelve el resultado de un producto punto realizado en \mathbb{R}^∞ , definido como [[Kowalczyk, 2017](#)]:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2\right) \quad (2.19)$$

donde $\gamma > 0$ establece el ancho de la curva de campana. Cuando γ es muy pequeña el modelo se comporta como un SVM lineal, mientras que cuando es muy grande el modelo es fuertemente influenciado por cada vector soporte [[Kowalczyk, 2017](#)]. Al lector interesado en las expresiones de \mathbf{w} y b , a partir de los multiplicadores de Lagrange se le sugiere revisar [[Zaki and Meira Jr, 2019](#)].

2.6.3. Regresión Logística

Dado un conjunto de atributos predictores o variables independientes $\tilde{\mathbf{X}} = (X_0, X_1, \dots, X_p)^\top$ y una respuesta *categorica* o variable dependiente Y , el objetivo de la *regresión logística* es predecir la probabilidad de las variables de respuesta basados en las variables independientes. De hecho, la regresión logística es una técnica de clasificación que, dado un punto $\mathbf{x}_j \in \mathbb{R}^{p+1}$, predice $P(c_i|\mathbf{x}_j)$ para cada clase c_i en el dominio de Y (el conjunto posible de clases) [[Zaki and Meira Jr, 2019](#)].

Para el caso de clasificación binaria sea $\mathbf{D} = (\mathbf{x}_i, y_i), i = 1, \dots, N$ el conjunto de entrenamiento en el *espacio de entrada*, donde $\mathbf{x}_i \in \mathbb{R}^p$ y $y_i \in \{+1, -1\}$. Definiendo $\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}} = \mathbf{w}^\top \mathbf{x} + b$ donde $\tilde{\mathbf{x}}_i = [1, x_1, x_2, \dots, x_p]^\top \in \mathbb{R}^{p+1}$ y $\tilde{\mathbf{w}} = [b, w_1, w_2, \dots, w_p]^\top \in \mathbb{R}^{p+1}$, entonces el conjunto de datos de entrenamiento *augmentado* es $\tilde{\mathbf{D}} = (\tilde{\mathbf{x}}_i, y_i), i = 1, \dots, N$. Así, la probabilidad de que la respuesta sea $Y = +1$ o bien $Y = -1$, usando la función logística $g(z)$ con entrada $\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}$, es [[Zaki and Meira Jr, 2019](#)]:

$$\begin{aligned} P(Y = +1|\tilde{\mathbf{X}} = \tilde{\mathbf{x}}) &= g(\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}); & g(z) &= \frac{1}{1 + \exp(-z)} \\ P(Y = -1|\tilde{\mathbf{X}} = \tilde{\mathbf{x}}) &= 1 - P(Y = 1|\tilde{\mathbf{X}} = \tilde{\mathbf{x}}) & &= g(-\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}) \end{aligned} \quad (2.20)$$

respectivamente.

Para aprender el vector de peso $\tilde{\mathbf{w}}$ se usa el método de *máxima verosimilitud* (MLE por sus siglas en inglés), el cual usa el método de *gradiente ascendente* para computar el valor óptimo [Zaki and Meira Jr, 2019].

Una vez que el modelo ha sido entrenado, se puede predecir la respuesta para cualquier punto de evaluación aumentado $\tilde{\mathbf{x}}$ mediante:

$$y = \begin{cases} +1 & \text{si } g(\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}) \geq 0.5 \\ -1 & \text{si } g(\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}) < 0.5 \end{cases} \quad (2.21)$$

2.7. Desempeño de BCI

2.7.1. Medida de evaluación de clasificador

		Predicción	
		Positivos	Negativos
Observación	Positivos	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Figura 2.9: Matriz de confusión de un problema de clasificación binaria

Para la evaluación del clasificador se pueden usar diferentes métricas, todas ellas derivadas de la *matriz de confusión*. En el caso de clasificación binaria, la matriz de confusión contiene el número de *verdaderos positivos* (VP), *verdaderos negativos* (VN), *falsos positivos* (FP), *falsos negativos* (FN) como se muestra en la [Figura 2.9](#).

Para esta investigación se usó la métrica de *precisión* (*Accuracy* en inglés), la cual indica la fracción de predicciones correctas sobre el *conjunto de evaluación*:

$$Acc = \frac{VP + VN}{VP + VN + FP + FN} \quad (2.22)$$

La precisión proporciona un estimado de la probabilidad de una predicción correcta; así, entre más alta sea la precisión, mejor es el clasificador [Zaki and Meira Jr, 2019].

La razón para seleccionar la precisión (*Acc*) como métrica es que el conjunto de *datos*

a analizar está *balanceado* [Thomas et al., 2013], además de que esta métrica es usada ampliamente en la literatura de decodificación del MRCP [Shakeel et al., 2015].

2.7.2. Evaluación del clasificador

Típicamente, el conjunto de datos de entrada \mathbf{D} se divide al azar en un *conjunto de entrenamiento* y un *conjunto de evaluación*. El conjunto de entrenamiento se usa para aprender el modelo M , mientras que el conjunto de evaluación para evaluar la métrica θ , véase la Figura 2.10a. La *validación cruzada* y el *remuestreo* son dos métodos comunes para calcular el valor esperado y la varianza de una métrica de rendimiento dada [Zaki and Meira Jr, 2019].

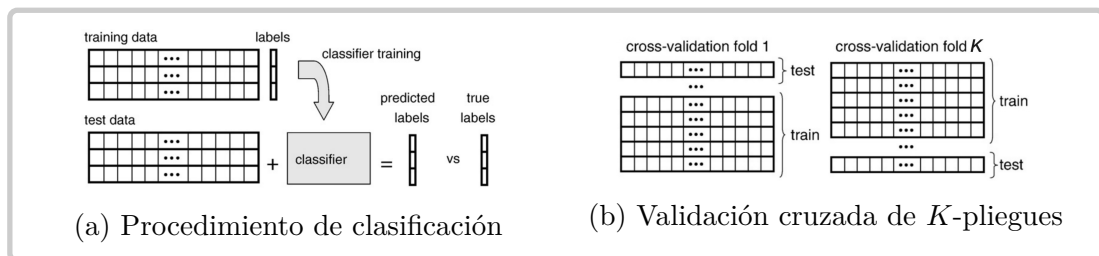


Figura 2.10: Esquema típico de clasificación

Fuente: [Pereira et al., 2009]

Validación Cruzada de K -pliegues

La validación cruzada divide el conjunto de datos \mathbf{D} en K partes de igual tamaño, llamados *pliegues*, es decir, $\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_K$. Cada pliegue \mathbf{D}_i es, en turno, tratado como el conjunto de evaluación, con los pliegues restantes comprendiendo el conjunto de entrenamiento $\mathbf{D} \setminus \mathbf{D}_i = \bigcup_{j \neq i} \mathbf{D}_j$, véase la Figura 2.10b. Después del entrenamiento del modelo M sobre $\mathbf{D} \setminus \mathbf{D}_i$, se evalúa el rendimiento sobre el conjunto de evaluación \mathbf{D}_i para obtener el i -ésimo estimado θ_i . El valor esperado de la métrica de rendimiento puede estimarse como [Zaki and Meira Jr, 2019]:

$$\hat{\mu}_\theta = \frac{1}{K} \sum_{i=1}^K \theta_i \quad (2.23)$$

y su varianza como:

$$\hat{\sigma}_\theta^2 = \frac{1}{K} \sum_{i=1}^K (\theta_i - \hat{\mu}_\theta)^2 \quad (2.24)$$

Nótese que la validación cruzada de K -pliegues puede repetirse múltiples veces. Usualmente K se selecciona 5 ó 10. El caso especial cuando $K = n$ es llamado validación cruzada *deja-uno-fuera*, donde el conjunto de evaluación comprende un punto simple y los datos restantes se usan para entrenamiento [Zaki and Meira Jr, 2019].

2.7.3. Significancia estadística para precisión de clasificación

Reportar la métrica de clasificación sola no es a menudo suficiente. Incluso precisiones tan altas como 90 % pueden no tener sentido si el número de ensayos es muy bajo o si las clases no están balanceadas [Billinger et al., 2012]. Para un número de clases dada c , el porcentaje de *nivel de oportunidad teórico* (*chance level* en inglés) de clasificación está dado por $100/c$. Por ejemplo, para un problema de 2-clases, el nivel de oportunidad es $100/2 = 50\%$. Este umbral está basado en la suposición de un tamaño de muestras infinito [Combrisson and Jerbi, 2015]. En la práctica, el *nivel de oportunidad empírico/práctico* depende del número de muestras disponibles [Combrisson and Jerbi, 2015]. Un método ampliamente usado para determinar *significancia estadística* es usar *intervalos de confianza* [Gupta, 2012].

Cuando la probabilidad de éxito es p , entonces la probabilidad de lograr exactamente K éxitos de N ensayos totales independientes sigue una *distribución binomial*. Así, es posible construir un intervalo de confianza con un *nivel de confianza* de $1 - \alpha$, alrededor de cualquier precisión de clasificación p ; por ejemplo para el nivel de oportunidad teórico p_0 , con $K = N/c$. Un intervalo de confianza ampliamente usado para la métrica de precisión en BCI es el *Wald Ajustado*, el cual se basa en aproximar la distribución binomial con una distribución normal y está dado por [Billinger et al., 2012]:

$$\hat{p} = \frac{K + 2}{N + 4} \quad (2.25a)$$

$$p_u = \hat{p} + z_{1-\alpha/2} \sqrt{\frac{\hat{p}(1 - \hat{p})}{N + 4}} \quad (2.25b)$$

$$p_l = \hat{p} - z_{1-\alpha/2} \sqrt{\frac{\hat{p}(1 - \hat{p})}{N + 4}} \quad (2.25c)$$

en donde \hat{p} es la *probabilidad de clasificación correcta estimada ajustada* (agregando 2 éxitos y 2 fracasos) con K ensayos correctos y N ensayos totales, p_l y p_u los *límites de confianza inferior y superior*, $z_{1-\alpha/2}$ el *valor crítico z* (de un *valor- p* $1 - \alpha/2$ de cola izquierda en la distribución normal) para el intervalo de confianza de dos-lados y α el *nivel de significancia*, es decir, la probabilidad de que el *parámetro verdadero de población* (en nuestro caso, p verdadero) se encuentre fuera del intervalo de confianza. Para el caso de un intervalo de confianza de un-lado se puede usar $z_{1-\alpha}$ en lugar de $z_{1-\alpha/2}$.

Finalmente, usando el intervalo de confianza Wald Ajustado se pueden usar dos formas para determinar si una precisión de clasificación \hat{p} es significativa. La primera es verificar que \hat{p} es estrictamente mayor que el límite de confianza superior de p_0 , es decir, si $\hat{p} > p_u$ [Müller-Putz et al., 2008], típicamente usada en la literatura de MRCP. La segunda es verificar que el límite de confianza inferior de \hat{p} es estrictamente mayor que el nivel de oportunidad teórico p_0 , es decir, si $p_l > p_0$ [Billinger et al., 2012]. Es importante mencionar que sólo estamos interesados en precisiones mayores que el nivel de oportunidad teórico.

Capítulo 3

Potencial Cortical Relacionado a Movimiento (MRCP)

Este capítulo da una introducción a la señal de control endógena ampliamente conocida como Potencial Cortical Relacionado a Movimiento (MRCP), la cual fue analizada en este trabajo. Se discuten los principales componentes de este, el modo de registro, así como los factores que lo afectan.

3.0.1. Definición y componentes del MRCP

El *Potencial Cortical Relacionado a Movimiento* (MRCP) es un SCP de baja amplitud (5-30 μV [Wright et al., 2011]) que ocurre durante un movimiento ejecutado (ME) o imaginado (MI) [Shibasaki and Hallett, 2006]. Para usarse en una BCI el MRCP debe ser detectado muy cerca del inicio (*onset* en inglés) del movimiento, de lo contrario el usuario no se sentiría en control [Sani et al., 2016].

Debido a que la latencia (es decir, el retardo entre la *intención motora* y la correspondiente *detección*) del MRCP (usualmente menor a $\pm 300\text{ms}$ [Karimi et al., 2017]) se encuentra dentro del rango de -500 a 300ms con respecto al onset de movimiento, importante para inducir cambios de plasticidad [Jochumsen et al., 2014b], ha sido recientemente usado para aplicaciones BCIs en tiempo real, particularmente en sistemas de neuromodulación [Mrachacz-Kersting et al., 2016].

El MRCP se puede dividir en dos fases, *pre-movimiento* y *post-movimiento*. La intención de movimiento está codificada en la fase de pre-movimiento [Shibasaki and Hallett, 2006], véase la [Figura 3.1](#).

tora suplementaria (SSMA), mientras que la CNV-tardía comienza de 1 a 0.5s antes del estímulo de ejecución (S2), con el pico de negatividad alrededor de S2 [Walter et al., 1964], y tiene su máxima amplitud sobre la corteza motora [Hamano et al., 1997].

Finalmente, la etapa de post-movimiento a ritmo propio se le conoce como *potencial de monitoreo de movimiento* (MMP), puede durar de 1 a 1.5s después del onset de movimiento y se piensa que está relacionada con el control y precisión del movimiento [Shibasaki and Hallett, 2006]. En la CNV, esta última forma de onda ha sido relacionada tanto a la realimentación somatosensorial como al nivel de atención durante la ejecución de la tarea [Hamano et al., 1997].

Hasta ahora no hay un consenso acerca de las diferencias exactas entre BP y CNV, sin embargo se pueden resaltar aquellas más reconocidas. La distribución es más lateralizada al sitio contralateral en BP, mientras que más bilateral en CNV [Nam et al., 2018]. Cuando el tiempo entre el estímulo S1 y S2 es mayor que 5s, la forma de la CNV-tardía es igual a la del BP [Deecke et al., 1976]. [Jankelowitz and Colebatch, 2002] reportaron que la depresión negativa es más temprana en CNV que en BP. En [Ruchkin et al., 1986] se encontró que el CNV puede generarse en ausencia de una respuesta motora. A pesar de estas diferencias el consenso es que ambos representan una preparación o intención de movimiento.

3.0.2. Registro

Usualmente los MRCPs se registran sobre las áreas corticales motoras (véase la [Figura 2.4](#)), normalmente con referencia a electrodos mastoides enlazados. Se recomienda estudiar a los participantes con los ojos abiertos (para minimizar la actividad alfa) y enfocando su mirada en un solo punto (para evitar el movimiento de ojos) [Colebatch, 2007]. Como en cualquier otro registro de EEG, se requiere mantener las impedancias de electrodos bajas como se indicó en la [Subsección 2.3.1](#). La banda de paso de frecuencias del amplificador de registro normalmente se escoge desde 0.1Hz o menos a entre 100 y 300 Hz [Colebatch, 2007].

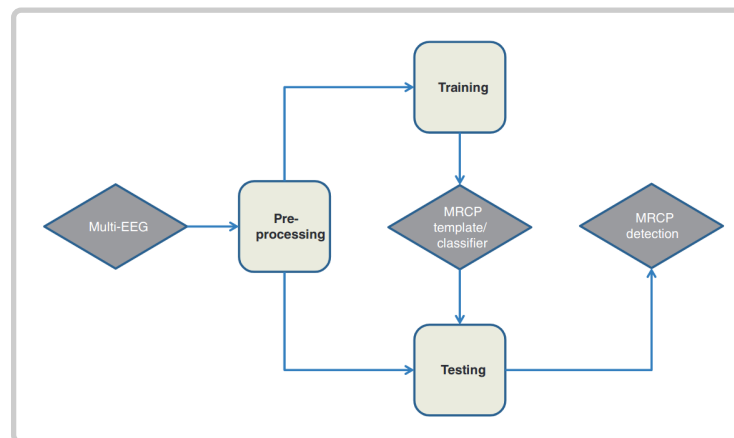


Figura 3.2: Procesamiento para detección de MRCP

Fuente: [Jiang et al., 2014]

3.0.3. Procesamiento de señal de MRCP para detección de la intención de movimiento

La detección de la intención de movimiento mediante la señal MRCP usualmente consiste de tres pasos: *preprocesamiento*, *entrenamiento* y *detección*, véase la Figura 3.2. Es importante resaltar que un algoritmo de detección (*detector*) únicamente ‘*detecta*’ la señal de interés (es decir, si/no se identifica el MRCP de ensayo simple en la transmisión continua EEG), mientras que un algoritmo de clasificación (*clasificador*) discrimina clases (por ejemplo MRCPs de ensayo simple entre distintas condiciones de movimiento) [Jochumsen et al., 2015]. Para resolver el problema de detección, además de los clasificadores tradicionales que usan como clases: *señal* y *no-señal* como en [Bhagat et al., 2016; López-Larraz et al., 2016], el uso del *filtro adaptado* (*Matched Filter* en inglés), con extracción supervisada de la plantilla MRCP, es una técnica que ha mostrado interés en experimentos en línea durante la última década, por ejemplo en [Xu et al., 2013; Jochumsen et al., 2014a, 2015; Hassan et al., 2015].

La etapa de preprocesamiento consiste principalmente de un filtrado pasa-bandas. A pesar de que no hay un consenso sobre los mejores parámetros para el filtro pasa-banda, en Garipelli et al. [2013] se encontró que [0.1, 1] Hz produjo la mejor *precisión* de detección para CNV, sin embargo, la banda [0.05, 5] Hz es aceptable [Jiang et al., 2014]. En esta etapa es posible incluir también pasos de procesamiento opcionales como filtrado espacial, técnicas de separación de fuentes, entre otros, a fin de mejorar las características del MRCP y/o remover artefactos como EOG y EMG [Jiang et al.,

2014].

Posterior al preprocesamiento se entrena un *detector MRCP*, ya sea un filtro adaptado o un clasificador, usando un conjunto de datos de una *sesión de entrenamiento*. Normalmente, durante esta sesión se le pide al participante ejecutar una serie de movimientos ME o MI (por ejemplo dorsiflexión, flexión de muñeca, etc.) ya sea siguiendo una señal o a su propio ritmo [Jiang et al., 2014]. Una sesión típica de entrenamiento consiste de 20 a 30 tareas [Jiang et al., 2014]. A fin de obtener el MRCP es necesario una referencia que se asocia al inicio del movimiento. En un paradigma basado en estímulos, ya sea en tareas de MI o ME, el inicio de la tarea está predefinido por el sistema, sin embargo, en uno a ritmo propio sólo es posible de forma precisa para ME, siendo el EMG del músculo objetivo una forma para obtener el instante del inicio del movimiento [Jiang et al., 2014]. Una vez que se tiene el onset de movimiento(MO)/imaginación(IO) se crean *épocas* alrededor de éste ($t = 0$), normalmente 1.5-2s antes y 0.5-1s después [Colebatch, 2007], conocidas cada una como *MRCP de ensayo simple (Single-trial MRCP* en inglés). Al promedio de todos los MRCPs de ensayo simple ([Colebatch, 2007] recomienda al menos de 50 a 100 ensayos) es lo que usualmente se conoce como ‘*MRCP*’ o ‘*Plantilla MRCP*’, sin embargo, algunos autores le denominan *MRCP Promedio (Average MRCP* en inglés). Adicionalmente, al promedio de MRCPs de todos los participantes en un estudio se conoce como *MRCP de Gran-Promedio (Grand-average MRCP* en inglés). Nótese que debido a que el MRCP requiere del promedio de varias decenas de ensayos, es poco práctico en un experimento en línea. Es por ello que para la señal de control de una BCI, generalmente se procesa el MRCP de ensayo-simple.

Finalmente, una vez que el detector se ha entrenado, ya sea directamente con la plantilla MRCP en el filtro adaptado o bien con características de los MRCPs de ensayo simple en el clasificador, la etapa de detección en línea puede comenzar. Aquí, el detector recibe los datos de evaluación usando el mismo preprocesamiento y genera como salida un valor binario, en donde por ejemplo ‘+1’ significa MRCP y ‘-1’ No-MRCP [Jiang et al., 2014]. Adicionalmente, en la práctica para mejorar la robustez de todo el proceso de detección, la salida bruta del detector puede someterse a un post-procesamiento, tal como el voto mayoritario (MV), por ejemplo una detección positiva final únicamente ocurriría cuando dos de las últimas tres detecciones fueran positivas [Jiang et al., 2014].

3.0.4. Factores que afectan el MRCP

La latencia y magnitud de MRCPs puede ser influenciada por varios factores con relación al tipo de [Shibasaki and Hallett, 2006]: *i*) *tarea* (es decir, si son reacciones o acciones), *ii*) *movimiento*, por ejemplo si son unilaterales/bilaterales, realizados con el lado dominante o no y *iii*) *característica de movimiento*, tales como nivel de intención, fuerza, velocidad, precisión, tasa de repetición, complejidad, etc., así como *iv*) *condiciones propias del usuario* tales como: concentración, lesiones patológicas de estructuras cerebrales, entre otras.

Capítulo 4

Visión Artificial

Este capítulo hace una breve presentación de lo que es un sistema de visión artificial. Además, se explican las técnicas de procesamiento y análisis de imágenes usadas en el algoritmo de visión artificial de este trabajo.

4.1. Sistema de Visión Artificial

La *Visión Artificial* (VA) es un subcampo de la inteligencia artificial y se refiere a la utilización de la descripción de una escena, con base en su imagen, para interactuar de manera inteligente con dicha escena mediante un sistema robótico, como los brazos manipuladores, robots móviles, humanoides, drones, vehículos autónomos, etc. Cabe mencionar que esta descripción o modelo se debe hacer con base en las imágenes de la escena analizada y que debe incluir tan solo aquella información de utilidad para la tarea encomendada al robot, aunque puede incorporar información a priori o bien información de otros sensores. Un sistema basado en VA está compuesto principalmente de los siguientes subsistemas [Zannatha, 2012] (véase la [Figura 4.1](#)):

1. Adquisición de Imágenes (Escena \rightarrow Imagen)

Este proceso consiste en cuatro pasos: 1) la formación de la imagen óptica en un plano sensible, 2) la conversión en una señal eléctrica, 3) la digitalización de dicha señal eléctrica y finalmente 4) el almacenamiento de la imagen digital en la memoria del sistema de cómputo del robot [Zannatha, 2012].

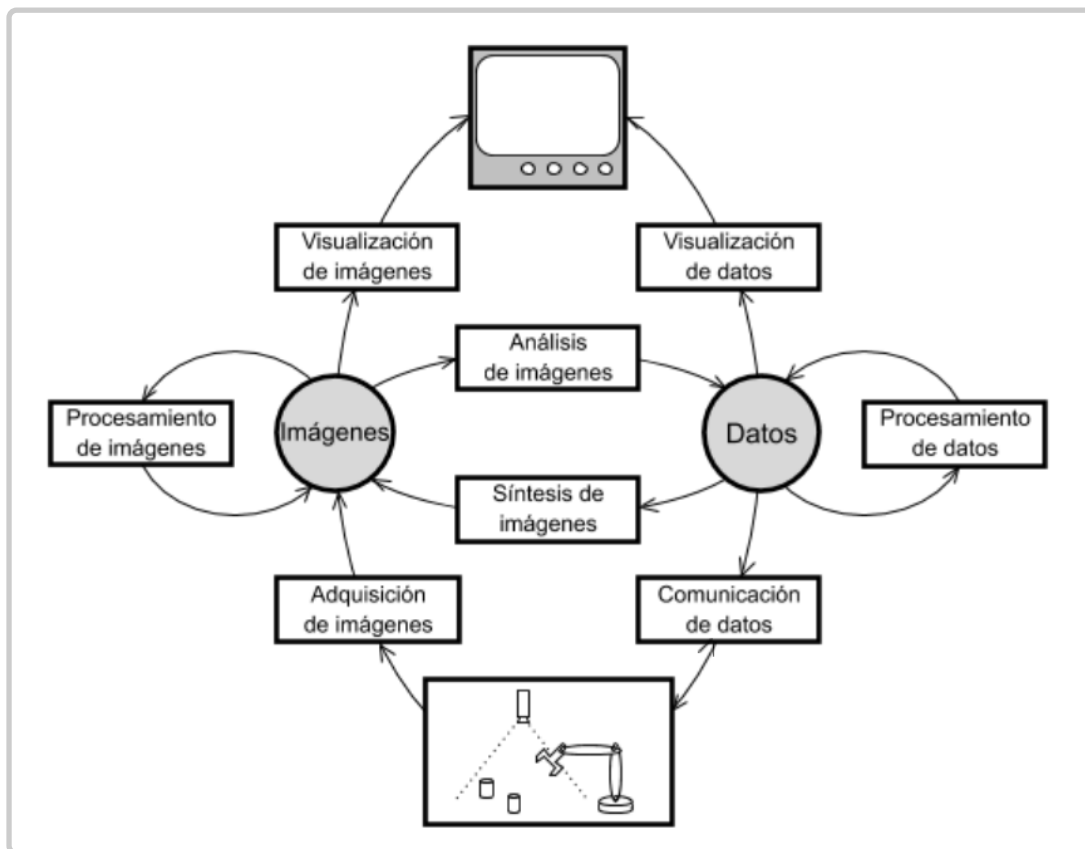


Figura 4.1: Diagrama de un Sistema de Visión Artificial para robots

2. Procesamiento de Imágenes (PDI) (Imagen \rightarrow Imagen)

Esta etapa en todo sistema de VA tiene únicamente dos funciones: 1) mejorar la imagen, es decir, eliminar ruidos y artefactos, compensación de problemas de iluminación y eliminación de sombras, entre otras tareas; y 2) realzado de características tales como esquinas y contornos o mejoras en la nitidez de la imagen. La primera de estas funciones suele obtenerse mediante la aplicación de filtros espaciales pasa bajas, lo cual deteriora características de alta frecuencia como esquinas y contornos, las cuales se realzan mediante filtraje de paso alto, el cual acaba por amplificar el ruido. A pesar de estas dificultades, el PDI no representa mayor dificultad, ya que se cuenta con una gran cantidad de procedimientos en todo sistema de visión como OpenCV o MATLAB [Zannatha, 2012].

3. Análisis de Imágenes (ADI) (Imagen \rightarrow Datos)

Este sistema es el encargado de hacer una descripción de la imagen, es decir, debe

obtener los modelos que permiten hacer todo tipo de inferencia o de medición que requiere el robot para realizar su tarea de interacción con el entorno. Una vez que la imagen ha sido procesada en el subsistema de análisis, ésta queda representada por un conjunto de datos o modelos que describen la imagen en los términos requeridos por la aplicación. Así, las tres principales etapas en el ADI son: 1) la segmentación ya sea mediante criterios de continuidad (bordes, esquinas y contornos) o bien similitud (regiones y texturas); 2) la extracción de características tales como líneas, puntos, texturas, regiones, colores, entre otras más, es decir, se trata de obtener los parámetros de los modelos elegidos; y 3) la descripción de la imagen, es decir, la creación de las descripciones o modelos de la imagen utilizando las características extraídas y los parámetros que se midieron [Zannatha, 2012].

4. Procesamiento de datos

Los modelos o descripciones de bajo nivel obtenidos en el subsistema ADI se pueden procesar en este subsistema a fin de obtener descripciones más detalladas o de mayor nivel. Algunos ejemplos de estos procesamientos son la reconstrucción 3D, el SLAM (Simultaneous Localization And Mapping), interpretación de las señales de tránsito, etc. [Zannatha, 2012].

5. Comunicación (bidireccional) de datos con el robot

Este subsistema tiene dos funciones esenciales: 1) recibir información complementaria por parte de los sensores del robot (posición y velocidad articular, mediciones inerciales, información de Lidar, sensores infrarrojos o ultrasónicos) y 2) enviar al robot datos como posición y orientación de objetos por manipular, identidad de los objetos de la escena, su relación de orden (encima, detrás, al lado, etc.), seguimiento de la carretera en el caso de vehículos autónomos, presencia de obstáculos, etc. [Zannatha, 2012].

A continuación se describe brevemente las técnicas de procesamiento y análisis de imagen usados en el presente trabajo.

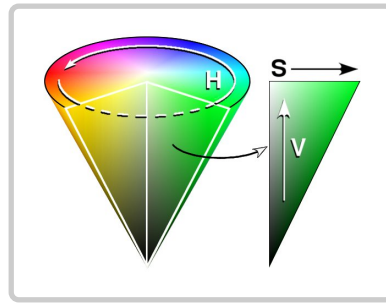


Figura 4.2: Espacio de color HSV

4.2. Técnicas de Procesamiento de Imagen

4.2.1. Espacio de color

Antes del procesamiento de imagen es importante definir el espacio de color. Uno de los problemas más comunes en la práctica son las condiciones de luz. De esta manera, es necesario aislar o controlar la iluminación. Uno de los espacios de color que permite esto, es el espacio HSV en donde es posible controlar el *matiz* H ($0-360^\circ$) que representa el tipo de color, la *saturación* S ($0-100$) que define la pureza del color y el *valor* V ($0-100$) que indica el brillo del color. Dicho espacio se explica por el conocido *cono HSV*, mostrado en [Figura 4.2](#).

4.2.2. Segmentación

La *segmentación de imágenes* es el proceso de partición de una imagen en regiones *homogéneas* y *conectadas*. La homogeneidad puede ser definida en términos del valor de escala de grises, color, textura, forma, movimiento, etc. Así el objetivo de la segmentación consiste en asignar etiquetas a todos y cada uno de los píxeles de una imagen y agruparlos en función de algunas medidas de similitud. Antes de aplicar cualquier algoritmo de segmentación, generalmente se aplican algunos pasos de pre-procesamiento. Adicionalmente, el post-procesamiento (tal como operaciones morfológicas) se realiza para obtener mejores resultados de segmentación [[Bhuyan, 2019](#)].

Existen una gran variedad de algoritmos de segmentación en la literatura, tales como la umbralización, basados en región, basados en detección de bordes, por agrupamiento, etc. En este trabajo ocupamos el método de *umbralización de histograma*, que es uno de los más sencillos.

Umbralización

La *umbralización* consiste en asignar un valor (generalmente binario, pero no necesariamente) a una máscara basado en la imposición de umbrales bajo cierto criterio. Por ej. dada una imagen en escala de grises representada por $f(x, y)$, la salida es una máscara dada por:

$$g(x, y) = \begin{cases} 1 & \text{si } f(x, y) \geq T \\ 0 & \text{si } f(x, y) < T, \end{cases} \quad (4.1)$$

donde T es un umbral que discrimina. El método de umbralización *global* más conocido es el basado en histograma. Un *histograma* indica la distribución de un componente de color en una imagen. Este método se usa cuando una imagen tiene regiones homogéneas bien definidas, es decir, no es de mucha ayuda si la imagen exhibe una gran variedad de intensidades de píxeles. El (los) valor(es) de umbral pueden ser determinados manualmente (p. ej. el punto valle entre dos picos del histograma), semi-automática (p. ej. el mejor umbral de la vecindad de píxeles actual) o automática (p. ej. los métodos de *Otsu* y *Riddler-Calvard*) [Bhuyan, 2019].

Para el caso de la umbralización de histograma en el espacio de color HSV, se calcula una máscara por cada componente, es decir, se calcula un histograma y se selecciona un umbral bajo (L) y alto (H) para cada componente como se indica en [Ecuación 4.2](#). Finalmente, aplicando la operación *bit-a-bit* AND a las 3 *máscaras binarias*, podemos obtener la imagen binaria que contiene las áreas segmentadas.

$$\begin{aligned} g_1(x, y) &= \begin{cases} 1 & \text{si } H_L \leq H(x, y) \leq H_H \\ 0 & \text{De otro modo} \end{cases} \\ g_2(x, y) &= \begin{cases} 1 & \text{si } S_L \leq S(x, y) \leq S_H \\ 0 & \text{De otro modo} \end{cases} \\ g_3(x, y) &= \begin{cases} 1 & \text{si } V_L \leq V(x, y) \leq V_H \\ 0 & \text{De otro modo} \end{cases} \end{aligned} \quad (4.2)$$

4.2.3. Filtrado espacial

El *filtrado espacial* es una operación que se aplica a una imagen para *resaltar* o *suprimir* información espacial. Matemáticamente, es una operación de *convolución* entre

una imagen y un *kernel* (también conocido como *matriz de convolución* o *máscara*) predefinido. Esta operación es *local* en el sentido de que se modifica el valor de cada píxel de acuerdo con los valores de sus píxeles vecinos. La *frecuencia espacial* se define como la magnitud de cambios de intensidad de píxel por unidad de distancia en una zona de la imagen (i.e., No. *cambios*/ No. de *pix*²). De esta manera, áreas de la imagen con pocos cambios se consideran como contenido de *baja-frecuencia*, mientras que aquellas con muchos cambios como contenido de *alta-frecuencia*. Así, de acuerdo a la banda de paso, los filtros espaciales se pueden clasificar en: *filtros pasa-bajas* que se usan para *enborronamiento* o *suavisado* y para reducción de ruido (p. ej. *promediado*, *gaussiano*, *mediana*, etc.), *filtros pasa-altas* que hacen *afilado*, es decir, resaltar detalles finos (p. ej. el *laplaciano*, *sobel*, etc.) o *filtros híbridos* que combinan características de los dos anteriores (p. ej. el *filtro bilateral*). En este trabajo se usó los filtros mediana y bilateral.

4.2.4. Filtro Mediana

El *filtro mediana* es un filtro de orden estadístico, en donde cada píxel \mathbf{p} se reemplaza por la mediana de la intensidad de sus píxeles vecinos \mathbf{q} , i.e.:

$$g(\mathbf{p}) = \text{mediana}(\mathcal{S}), \quad \mathcal{S} = \{I_{\mathbf{q}} | \mathbf{q} \text{ es vecino de } \mathbf{p}\} \quad (4.3)$$

donde $g(\mathbf{p})$ es la imagen filtrada. Este filtro es particularmente útil para remover el *ruido de disparo* (también conocido como *ruido de sal y pimienta*) que ocasiona valores de intensidad muy grandes, y se debe a que el valor del ruido de disparo suele estar muy por fuera de los valores reales en la vecindad. Adicionalmente, esta técnica no emborrona mucho los bordes y puede ser aplicada iterativamente [Szeliski, 2010].

4.2.5. Filtro Bilateral

El *filtro bilateral* es una técnica no lineal que puede emborronar una imagen preservando sus bordes. Matemáticamente se expresa como [Paris et al., 2008]:

$$g(\mathbf{p}) = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in \mathcal{S}} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|) G_{\sigma_r}(|I_{\mathbf{p}} - I_{\mathbf{q}}|) I_{\mathbf{q}} \quad (4.4a)$$

$$W_{\mathbf{p}} = \sum_{\mathbf{q} \in \mathcal{S}} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|) G_{\sigma_r}(|I_{\mathbf{p}} - I_{\mathbf{q}}|) \quad (4.4b)$$

$$G_{\sigma}(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}\|^2}{2\sigma^2}\right), \quad \mathbf{x} \in \mathbb{R}^n \quad (4.4c)$$

donde $g(\mathbf{p})$ es la imagen filtrada, que se obtiene del promedio ponderado normalizado de la intensidad (I) de cada píxel \mathbf{q} en la vecindad \mathcal{S} alrededor del píxel central \mathbf{p} . El peso de $I_{\mathbf{q}}$ es función de su *cercanía espacial* dado por el *kernel gaussiano* (2D) *espacial* G_{σ_s} y de su *diferencia de intensidad* por el *kernel gaussiano* (1D) *de rango* G_{σ_r} . De esta manera, aquí $\sigma_s > 0$ es un parámetro que define el tamaño de la vecindad, mientras que $\sigma_r > 0$ define el tamaño de diferencia de color. Es importante notar que cuando $\sigma_r \rightarrow 0$ (independiente de σ_s) el borde es más afilado, mientras que si $\sigma_r \rightarrow \infty$ la [Ecuación 4.4](#) tiende a un *filtro gaussiano*. Por otra parte, si $\sigma_s \rightarrow 0$ cuando $\sigma_r \rightarrow 0$ no ocurre suavizado, mientras que si $\sigma_s \rightarrow \infty$ no emborronará un borde siempre que σ_r sea menor que la intensidad del borde [Paris et al., 2008]. Una desventaja de este método es que es considerablemente más lento que sus contrapartes de emborronamiento.

4.3. Análisis de imagen

Debido a que en este trabajo obtuvimos la postura (posición + orientación) 2D de objetos sobre una mesa de trabajo desde una imagen (corregida en perspectiva), se presenta en esta sección los detalles de la Transformación de Homografía.

4.3.1. Homografía

El *modelo de proyección en perspectiva*, se ilustra en la [Figura 4.3a](#) y está representado por la *matriz de proyección homogénea* o *matriz de la cámara* $\mathbf{P} \in \mathbb{R}^{3 \times 4}$. Esta matriz permite para mapear un punto del mundo $\mathbf{x} \in \mathbb{P}^3$, a un punto en la imagen $\mathbf{x}' \in \mathbb{P}^2$,

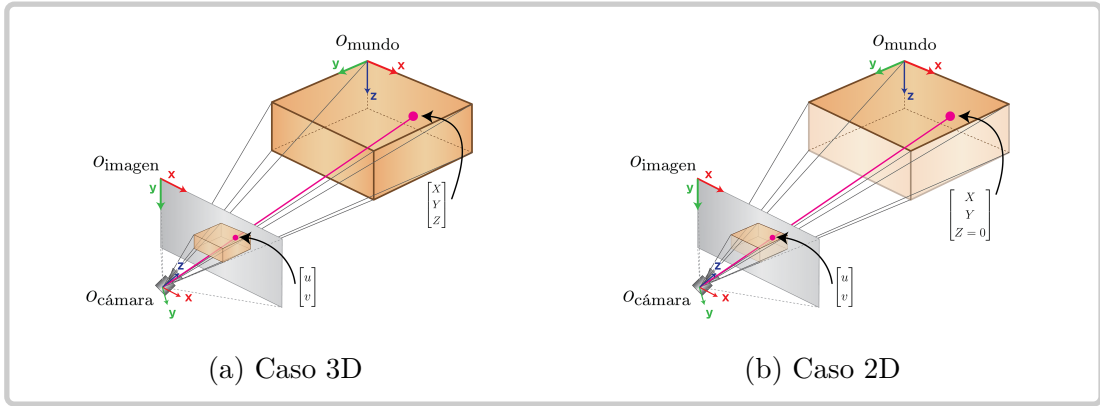


Figura 4.3: Modelo de proyección en perspectiva

esto es, $\mathbf{P} : \mathbb{P}^3 \rightarrow \mathbb{P}^2$, y está definida como [Hartley and Zisserman, 2003]:

$$\lambda \mathbf{x}' = \mathbf{P}\mathbf{x} = \mathbf{K}[\mathbf{R}|\mathbf{t}]\mathbf{x} = \mathbf{K}[\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{r}_3 \ \mathbf{t}]\mathbf{x}$$

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (4.5)$$

donde \mathbf{K} es la *matriz de calibración de cámara* o *matriz de parámetros intrínsecos*, $[\mathbf{R}|\mathbf{t}]$ la *matriz de parámetros extrínsecos* que representa la postura (posición \mathbf{t} + orientación \mathbf{R}) del sistema coordenado de la cámara con respecto del sistema coordenado del mundo, λ un factor de escala no nulo arbitrario, f_x, f_y la distancia focal en términos de la distancia entre píxeles, (c_x, c_y) el punto principal de la imagen y s el parámetro de sesgo (normalmente cero para la mayoría de las cámaras). Nótese que \mathbf{P} tiene 11 grados de libertad pues es una matriz 3×4 definida hasta una escala arbitraria [Hartley and Zisserman, 2003], los cuales corresponden a los parámetros $f_x, f_y, s, c_x, c_y, t_x, t_y, t_z$ y los tres ángulos de orientación contenidos en la matriz ortogonal \mathbf{R} . Finalmente, una vez obtenida \mathbf{P} , los puntos en la imagen en las coordenadas no-homogéneas se obtienen dividiendo $\lambda \mathbf{x}'$ por λ . Para el caso particular en el cual todos los puntos de interés yacen en un plano, por ejemplos el suelo o en una mesa, todos los puntos de interés tendrán $Z = 0$. De este modo, sin pérdida de generalidad, el modelo de proyección en

perspectiva se reduce a (véase [Figura 4.3b](#)):

$$\begin{aligned} \lambda \mathbf{x}' &= \mathbf{H}\mathbf{x} = \mathbf{K}[\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t}]\mathbf{x} \\ \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ r_{31} & r_{32} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \end{aligned} \quad (4.6)$$

en donde $\mathbf{H} \in \mathbb{R}^{3 \times 3}$ se conoce como *Homografía*, la cual se define como un mapeo lineal invertible $\mathbf{H} : \mathbb{P}^2 \rightarrow \mathbb{P}^2$, que transforma coordenadas homogéneas entre dos planos y se define hasta un factor de escala. Por tanto, la Homografía \mathbf{H} tiene solo 8 grados de libertad. Normalmente $h_{33} = 1$ y se justifica por la observación de que \mathbf{H} está determinada solamente hasta la escala, y esta escala puede ser seleccionada tal que $h_{33} = 1$ [[Hartley and Zisserman, 2003](#)]. Nótese que por cada punto correspondiente conocido $(u, v) \rightarrow (X, Y)$, de la [Ecuación 4.6](#) se derivan las ecuaciones:

$$u = \frac{h_{11}X + h_{12}Y + h_{13}}{h_{31}X + h_{32}Y + 1} \quad (4.7)$$

$$v = \frac{h_{21}X + h_{22}Y + h_{23}}{h_{31}X + h_{32}Y + 1}, \quad (4.8)$$

las cuales al multiplicar por el denominador y reordenando términos obtenemos el siguiente sistema de ecuaciones lineales (SEL):

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} X & Y & 1 & 0 & 0 & 0 & -Xu & -Yu \\ 0 & 0 & 0 & X & Y & 1 & -Xv & -Yv \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix}. \quad (4.9)$$

Esto significa, que con un solo punto correspondiente conocido obtenemos un SEL de 2 ecuaciones con 8 incógnitas, lo cual es un problema mal planteado. De esta manera, para obtener 8 ecuaciones para que este SEL se vuelva cuadrado y (con suerte) invertible, necesitamos al menos 4 puntos correspondientes conocidos. Por lo tanto, \mathbf{H} se obtiene resolviendo el SEL con $M \geq 4$ puntos correspondientes conocidos con el requerimiento adicional de que no halla $M - 1$ puntos *colineales*, y es expresado como [Hartley and Zisserman, 2003]:

$$\underbrace{\begin{bmatrix} u_1 \\ v_1 \\ \vdots \\ u_M \\ v_M \end{bmatrix}}_{\mathbf{b} \in \mathbb{R}^{2M}} = \underbrace{\begin{bmatrix} X_1 & Y_1 & 1 & 0 & 0 & 0 & -X_1 u_1 & -Y_1 v_1 \\ 0 & 0 & 0 & X_1 & Y_1 & 1 & -X_1 v_1 & -Y_1 v_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_M & Y_M & 1 & 0 & 0 & 0 & -X_M u_M & -Y_M v_M \\ 0 & 0 & 0 & X_M & Y_M & 1 & -X_M v_M & -Y_M v_M \end{bmatrix}}_{\mathbf{A} \in \mathbb{R}^{2M \times 8}} \underbrace{\begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix}}_{\mathbf{h} \in \mathbb{R}^8}, \quad (4.10)$$

en donde el *SEL no-homogéneo* $\mathbf{A}\mathbf{h} = \mathbf{b}$ puede resolverse para la incógnita \mathbf{h} , usando técnicas estándar para resolver SEL (tal como *eliminación gaussiana*) en el caso donde $M = 4$ (el caso mínimo), o por técnicas de *mínimos-cuadrados* en el caso de un conjunto sobredeterminado de ecuaciones ($M > 4$). Otra alternativa para determinar \mathbf{H} es resolver un *SEL homogéneo* $\mathbf{A}\mathbf{h} = \mathbf{0}$ típicamente mediante el algoritmo de *transformación directa lineal* (DLT) [Hartley and Zisserman, 2003]. Finalmente una vez obtenida \mathbf{H} , las coordenadas no-homogéneas en la imagen se obtienen mediante Ecuación 4.7 y Ecuación 4.8.

Capítulo 5

Métodos

En este capítulo se explica la base de datos que se usó para el análisis de datos de EEG. Posteriormente, se describe el procesamiento de señal que se llevó a cabo para clasificación de lateralidad, en donde que usaron características del primer componente del MRCP, conocido como Potencial de Preparación (RP). Finalmente, se explica los elementos del sistema de manipulación propuesto, es cual consiste de: 1) un subsistema de visión, 2) un subsistema de robot y 3) una base de conocimiento.

5.1. Base de datos

Existen diversas bases de datos de EEG que estudian movimientos ejecutados o imaginados (MI/ME) de miembros superiores. Algunas ampliamente reconocidas que involucran miembro derecho/izquierdo son: *BCI Competition II Dataset III*, *BCI Competition II Dataset IV*, *BCI Competition IV Dataset II* y *Physionet EEG Motor Movement/Imagery* (EEGMMIDB) [Goldberger et al., 2000]. Para este trabajo seleccionamos esta última, la cual se describe a continuación.

5.1.1. Paradigma

La base de datos EEGMMIDB consiste de datos EEG de 109 sujetos sanos. Cada participante realizó 14 corridas experimentales: 2 corridas de referencia de un minuto (una con ojos abiertos y la otra con ojos cerrados), y tres corridas de dos minutos de cada una de las siguientes tareas [Goldberger et al., 2000]:

- **Tarea 1.** Abrir y cerrar muñeca derecha/izquierda si aparecía una pista en el lado derecho/izquierdo de la pantalla, respectivamente.
- **Tarea 2.** Imaginar: abrir y cerrar muñeca derecha/izquierda si aparecía una pista en el lado derecho/izquierdo de la pantalla, respectivamente.
- **Tarea 3.** Abrir y cerrar ambas muñecas o mover ambos pies si aparecía una pista arriba/abajo en la pantalla, respectivamente.
- **Tarea 4.** Imaginar: abrir y cerrar ambas muñecas o mover ambos pies si aparecía una pista arriba/abajo en la pantalla, respectivamente.

En cada corrida de dos minutos, los ensayos consistieron de una fase de descanso y la fase de la tarea, ambas con una duración de 4s [Goldberger et al., 2000]. En Figura 5.1 resumimos este protocolo experimental.

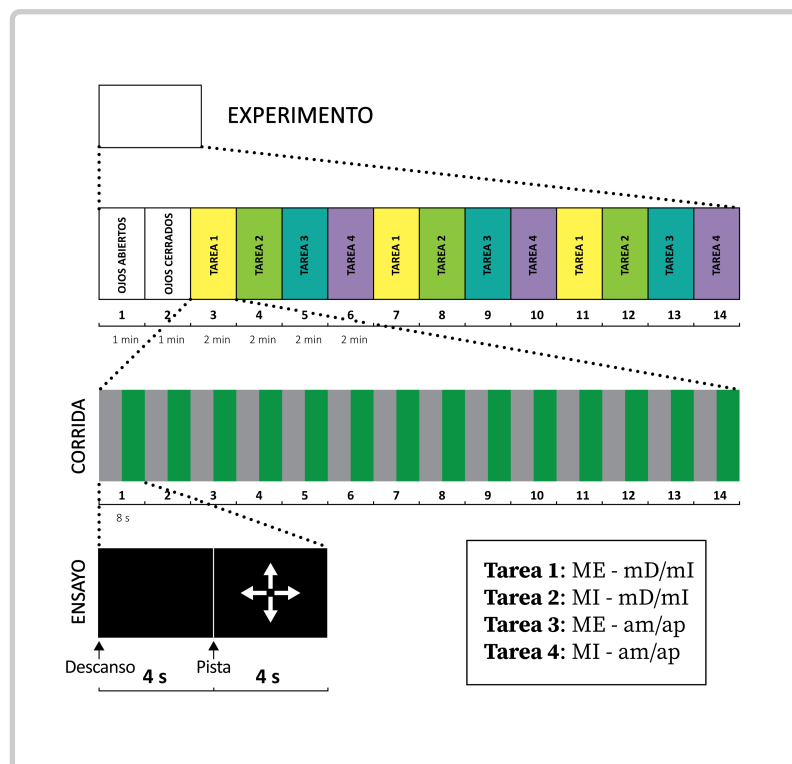


Figura 5.1: **Protocolo experimental de base de datos EEGMMIDB.** ME-mD/mI pertenece a Movimiento Ejecutado muñeca derecha/izquierda, MI-mD/mI a Movimiento Imaginado muñeca derecha/izquierda, ME-am/ap a Movimiento Ejecutado ambas muñecas/ ambos pies, MI-am/ap a Movimiento Imaginado ambas muñecas/ ambos pies.

5.1.2. Registro

El registro EEG se llevó a cabo usando el software BCI2000, un montaje de 64 canales (excluyendo algunos electrodos) de acuerdo al sistema 10-10, véase [Figura 5.2](#), y a una frecuencia de muestreo de 160 Hz [[Goldberger et al., 2000](#)]. La descripción oficial no menciona la posición los electrodos de referencia (REF) y tierra (GND), sin embargo suponemos que fue un montaje referencial, pues el estándar en BCI2000 frecuentemente coloca el electrodo de tierra sobre un mastoide (es decir, el hueso detrás de la oreja) y la referencia sobre el lóbulo de la oreja del lado opuesto [[Schalk and Mellinger, 2010](#)]. Adicionalmente tampoco se menciona algún preprocesamiento durante el registro.

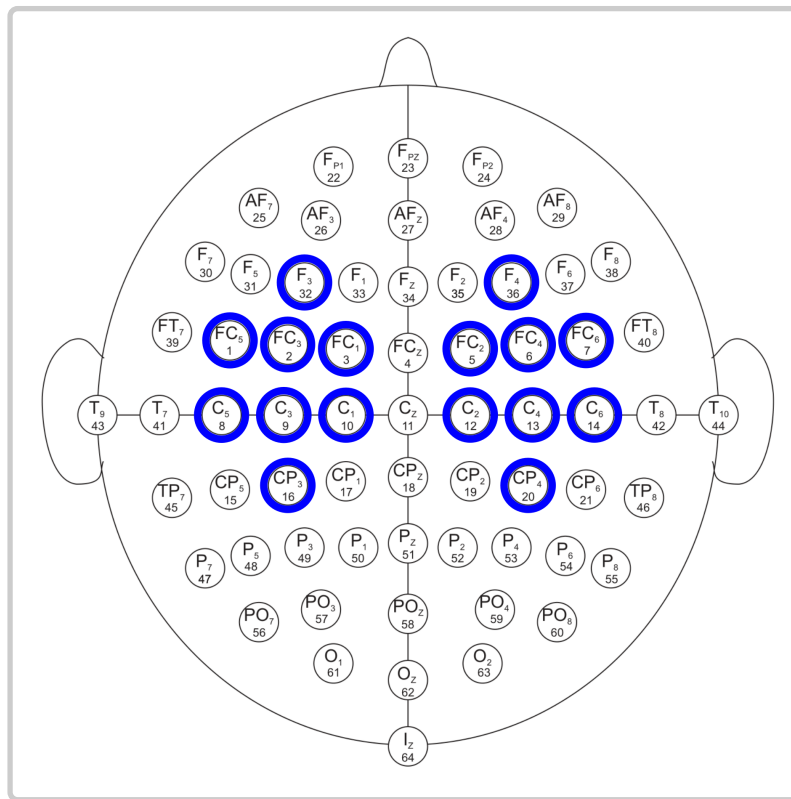


Figura 5.2: Montaje de EEG de la base de datos EEGMMIDB. Los canales analizados en este trabajo se indican en azul.

5.1.3. Estructura de datos

Los datos se proporcionan en Formato de Datos Europeo (EDF+). Adicionalmente se incluyen canales de anotación, en donde cada anotación incluye uno de tres códigos: T0

correspondiente a descanso, T1 al inicio de movimiento (real o imaginado) de muñeca izquierda o ambas muñecas y T2 al inicio de movimiento (real o imaginado) de muñeca derecha o ambos pies [Goldberger et al., 2000].

En total se consiguen en promedio al menos 42 eventos de *descanso* y 21 eventos de cada *tarea realizada* por cada tarea del protocolo experimental (i.e., **Tarea 1,2,3,4**) y cada sujeto.

5.2. Procesamiento de datos

En esta sección se presenta el análisis fuera de línea que se empleó para clasificar lateralidad de movimientos de miembros superiores considerando características del *Potencial Cortical Relacionado a Movimiento* (MRCP) de ensayo simple. Para llevar a cabo esto, se analizaron los datos EEG de la **Tarea 1** (corridas 3, 7 y 11) de la base de datos EEGMMIDB de 18 sujetos, véase la [Subsección 5.1.1](#). La *canalización* de procesamiento (conocida como *Pipeline* en inglés) propuesta se ilustra en la [Figura 5.3](#) y a continuación se explica brevemente en que consiste cada paso.

5.2.1. Selección de canales

Los canales que cubren las principales áreas corticales (de acuerdo a la organización somatotópica) asociadas a preparación de movimiento y ejecución de manos son C3, C4 y canales circundantes, lo cual principalmente se debe a la laterización de funciones motoras [Pfurtscheller and Lopes da Silva, 1999; Shibasaki and Hallett, 2006]. Debido a lo anterior y a fin de mantener la simplicidad computacional, en nuestro análisis seleccionamos los canales F3, F4, FC5, FC3, FC1, FC2, FC5, FC6, C5, C3, C1, C2, C4, C6, CP3 y CP4, los cuales están indicados en color azul en la [Figura 5.2](#).

5.2.2. Software

El análisis fue llevado a cabo en una PC HP con 2.30GHz Intel Core i5-6300HQ usando Ubuntu 16.04 LTS. El procesamiento de los datos EEG se realizó con el paquete software **MNE-Python** v0.17, el cual provee diversas herramientas para el procesamiento de datos EEG/ MEG, tales como análisis en tiempo real y visualización, filtraje, ICA, estadística/ *machine learning*, localización de fuentes, etc. [Esch et al., 2019].

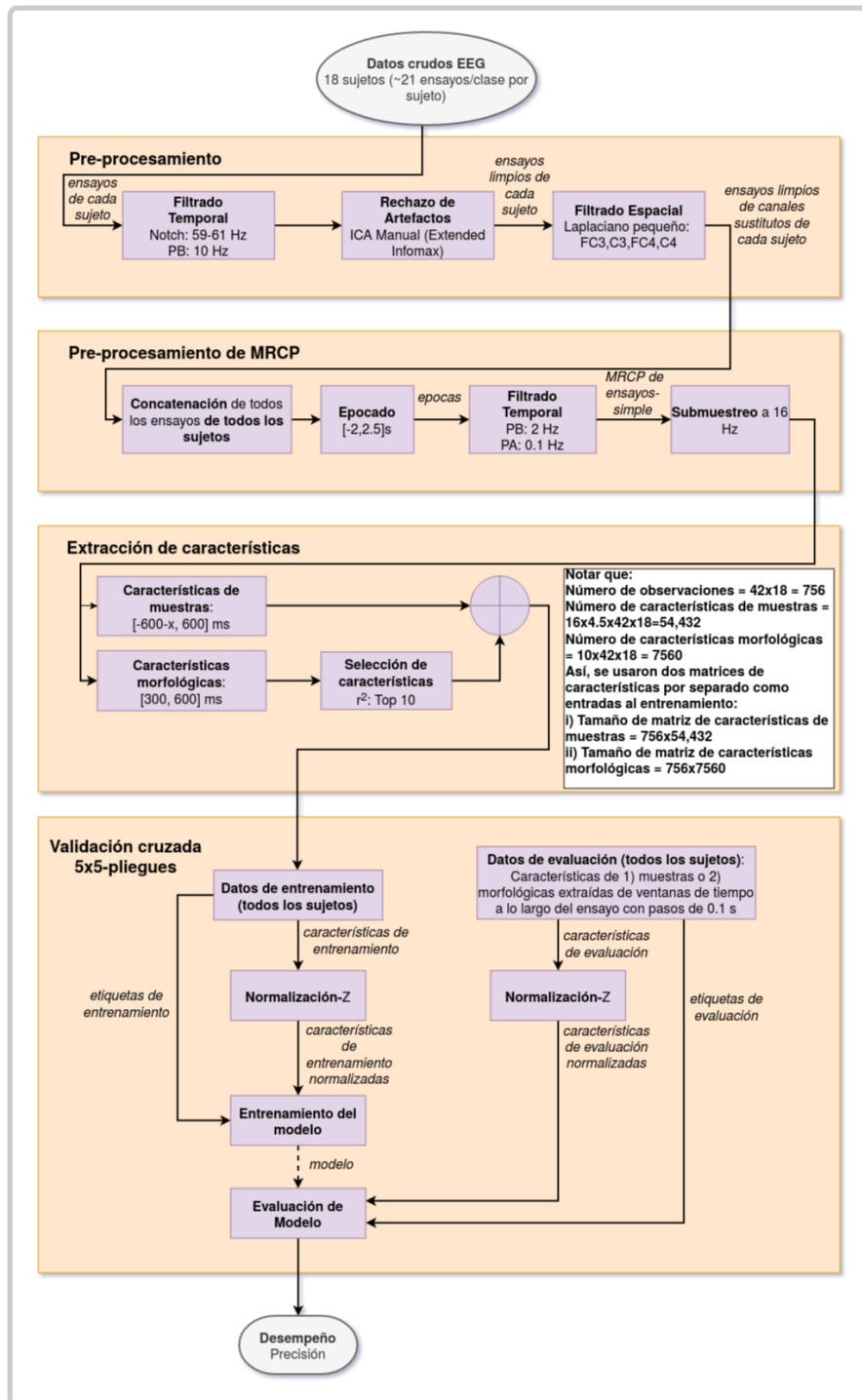


Figura 5.3: Canalización del procesamiento de datos

5.2.3. Preprocesamiento

En nuestro estudio, el preprocesamiento fue analizado por cada sujeto, generando en promedio 21 épocas del MRCP de ensayo simple por cada una de las tareas a realizar (es decir, derecha/izquierda). Posteriormente, los datos de todos los sujetos fueron concatenados y se procedió con los pasos de extracción de características, selección de características y clasificación.

Dados los datos continuos de los 16 canales EEG y el canal de anotación, se aplicó una etapa de *filtraje temporal*, *rechazo de artefactos* y *preprocesamiento para MRCP*. A continuación se explican brevemente estas etapas.

Filtrado temporal

Primeramente, se aplicó un *filtro notch* de tipo *Butterworth* de 4° orden con una banda de supresión de [59,61] Hz para remover el ruido de la línea de alimentación AC de 60 Hz (debido a que la base de datos usada fue registrada en Estados Unidos). Posteriormente, se aplicó un *filtro pasa-bajas* de tipo *Butterworth* de 4° orden de fase cero (filtro no-causal)(función `filtfilt()` de `scipy`) con una frecuencia de corte de 10 Hz a fin de mantener las frecuencias de la señal de interés ($\sim < 5$ Hz), así como facilitar la remoción de artefactos.

Rechazo de artefactos

En esta etapa, los datos continuos EEG filtrados fueron sujetos a una remoción de artefactos mediante un *Análisis de Componentes Independientes* (ICA) con el algoritmo *Extended-Infomax*, usando la biblioteca MNE. Primeramente, con la matriz de desmezcla \mathbf{W} , generada por ICA de MNE (opcionalmente MNE permite aplicar una reducción de dimensiones con PCA a la entrada), se obtuvieron los cursos de tiempo de los componentes independientes ($\mathbf{Y} = \mathbf{W}\mathbf{X}$) y los patrones espaciales (\mathbf{W}^{-1}). Posteriormente, se realizó una identificación de artefactos manualmente basados en el comportamiento tanto de los cursos de tiempo como de los patrones espaciales. Una vez identificados los componentes asociados a artefactos se procedió a reconstruir la señal continua EEG, al establecer a cero los filtros espaciales respectivos en $\widetilde{\mathbf{X}}_{clean} = \mathbf{W}^{-1}\mathbf{W}\mathbf{X}$, obteniendo finalmente una señal EEG libre de los artefactos típicos.

Filtrado espacial

A fin de mejorar la resolución espacial, se aplicó un filtro *laplaciano pequeño*, véase la [Ecuación 2.3](#), el cual resta las contribuciones de los canales alrededor de un electrodo que es relevante a la condición de estudio, en nuestro caso seleccionamos los canales FC3, C3, FC4 y C4 debido a que la activación es máxima en los canales circundantes a C3 y C4 durante la contralateralización asociada al movimiento de muñeca derecha/izquierda. A estos nuevos canales, sFC3, sC3, sFC4, sC4 se les conoce como *canales sustitutos*.

Preprocesamiento para MRCP

La señal MRCP se extrae comúnmente promediando los componentes del EEG que están bloqueados en el tiempo y la fase (conocido como *time-locked* y *phase-locked* en inglés) con el inicio del movimiento voluntario [[Savić et al., 2020](#)]. Sin embargo, como se mencionó al inicio, para el diseño de una BCI es necesario extraer características desde los MRCP de ensayo simple. De esta manera, el preprocesamiento para los MRCPs de ensayo simple consistió en:

- **Creación de épocas**

La señal EEG continua limpia se segmentó en épocas del MRCP de ensayo simple en el rango [-2, 2.5]s alrededor del inicio del estímulo de pantalla ($t = 0$). Es importante resaltar que debido a que la base de datos EEGMMIDB no cuenta con mediciones de EMG que permiten obtener el inicio de la tarea de forma más precisa, se decidió escoger el inicio del estímulo de pantalla como $t = 0$. Los datos de las épocas se corrigieron en la línea de base en todos los canales restando de cada época el valor promedio de la señal en el intervalo de -2.0 a -1.7 s. Para cada canal, se rechazaron las épocas si excedían $\pm 150 \mu\text{V}$.

Debido a que este estudio toma en cuenta como medida de rendimiento del clasificador la *precisión*, esta medida es válida únicamente si las clases están balanceadas [[Lotte et al., 2018](#)]. Por tanto, se realizó un balance de las clases, en donde en promedio se mantuvieron 21 observaciones por cada clase.

- **Filtrado temporal**

Para la extracción del MRCP de ensayo simple, se le aplicó a las épocas un filtro

pasa-bajas de tipo Butterworth de 4° orden de fase cero (filtro no-causal) con una frecuencia de corte de 2 Hz y uno pasa-altas con una frecuencia de corte de 0.1 Hz.

■ **Submuestreo**

Las épocas preprocesadas fueron submustradas a 16 Hz para facilitar el esfuerzo computacional. Como el límite de frecuencia superior de 2 Hz del paso de banda utilizado, sigue siendo inferior a 0.5 veces la frecuencia de muestreo a 5 Hz, se mantiene el criterio de Nyquist.

Las épocas generadas por el preprocesamiento MRCP fueron usadas tanto para el *análisis neurofisiológico* como para la *extracción de características*.

5.2.4. **Análisis neurofisiológico**

Además de clasificación también estuvimos interesados en analizar los correlatos neuronales MRCP entre las condiciones. Las condiciones (en nuestro caso derecha/izquierda) fueron agrupadas promediando sus valores. A fin de mostrar las diferencias estadísticamente significativas entre los correlatos MRCP de las clases de movimiento, para cada clase, se calculó un intervalo de confianza con un $\alpha = 0,05$ a través de todos los ensayos mediante el uso de una estadística *bootstrap* de t-percentil no paramétrico. Esto se realizó para cada canal individualmente. Se eligió este enfoque para encontrar una aproximación significativa de la distribución en cada punto de tiempo en un ensayo, con respecto al inicio del movimiento.

Adicionalmente, realizamos una prueba estadística por-muestra usando la prueba no paramétrica *Wilcoxon Rank-Sum*, en donde la *hipótesis nula* (H_0) fue que, para cada tipo de movimiento, los MRCPs tienen el mismo valor promedio en la misma ubicación temporal. El nivel de significancia fue establecido a $\alpha = 0,05$, con una *corrección Holm-Bonferroni* [Maris and Oostenveld, 2007] ($\alpha = 0,05/72$) para corrección de múltiples comparaciones.

5.2.5. **Extracción de características**

En el contexto de este trabajo, la extracción de características consistió en usar características desde MRCP de ensayo simple que pueden discriminar la lateralidad del movimiento. Debido a que el paradigma de la base de datos EEGMMIDB es basado

en pista y a que seleccionamos $t = 0$ como el inicio del estímulo mostrado en pantalla, el pico negativo máximo del MRCP Promedio ocurre después de $t = 0$, como se ha indicado en [Jahanshahi et al., 1995; Haynes, 2011], quizás debido al tiempo de reacción. Debido a esto, para la extracción de características tomamos como estimado del inicio del movimiento, el instante del pico negativo máximo del MRCP Promedio, aprox. en $t = 600$ ms respecto al estímulo de pantalla.

Como se mencionó en Subsección 3.0.1, aprox. 400 ms antes del inicio del movimiento, el RP aumenta repentinamente su pendiente con una mayor negatividad sobre el hemisferio contralateral a la parte del cuerpo movida. Este último componente del RP se conoce como *potencial de preparación lateralizado* (LRP). Esta propiedad del LRP sugiere que es un buen marcador neurofisiológico para predecir *qué* tipo de movimiento ocurrirá, p. ej. si la mano derecha o izquierda se moverá [Schultze-Kraft et al., 2017]. Debido a lo anterior, dadas las épocas preprocesadas para el MRCP, consideramos dos tipos de características temporales previas a $t = 600$ ms: *i*) ventanas variables de amplitudes de muestras y *ii*) basadas en la morfología del LRP, obtenidas del segmento [300, 600] ms, tales como amplitud negativa máxima (*peakmin*), promedio (*mean*), amplitud pico-pico (*peak2peak*) y pendiente e intersección de la regresión lineal (*slope* y *coef* respect.), véase Figura 5.4, las cuales son ampliamente usadas en la literatura de MRCP.

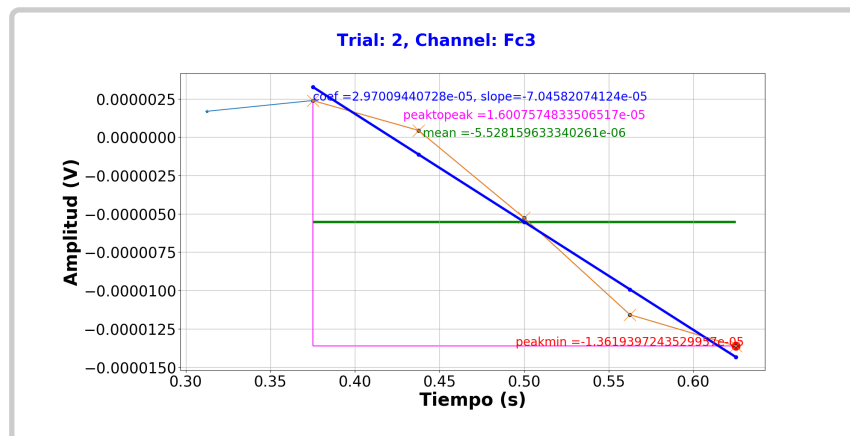


Figura 5.4: Ejemplo de las características morfológicas consideradas en un MRCP de ensayo simple

5.2.6. Selección de características

Para el caso de las características morfológicas, aplicamos un método de selección de características. Para identificar el mejor subconjunto de características, aplicamos el método estadístico univariado *coeficiente- r^2* , que es calculado como el cuadrado del *coeficiente de correlación biserial puntual* [Blankertz et al., 2011], definido como:

$$r(x) = \frac{\sqrt{N_1 N_2}}{N_1 + N_2} \frac{\text{mean}\{x_i | y_i = 1\} - \text{mean}\{x_i | y_i = 2\}}{\text{std}\{x_i | y_i = 1, 2\}}, r^2 = r \cdot r \quad (5.1)$$

donde x_i denota el valor de la característica analizada de la i -ésima muestra, y_i es la etiqueta de clase correspondiente y N_1 , N_2 los números de muestras en clase 1 y clase 2, respectivamente. El coeficiente de correlación biserial puntual es una medida de correlación entre una variable continua (observaciones de una característica) y una variable dicotómica (clases). Es importante resaltar que esta medida se basa en la suposición de que las distribuciones de clase son gaussianas [Blankertz et al., 2011]. Cuanto más alto es el valor de r^2 obtenido, mayor es la diferencia de valores promedio entre clase 1 y clase 2, y menor es la desviación estándar de todas las muestras.

Finalmente, seleccionamos las 10 características con puntuación r^2 más alta, como entradas al clasificador.

5.2.7. Selección de modelo

La *selección de modelo* es el proceso de seleccionar un modelo final de aprendizaje automático de entre una colección de modelos candidatos para un conjunto de datos de entrenamiento [James et al., 2013]. La selección de modelos es un proceso que se puede aplicar tanto en diferentes tipos de modelos (por ejemplo, regresión logística, SVM, KNN, etc.) como en modelos del mismo tipo configurados con diferentes hiperparámetros de modelo (por ejemplo, diferentes núcleos en una SVM). En este trabajo usamos el método de *validación cruzada de K -pliegues* para la selección de modelo.

Validación Cruzada

Dadas las características de los segmentos de 0 a 600 ms (indicadas en [Subsección 5.2.5](#)) con mejor puntuación r^2 , se usó una *validación cruzada de 5 pliegues repetida 5 veces* para la selección de modelo, en donde se dividió el conjunto de ensayos en datos de

entrenamiento y validación. Sin embargo, en este trabajo la validación se realizó a lo largo del ensayo completo, es decir, de -2 a 2.5 s, esto con el fin de evaluar el comportamiento de la medida de rendimiento a lo largo del tiempo, similar a como se haría en un experimento en línea. De esta manera, en cada pliegue de validación cruzada, los datos de entrenamiento fueron características obtenidas de ventanas de tiempo previas a 600 ms, mientras que los datos de evaluación fueron los datos de evaluación de cada ventana de tiempo (del mismo tamaño) que fue desplazada en pasos de 0.1 s a lo largo del ensayo, es decir, de -2 a 2.5 s. A continuación, se enuncian los pasos que se consideraron dentro de la validación cruzada:

- **Escalado de características**

El paso de escalado de características (también conocido como *normalización de datos*), consiste en estandarizar el rango de valores de las características, especialmente en aquellos casos donde los valores son muy diferentes en escala [Zaki and Meira Jr, 2019]. En este estudio aplicamos una *normalización-z*, que se define como sigue.

Sea X una característica (atributo) y sea x_1, x_2, \dots, x_n una muestra aleatoria dibujada de X (Note que incluye todas las clases!). En *normalización-z*, cada característica es reemplazada por su *z-score*:

$$x'_i = \frac{x_i - \hat{\mu}}{\hat{\sigma}} \quad (5.2)$$

donde $\hat{\mu}$ es el promedio muestra y $\hat{\sigma}$ es la desviación estándar muestra de X . Después de la transformación la nueva característica tiene promedio $\hat{\mu}' = 0$, y desviación estándar $\hat{\sigma}' = 1$.

De esta manera, las muestras de entrenamiento de todas las características fueron estandarizadas e ingresadas al clasificador. Posteriormente, las muestras de validación de todas las características fueron estandarizadas usando el promedio y desviación estándar obtenidas de los datos de entrenamiento y entonces clasificadas usando el modelo previamente entrenado.

- **Aplicación de algoritmos de aprendizaje**

Los algoritmos de aprendizaje evaluados fueron el Análisis Discriminante Lineal (LDA), Máquina de Soporte Vectorial (SVM) y Regresión Logística (LR). Debido a que se usaron los algoritmos de la biblioteca *scikit-learn* de Python, los

hiperparámetros fueron establecidos como sigue:

- LDA. La implementación de LDA (`sklearn.lda.LDA`) permite ajustar pocos parámetros, entre ellos el parámetro de contracción γ . [Blankertz et al., 2011] recomienda usar contracción cuando se tiene un gran número de características. Una ventaja de esta implementación es que permite obtener el valor de γ mediante un método analítico integrado (lema Ledoit-Wolf). De esta manera, establecimos los siguientes parámetros:
 - `shrinkage`: `auto`
- SVM. Para el caso de SVM (`sklearn.svm.SVC`) hay más flexibilidad de hiperparámetros, particularmente el *parámetro* C (que es el valor inverso del parámetro de regularización C en Ecuación 2.17), que controla el compromiso entre la penalización de las variables flojas (malas clasificaciones) y ancho de margen, el tipo de *kernel*, el cual nos permite convertir datos no lineales a datos lineales en un espacio de dimensión más alta, en este caso probamos el lineal y de base radial. Por último, γ que da la fuerza de influencia los vectores soporte. De esta manera, establecimos los siguientes parámetros:
 - `C`: 10, 1, 0.1, 0.01
 - `kernel`: `linear`, `rbf`
 - `gamma`: 0.1, 1, 10
- LR. Finalmente para LR (`sklearn.linear_model.LogisticRegression`) hay parámetros que están más allá de este estudio, sin embargo probamos el *parámetro* $C \geq 0$ (que especifica la inversa de la fuerza de regularización), que controla el compromiso entre minimizar la norma- L_2 del vector de pesos ($\|\mathbf{w}\|^2$) y la función de costo [Zaki and Meira Jr, 2019]. De esta manera, establecimos los siguientes parámetros:
 - `C`: 10, 1, 0.1, 0.01

■ Evaluación de desempeño

Se calculó la precisión y se guardó el promedio de todos los pliegues.

Finalmente, se compararon los cursos de tiempo de la precisión de clasificación para las características y clasificadores propuestos. Como en este caso, consideramos un problema de dos clases, el nivel de oportunidad teórico fue 50 %, mientras que el nivel

de oportunidad práctico (obtenido de [Ecuación 2.25](#)) de 64 % ($\alpha=5\%$) para cada sujeto (21 ensayos/clase) y de 53 % ($\alpha=5\%$) para todos los sujetos (18*21 ensayos/clase).

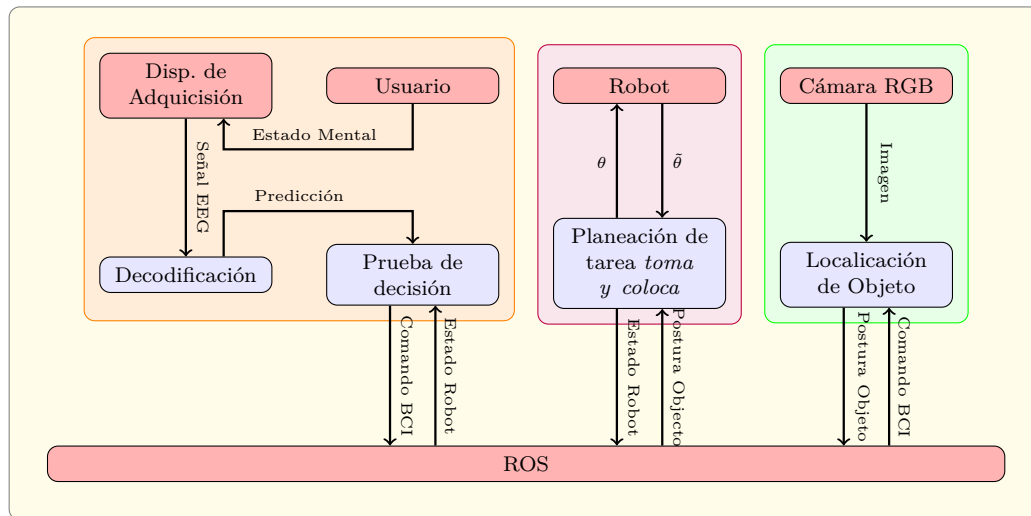


Figura 5.5: **Perspectiva general de nuestra Aplicación BCI.** El *Sistema BCI* (caja naranja) genera una señal de comando binaria (i.e., 0/1), que es detectada por el *Sistema de Visión* (caja verde), el cual calcula las posturas 2D del objeto seleccionado $(x, y, \phi)_{objeto}$ y del tazón $(x, y)_{tazon}$ y que son enviadas al *Sistema del Robot* (caja roja), en donde se genera la estrategia para ejecutar una tarea *toma y coloca* que se envía a un brazo robot de 5 GDL.

5.3. Aplicación BCI

La Aplicación propuesta para BCI se basa en la arquitectura mostrada en [Figura 5.5](#), la cual consiste de tres sistemas: *Sistema BCI* (aquí sintético), *Sistema de Visión* y el *Sistema del Robot*. Nuestro sistema fue implementado bajo Ubuntu 16.04 LTS y el *middleware Sistema Operativo del Robot* (ROS), siendo este último la base de conocimiento para intercambiar mensajes entre todos los sistemas. El prototipo físico se ilustra en [Figura 5.6a](#). A continuación se explica brevemente los sistemas de visión y del robot (véase más detalles en [Apéndice A](#)).

5.3.1. Sistema de Visión

Este sistema fue implementado con bibliotecas de OpenCV-C++ y Qt. El sistema de visión usó una cámara-RGB con una resolución de 800×600 píxeles a una velocidad de fotogramas de 30 Hz. La posición de la cámara fue arbitraria (siempre y cuando los puntos de calibración estuvieran visibles) antes de la etapa de calibración, pero después de esta etapa permanecía fija. El área de visión enfocaba una mesa de trabajo de 40×25 cm sobre la cual había tres objetos: $objetivo_1$, $objetivo_2$ y *tazón*. Los objetivos fueron

prismas cuadrangulares de 2 cm lado y 2.9 cm de alto, mientras que el tazón fue cilindro hueco de 2.9 cm de altura, 5 cm de diámetro interior y 6 cm de diámetro exterior. Así, el objetivo del sistema de visión fue calcular la postura 2D de dos objetos, véase [Figura 5.6b](#): **1**) un objetivo, que dependía del comando BCI ($0=objetivo_1=derecha$ o $1=objetivo_2=izquierda$) y **2**) el tazón. La mesa de trabajo estuvo delimitada por las esquinas externas (usadas para la etapa de calibración) de cuatro prismas cuadrangulares de la misma altura que los objetos (i.e., objetivo y tazón). La cara superior del objetivo (1 o 2) fue de color rosa, mientras que la del tazón fue de color morado. Es importante resaltar que los colores de las caras de los objetos para la tarea eran únicos y exclusivos dentro de la mesa de trabajo. El algoritmo de visión consistió principal-

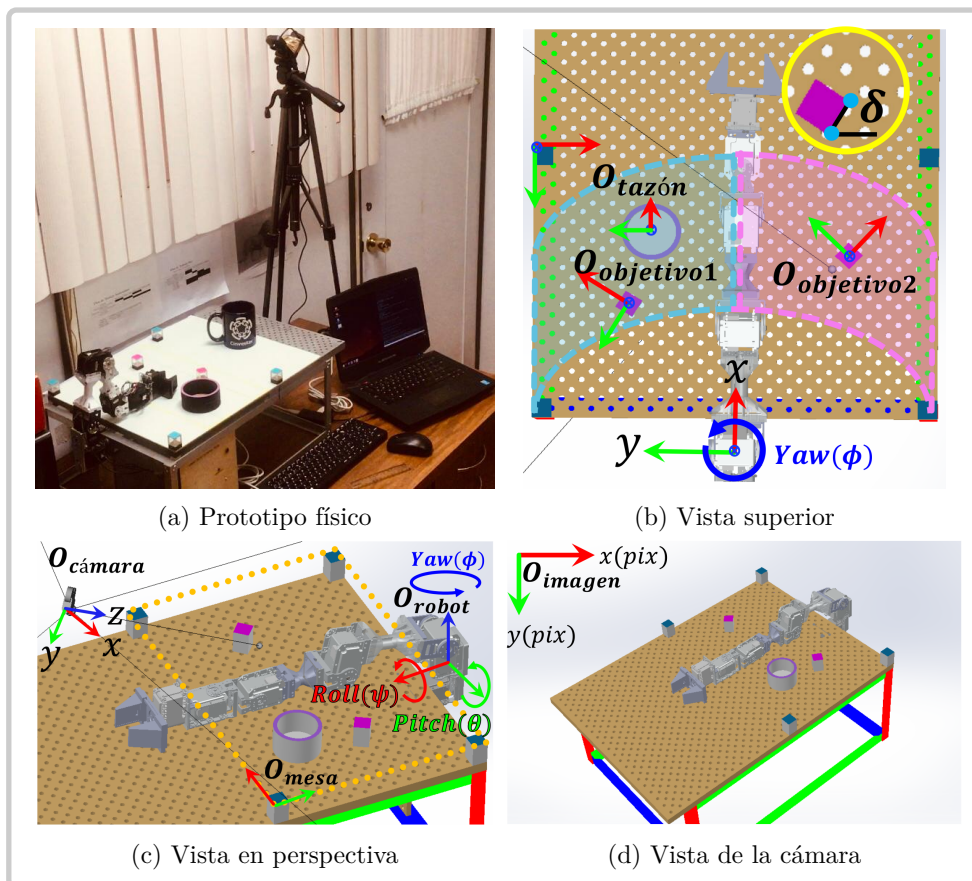


Figura 5.6: Sistema de manipulación propuesto

mente de las etapas que se indican en [Figura 5.7](#). A continuación se explica cada una de ellas.

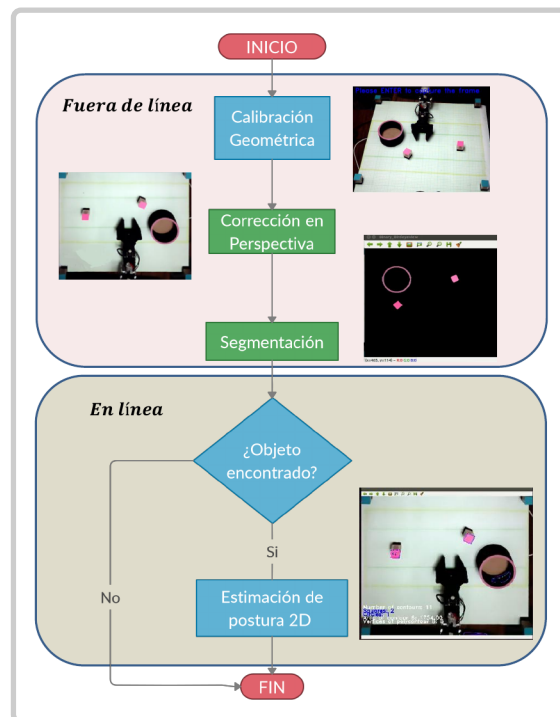


Figura 5.7: Principales etapas del algoritmo de visión artificial propuesto. Las etapas de *Análisis de Imagen* son mostradas en cajas azules, mientras que aquellas de *Procesamiento de Imagen* son indicadas en cajas verdes.

Calibración Geométrica y Corrección en perspectiva

La etapa de calibración consistió en hallar la transformación en perspectiva (*Homografía*) \mathbf{H} que nos permite convertir puntos en la imagen (pix) (véase Figura 5.6d) proveída por la cámara a sus correspondientes puntos coplanares (mm) en el mundo real (véase Figura 5.6c). Debido a que para calcular esta transformación se requieren al menos cuatro pares correspondientes de puntos, dichos puntos en el mundo fueron las esquinas externas de cuatro prismas cuadrangulares, es decir: ① (0,0), ② (400,0), ③ (400,500) y ④ (0,500). Para obtener \mathbf{H} , usamos la función `findHomography()` donde los *puntos fuente* fueron los puntos en la imagen, mientras que los *puntos destino* los puntos en el mundo.

La razón para hallar \mathbf{H} fue para obtener una imagen corregida en perspectiva, desde la cual podemos calcular la postura 2D del objetivo y tazón, i.e., $(x, y, \phi)_{\text{objeto}}$ y $(x, y)_{\text{tazón}}$. Para obtener esta imagen corregida en perspectiva, es necesario aplicar \mathbf{H} a cada punto homogéneo de la imagen y escalar como se indicó en Ecuación 4.7 y Ecuación 4.8.

Precisamente esto es lo que hace la función `WrapPerspective()`, por lo que aquí se usó. Así, ahora contamos con una imagen corregida en perspectiva en donde $1 \text{ pix} = 1 \text{ mm}$, desde la cual se pueden hacer mediciones a fin de obtener la posición de objetos sobre la mesa de trabajo.

Segmentación

Una forma de mejorar la etapa de segmentación, es aplicar previamente un emborronamiento gaussiano a la imagen, sin embargo deseamos preservar los bordes de los objetos a ser detectados. Para lograr esto usamos el filtro bilateral, el cual se implementó con la función `bilateralFilter()`, en donde los siguientes parámetros fueron sintonizados experimentalmente: *longitud de la vecindad de píxel* = 20, $\sigma_r = 40$, $\sigma_s = 10$. Luego, desde la imagen RGB seleccionamos manualmente los dos colores de los objetos y los transformamos a HSV, en donde se puede segmentar (i.e., aislar colores) más fácilmente. Para llevar a cabo esto, usamos la función `inRange()` en donde para cada color se establecieron los umbrales como $H_L = H - 15$, $H_H = H + 15$, $S_L = S - 15$, $S_H = S + 15$, $V_L = V - 15$, $V_H = V + 15$. Para remover el ruido de *sal y pimienta* de la imagen binaria resultante, se aplicó un filtro mediana, usando la función `medianBlur()` con un tamaño de vecindad de $k = 7$ píxeles.

Detección

Dada la imagen binaria que segmentó los colores que corresponden al objetivo (1 o 2) y al tazón, se realizó una detección de contornos usando la función `findContours()`. Para identificar que área corresponde a que objeto, se determinó el número de vértices de cada contorno usando la función `approxPolyDP()`, en donde 4 vértices indicaba el objetivo (1 o 2) y mayor a 4 vértices indicaba tazón, sin embargo en tiempo real p. ej. para el cuadrado no siempre generaba 4 vértices, principalmente debido a las condiciones de luz. Para lidiar con esto, previo a la detección de vértices se discriminó por el área de los contornos usando la función `contourArea()`, en donde de 30000 a 35000 pix^2 indicaba el tazón, mientras que de 1000 a 5000 pix^2 representaba el objetivo (1 o 2).

Estimación de Postura 2D

Si se encontró el objetivo (1 o 2) y el tazón se procedió a calcular su postura 2D. Recordemos que en esta imagen 1 pix equivale a 1 mm. De esta manera, dadas las imágenes binarias de los contornos aproximados de ambos objetos, la posición (x, y) de ambos objetos en el referencial del mundo, se puede determinar simplemente midiendo su centroide. Para esto, hacemos uso de los momentos de la imagen que son definidos como promedios ponderados de las intensidades de los píxeles en la imagen. Así, el centroide queda definido como:

$$(x, y) = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) = \left(\frac{\sum_{x,y} xI(x, y)}{\sum_{x,y} I(x, y)}, \frac{\sum_{x,y} yI(x, y)}{\sum_{x,y} I(x, y)} \right) \quad (5.3)$$

donde $I(x, y)$ es la intensidad del píxel y $m_{i,j}$ es el momento $i + j$ de la imagen. Para la implementación usamos la función `moments()`, cuya entrada puede ser las coordenadas del contorno o los puntos internos de un objeto.

Para encontrar la orientación del objetivo (1 o 2), se puede usar también los momentos de imagen, sin embargo en este trabajo decidimos usar la función `minAreaRect.angle()`, que devuelve el ángulo δ de un rectángulo rotado que encierra al contorno del objetivo (1 o 2) (véase [Figura 5.6b](#)), debido a que con este podemos determinar directamente el ángulo *yaw* ϕ (véase [Algoritmo 1](#)) requerido por el robot para llevar a cabo la tarea.

Algoritmo 1: Orientación del objetivo (1 o 2) en el referencial del robot

Entrada: Ángulo obtenido de la función `minAreaRect.angle()` de OpenCV: δ

Salida: Ángulo *yaw* para robot: $\phi_{objetivo}$

```

1 si  $y_{objetivo} < 0$  entonces
2 |  $\phi_{objetivo} = - (90 - |\delta|)$ 
3 en otro caso
4 |  $\phi_{objetivo} = |\delta|$ 
5 fin
```

5.3.2. Sistema del Robot

Este sistema fue implementado con bibliotecas de Python y Qt, y hace uso de un robot de 5 GDL con articulaciones rotacionales (véase [Figura 5.8a](#)), que para propósitos de este trabajo, las tres primeras articulaciones sirven para posicionar la articulación rotacional 4, en la cual su eje x siempre se busca mantener paralelo a la mesa y

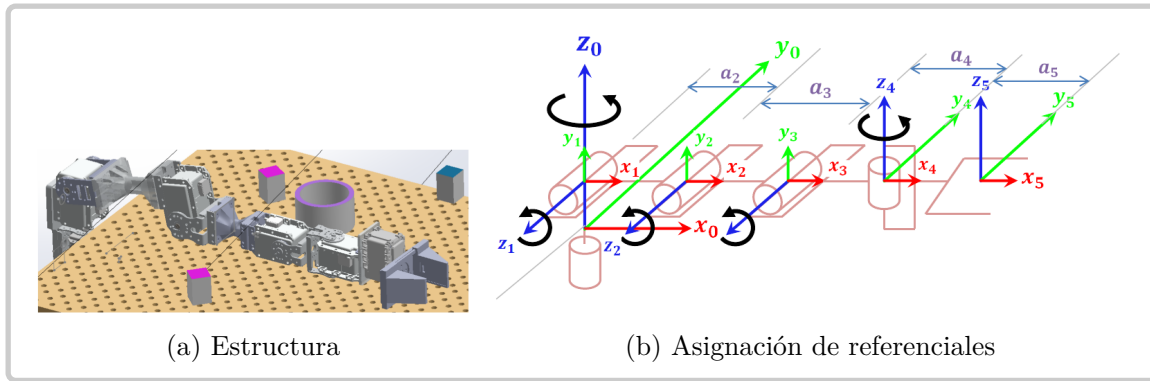


Figura 5.8: Brazo robótico de 5 GDL

finalmente la articulación 5 que genera la orientación necesaria de la pinza para tomar el objetivo (1 o 2). Así, el objetivo de este sistema fue: Dada la postura 2D del objetivo (1 o 2) $(x, y, \phi)_{objetivo}$ y el tazón $(x, y)_{tazón}$ en el referencial del robot O_{robot} , se busca generar el camino para llevar a cabo una tarea *toma y coloca* por el robot antes mencionado.

Modelo Cinemático Directo

Con el fin de calcular el espacio de trabajo del robot y desarrollar un simulador del robot, realizamos el cálculo de la cinemática directa del manipulador, que consiste en determinar la relación entre la postura del efector final del robot dadas sus coordenadas articulares, i.e., $\mathbf{x} = f(\mathbf{q})$. Para esto, usamos la convención de *Denavit-Hartenberg* (D-H), la cual hace uso de matriz de *transformación homogénea* (TH), $\mathbf{T}_A^B \in \mathbb{R}^{4 \times 4}$, que expresa la postura del referencial B con respecto al referencial A . En general, para dos articulaciones consecutivas se tiene que [Spong et al., 2006]:

$$\begin{aligned}
 T_{i-1}^i &= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a_x & s_x & n_x & p_x \\ a_y & s_y & n_y & p_y \\ a_z & s_z & n_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{a} & \mathbf{s} & \mathbf{n} & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \mathbf{R}_{i-1}^i & \mathbf{p}_{i-1}^i \\ \mathbf{0}^T & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{5.4}$$

donde θ_i, a_i, d_i y α_i son parámetros asociados con el eslabón i , conocidos como *ángulo de articulación*, *longitud del eslabón*, *desplazamiento de eslabón* y *torsión del eslabón*,

Eslabón	a_i	α_i	d_i	θ_i
	distancia a lo largo de x_i desde la intersección de los ejes x_i y z_{i-1} a o_i	el ángulo desde z_{i-1} a x_i medido alrededor de x_i	distancia a lo largo de z_{i-1} desde o_{i-1} a la intersección de los ejes x_i y z_{i-1} . Si la articulación i es prismática, d_i es variable.	el ángulo desde x_{i-1} a x_i medido alrededor de z_{i-1} . Si la articulación i es revoluta, θ_i es variable
1	0 cm	90°	4 cm	θ_1^*
2	13.0 cm	0°	0 cm	θ_2^*
3	11.7 cm	0°	0 cm	θ_3^*
4	5.0 cm	-90°	0 cm	θ_4^*
5	11.5 cm	0°	0 cm	θ_5^*

Tabla 5.1: Parámetros D-H del robot

y en general se conocen como los *parámetros D-H*. Finalmente, para el caso de un manipulador con eslabones en serie, el referencial n se describe con respecto al referencial 0 mediante la composición de transformaciones entre referenciales consecutivos.

De esta manera, primeramente realizamos la asignación de referenciales de acuerdo a la convención D-H [Spong et al., 2006], como se ilustra en Figura 5.8b. Luego, calculamos los parámetros D-H que se indican en Tabla 5.1 y obtuvimos las THs consecutivas entre los referenciales. Finalmente, la TH que relaciona la postura del efector final con el referencial base está descrita por:

$$\begin{aligned}
T_0^5 &= T_0^1 T_1^2 T_2^3 T_3^4 T_4^5 \\
&= \begin{bmatrix} -s_1 s_5 + c_1 c_5 c_{234} & -s_1 c_5 - s_5 c_1 c_{234} & -s_{234} c_1 & -11,5 s_1 s_5 + 13,0 c_1 c_2 + 11,5 c_1 c_5 c_{234} + 11,7 c_1 c_{23} + 5 c_1 c_{23} \\ s_1 c_5 c_{234} + s_5 c_1 & -s_1 s_5 c_{234} + c_1 c_5 & -s_1 s_{234} & 13,0 s_1 c_2 + 11,5 s_1 c_5 c_{234} + 11,7 s_1 c_{234} + 5 s_1 c_{234} + 11,5 s_5 c_1 \\ s_{234} c_5 & -s_5 s_{234} & c_{234} & 13,0 s_2 + 11,7 s_{23} + 11,5 s_{234} c_5 + 5 s_{234} + 4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.5)
\end{aligned}$$

Modelo Cinemático Inverso

Con el objetivo de minimizar el número de parámetros para describir la orientación del referencial de la pinza, se usaron los ángulos XYZ de Cardan, mejor conocidos como los ángulos *roll* (ψ), *pitch* (θ) y *yaw* (ϕ), los cuales definen una orientación en el espacio por tres rotaciones sucesivas alrededor de un referencial base: primero una rotación alrededor de x_0 un ángulo ψ (*roll*), luego una rotación alrededor de y_0 un ángulo θ (*pitch*), y finalmente una rotación alrededor de z_0 un ángulo ϕ (*yaw*), es decir:

$$R_{ZYX} = R_{z_0}(\phi) R_{y_0}(\theta) R_{x_0}(\psi) \quad (5.6)$$

Ahora, podemos proceder a declarar el problema de cinemática inversa (CI) del robot, el cual consiste en encontrar las variables articulares dada la postura del efector final, i.e., $\mathbf{q} = f^{-1}(\mathbf{x})$. En nuestro caso, debido a que la pinza debe tomar el objetivo (1 o 2) lateralmente, el problema de CI se reduce a lo siguiente: Dada la postura deseada del referencial de la pinza (que corresponde a la postura del objeto a ser tomado) en la representación mínima dada por:

$$\mathbf{x}_d = [x \quad y \quad z \quad \psi_{x_0} = 0 \quad \theta_{y_0} = 0 \quad \phi_{z_0}]^\top \quad (5.7)$$

se busca una solución cerrada, \mathbf{q} , tal que genere movimientos físicamente realizables, sin colisionar con la mesa de trabajo. De [Ecuación 5.6](#), esto equivale a: Dada la postura deseada del referencial de la pinza expresada por la TH:

$$T_{d0}^5 = \begin{bmatrix} \mathbf{R}_{d0}^5 & \mathbf{p}_{d0}^5 \\ \mathbf{0}^\top & 1 \end{bmatrix} = \begin{bmatrix} c_\phi & -s_\phi & 0 & p_x \\ s_\phi & c_\phi & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{d0}^5 & \mathbf{s}_{d0}^5 & \mathbf{n}_{d0}^5 & \mathbf{p}_{d0}^5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.8)$$

se desea hallar \mathbf{q} .

Para resolver el problema de CI, usamos el *método geométrico*, para el cual fue necesario el bosquejo de diversas vistas, véase [Figura 5.9](#). Primeramente notemos que la posición del referencial de la articulación 5 (aquí denotada como w_2) está dado por:

$$\mathbf{p}_{w_2} = \mathbf{p}_{d0}^5 - a_5 \mathbf{a}_{d0}^5 \quad (5.9)$$

Así, de [Figura 5.9a](#) se puede observar que θ_1 está definido por [Ecuación 5.10a](#).

$$\theta_1 = \text{atan2} \left(\frac{p_{w_2y}}{p_{w_2x}} \right) \quad (5.10a)$$

$$\theta_2 = \alpha \pm \beta = \text{atan2} \left(\frac{s}{r} \right) \pm \text{atan2} \left(\frac{a_3 \sin \theta_3}{a_2 + a_3 \cos \theta_3} \right) \quad (5.10b)$$

$$\theta_3 = \pm 2 \text{atan2} \left(\frac{\sqrt{(a_2 + a_3)^2 - (r^2 + s^2)}}{\sqrt{(r^2 + s^2) - (a_2 - a_3)^2}} \right) \quad (5.10c)$$

$$\theta_4 = -(\theta_2 + \theta_3) \quad (5.10d)$$

$$\theta_5 = \phi - \theta_1 \quad (5.10e)$$

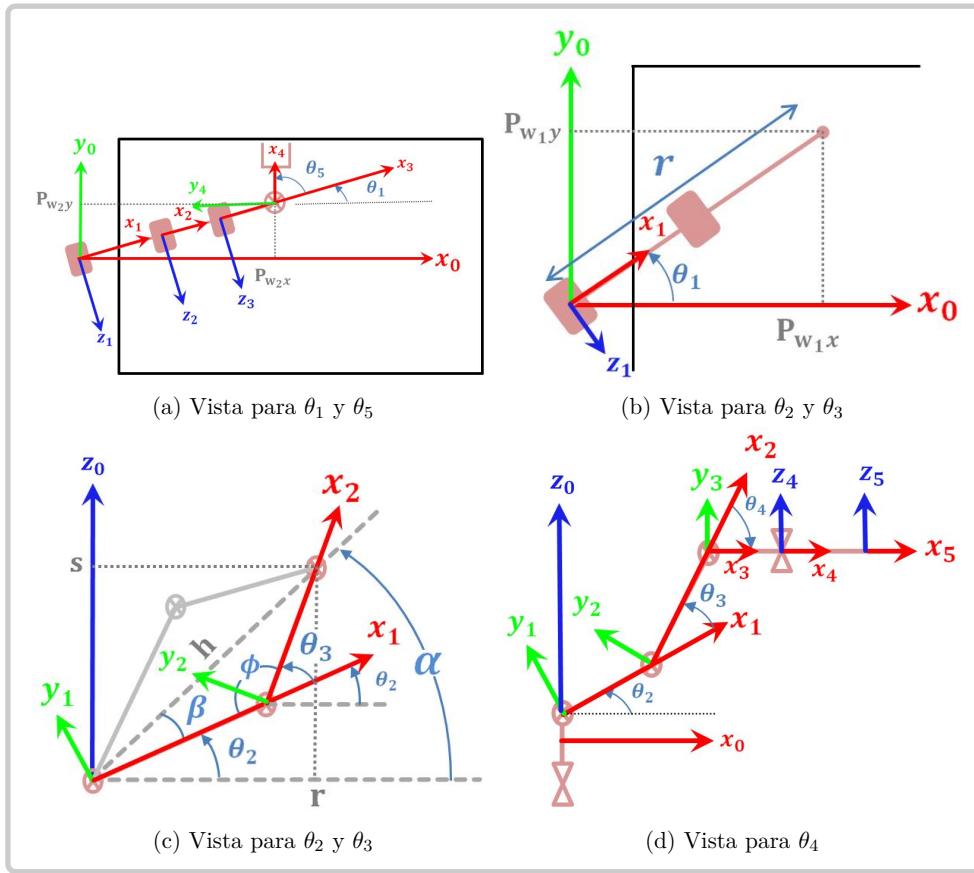


Figura 5.9: Geometrías para resolver la cinemática inversa del robot

Además, debido a que:

$$\mathbf{R}_{d_0^4} = R_{z_0}(\phi - \theta_5) = R_{z_0}(\theta_1) = \begin{bmatrix} c_{\theta_1} & -s_{\theta_1} & 0 \\ s_{\theta_1} & c_{\theta_1} & 0 \\ 0 & 0 & 1 \end{bmatrix} = [\mathbf{a}_{d_0^4} \quad \mathbf{s}_{d_0^4} \quad \mathbf{n}_{d_0^4}] \quad (5.11)$$

la posición del referencial de la articulación 4 (aquí denotada como w_1) es:

$$\mathbf{p}_{w_1} = \mathbf{p}_{w_2} - a_4 \mathbf{a}_{d_0^4} \quad (5.12)$$

De esta manera, conocido \mathbf{p}_{w_1} , se pueden obtener θ_2 y θ_3 como un caso planar. La Figura 5.9b muestra la proyección de \mathbf{p}_{w_1} sobre el plano $x_0 - y_0$, de donde podemos definir $r = +\sqrt{p_{w_1x}^2 + p_{w_1y}^2}$ y la Figura 5.9c muestra la misma posición pero en el plano $x_0 - z_0$ de donde se puede deducir que $s = p_{w_1z} - d_1$. Usando la

ley de cosenos $h^2 = a_2^2 + a_3^2 - 2a_2a_3\cos\phi$, así como las propiedades trigonométricas $\cos(a - b) = \cos(a)\cos(b) + \sin(a)\sin(b)$ y $\tan\frac{\theta}{2} = \pm\sqrt{\frac{1-\cos\theta}{1+\cos\theta}}$ se puede deducir que θ_2 y θ_3 están definidos por la Ecuación 5.10b y Ecuación 5.10c respectivamente, y en donde seleccionamos el signo '-' para la configuración *codo arriba*. Además, debido a que para propósitos de esta aplicación, la articulación 5 contribuye a la orientación ϕ (*yaw*), esta no debería moverse cuando la articulación 4 se mueve, por lo que θ_4 queda descrito por Ecuación 5.10d (véase Figura 5.9d). Finalmente, de Figura 5.9a, note que θ_5 queda definido como Ecuación 5.10e.

Control del Robot Manipulador

Como se observa en Figura 5.8a, el robot propuesto consiste de 5 articulaciones y una pinza. Esta configuración, permite al manipulador tener la habilidad de agarrar los prismas cuadrangulares lateralmente. Debido a que el robot usado no es un robot comercial y a fin de mantener la simplicidad, en este trabajo cada uno de los actuadores del robot, Dynamixel MX-28/MX-64, fue controlado en *Modo Articulación* (posición). En el *Modo Articulación*, los actuadores Dynamixel se ejecutan en función de sus consignas de posiciones con límites de posición establecidos en el archivo de configuración. Es decir, en este trabajo usamos una arquitectura de control cinemática, como la mostrada en Figura 5.10, en donde hay un control independiente por cada articulación (1-5). Debido a que el robot se moverá lentamente, las aceleraciones serán pequeñas, por lo que el par resistente de inercia se puede considerar despreciable. Además, debido a la estructura ligera de este, el par resistente debido al efecto de la gravedad es pequeño y para propósitos de este trabajo no se toma en cuenta.

Dentro de cada encapsulado de los actuadores *Dynamixel* está integrado un controlador tipo PID, que se puede ajustar con los registros P_{gain} , I_{gain} y D_{gain} mediante las ecuaciones en Ecuación 5.13.

$$K_p = P_{gain}/8 \quad (5.13a)$$

$$K_i = (I_{gain} * 1000)/2048 \quad (5.13b)$$

$$K_d = (D_{gain} * 4)/1000 \quad (5.13c)$$

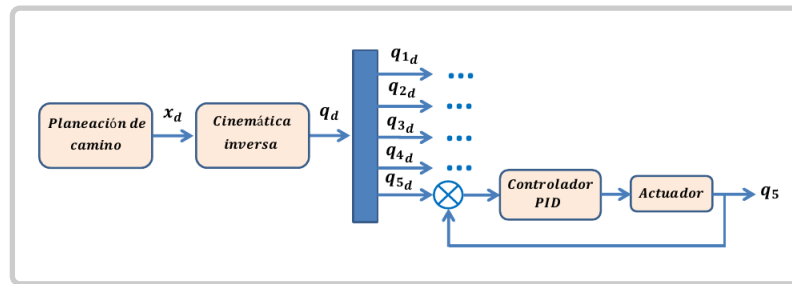


Figura 5.10: **Arquitectura de control del robot manipulador.** Dado una secuencia de coordenadas en el espacio cartesiano, mediante cinemática inversa se obtienen las coordenadas articulares deseadas de cada una de las articulaciones que integran al robot, y que son alcanzadas por cada controlador independientemente.

La sintonización de ganancias se realizó de forma heurística, sin embargo al final se notó que las ganancias con las que se logra un buen compromiso entre tiempo de subida y sobrepicos, es usando las ganancias por defecto del servomotor, dadas por $K_p = 4$ ($P_{gain} = 32$), $K_i = 0$ ($I_{gain} = 0$), $K_d = 0$ ($D_{gain} = 32$), esencialmente un controlador proporcional. A continuación se describe brevemente los pasos de la arquitectura de control en [Figura 5.10](#).

Primeramente, debido a que la postura 2D recibida desde el sistema de visión (en cm) está en el marco de referencia de la mesa, esta es transformada al marco de referencia del robot mediante:

$$T_{robot}^{mesa} = \begin{bmatrix} 0 & -1 & 0 & 55 \\ -1 & 0 & 0 & 20 \\ 0 & 0 & -1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.14)$$

A continuación, se definió una serie de puntos cartesianos clave para ejecutar la tarea de *toma y coloca*, dados por [Algoritmo 2](#). Cada punto de la estrategia en su representa-

Algoritmo 2: Tarea *toma y coloca*

Entrada: x_{objeto} , y_{objeto} , ϕ_{objeto} , $x_{tazón}$, $y_{tazón}$

Salida: Tarea cumplida

- 1 **mientras** *objeto y tazón sean detectados* **hacer**
 - 2 1) Ir a (x_{objeto} , y_{objeto} , 15, ϕ_{objeto})
 - 3 2) Ir a (x_{objeto} , y_{objeto} , 3,5, ϕ_{objeto}) y cierra la pinza
 - 4 3) Ir a (x_{objeto} , y_{objeto} , 15, ϕ_{objeto})
 - 5 4) Ir a ($x_{tazón}$, $y_{tazón}$, 15, 0) y abre la pinza
 - 6 5) Ve a la posición inicio del robot
 - 7 **fin**
-

ción mínima se transformó a su transformación homogénea respectiva. Posteriormente, una trayectoria continua entre cada postura de la estrategia de movimiento se logró mediante la aplicación de algoritmo de interpolación lineal esférica (SLERP) codificado en la función `ctraj` de la biblioteca de *Python Robotics*. En seguida, se procedió a obtener las coordenadas articulares requeridas por el robot mediante cinemática inversa. Finalmente, cada consigna de posición articular fue ingresada al correspondiente actuador del robot.

5.3.3. Base de conocimiento: ROS

El *Sistema Operativo del Robot* (ROS) es un marco de *Middleware* de licencia abierta que proporciona una infraestructura común para escribir software de robots. De hecho, las ventajas más sobresalientes de ROS es su modularidad y su comunidad de desarrollo en aumento, pues es cualquier persona puede diseñar, implementar y distribuir sus entidades ROS conocidas como *paquetes*. ROS tiene la habilidad de dividir la estructura en pequeños módulos llamados *nodos*, que pueden comunicarse entre ellos mediante un *nodo maestro*. Un nodo publica datos bajo un *tópico* y otros nodos pueden usar estos datos subscribiéndose a tal tópico.

Para el presente trabajo se utilizó la versión *ROS-Kinetic*, en donde se desarrollaron 4 paquetes: 1) *Paquete Robot*, en donde se establecía los parámetros y consignas a los actuadores *Dynamixel*, 2) el *Paquete Visión*, el cual determinaba la postura 2D de los objetos sobre la mesa de trabajo y la enviaba al robot mediante un mensaje personalizado, 3) el *Paquete Control*, el cual transformaba la postura 2D de los objetos a coordenadas en el referencial del robot y con la cual se diseñaba la estrategia para ejecutar la tarea *toma y coloca* y finalmente 4) el *Paquete BCI Sintética*, el cual emulaba un comando sintético para indicar estados mentales que para propósitos de este trabajo fueron movimiento DERECHA/IZQUIERDA, siempre y cuando el robot no estuviera en estado ocupado.

Capítulo 6

Resultados

En esta sección se presenta los resultados obtenidos de la decodificación de lateralidad usando las características de Potencial de Preparación (RP). Primeramente, se muestra el análisis neurofisiológico que permite identificar las características potenciales. Posteriormente, se muestran los cursos de tiempo de la precisión a lo largo del ensayo, usando ventanas de tiempo de diferentes tamaños. Finalmente, se muestra el resultado de seguimiento de posición del sistema de manipulación propuesto durante la ejecución de la tarea *toma y coloca*.

6.0.1. Análisis Neurofisiológico

El MRCP de Gran-Promedio para las condiciones de estudio (*derecha/ izquierda*) se ilustra en la [Figura 6.1](#). En esta se puede observar que este potencial eléctrico sobre estos canales muestra el patrón típico que emerge en un paradigma síncrono. Aproximadamente en $t = 300$ ms después de la presentación del estímulo en pantalla (flecha), se presentó un pico positivo, quizás debido al proceso cognitivo del estímulo. En seguida, ocurrió un pico negativo aprox. en $t = 0,56$ s que atribuimos al inicio de movimiento y relacionado con el pico negativo del MRCP.

Se puede observar que los intervalos de confianza ($\alpha = 0,05$) no se traslapan durante el pico de negatividad máxima en los canales sFC3 y sC3, indicando significancia estadística. Adicionalmente, de la prueba *Wilcoxon Rank-Sum* ($p < 0,05$) muestra-a-muestra, se verificó que hay diferencias significativas entre ambas condiciones de estudio (es decir, *derecha* e *izquierda*) en los canales sFC3, sC3 y en sFC4 (franjas grises). En la [Figura 6.2](#) se muestra un mapa *tiempo-canal* que resume el *puntaje-z* de todas las prue-

bas estadísticas, indicando los puntos de significancia estadística (cuadros rojos). De esta manera, podemos decir que el *potencial de preparación lateralizado* (LRP) hacia el área motora contralateral estuvo presente en los canales sFC3, sFC4 y sC3, y por lo tanto este componente del MRCP contiene información discriminativa sobre la clase de movimiento, es decir, la lateralidad.

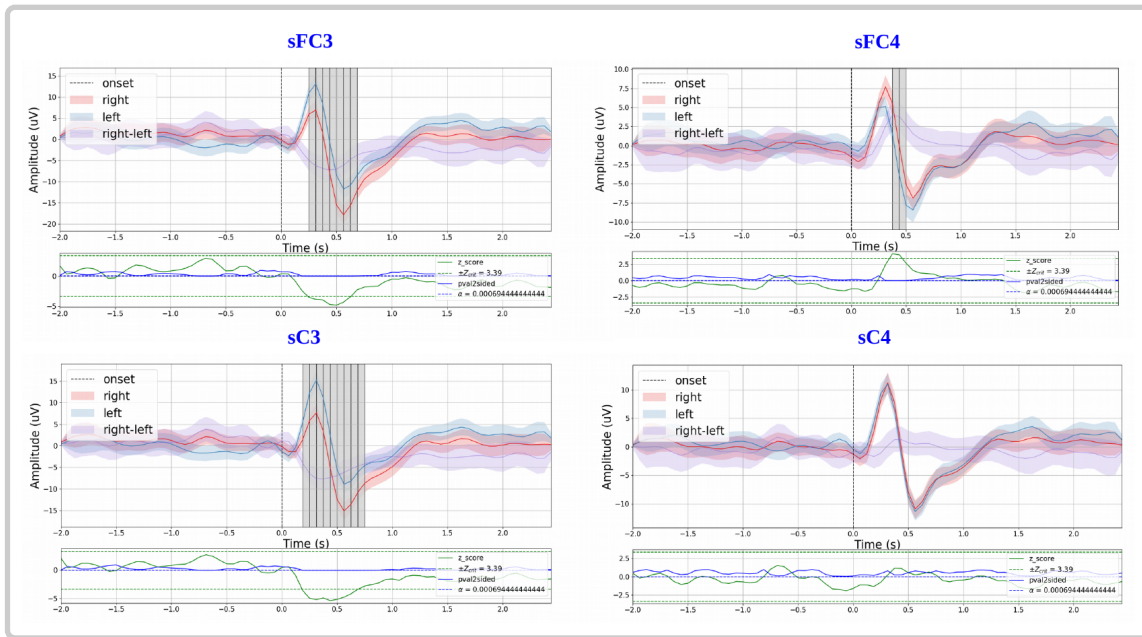


Figura 6.1: **Análisis neurofisiológico de movimiento de mano derecha/izquierda.** Se muestra el MRCP de Gran-Promedio en los canales sustitutos sFC3, sC3, sFC4 y sC4, promediado para 18 sujetos (derecha=383, izquierda=383) y respecto al inicio del estímulo de pantalla. Las áreas coloreadas muestran el intervalo de confianza con $\alpha = 0,05$. En cada canal se indica con franjas grises los puntos de tiempo significativamente diferentes ($p < 0,05$) resultado de la prueba *Wilcoxon Rank-Sum* con una *corrección Holm-Bonferroni*. En la parte inferior de cada canal se muestran los *valores-p* de las pruebas estadísticas.

6.0.2. Selección de características

Para el caso de las características morfológicas, se llevó a cabo una selección de características usando el *coeficiente- r^2* , cuyos puntajes se muestran en el mapa de calor de la [Figura 6.3](#), promediados para todos los sujetos y movimientos. Notemos que este método de selección de características univariado nos sugiere que las mejores características morfológicas del MRCP propuestas son el pico mínimo y el promedio, principalmente en los canales sC3, sFC3 y sFC4, los cuales coinciden con los canales significativos del

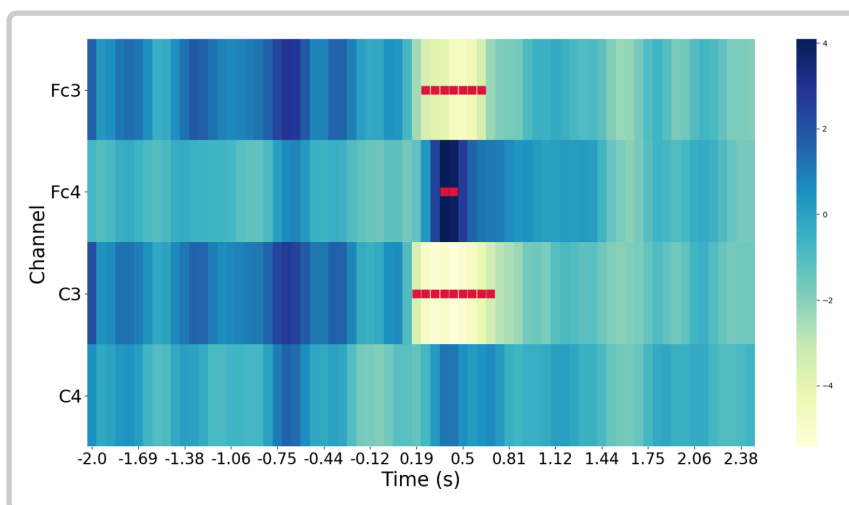


Figura 6.2: Mapa de calor de *puntaje-z* de las pruebas estadísticas *Wilcoxon Rank-Sum*. Aquellos puntos que indicaron significancia estadística se marcan con cuadros rojos.

análisis neurofisiológico del MRCP previamente descrito.

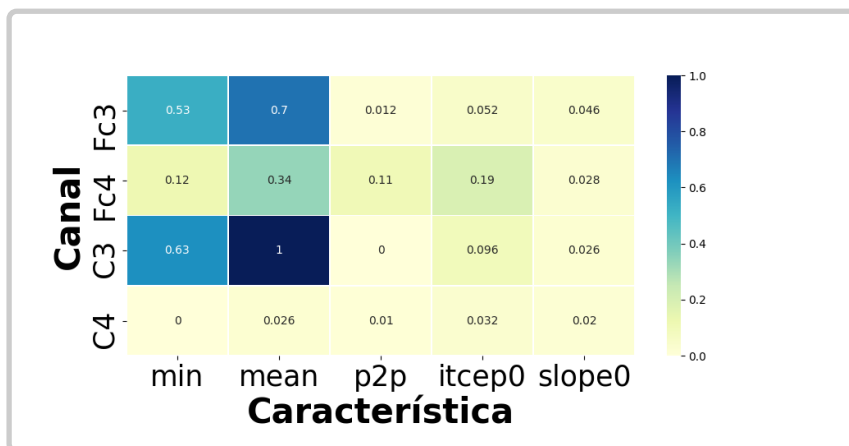


Figura 6.3: *coeficiente- r^2* para las características morfológicas propuestas del MRCP. La representación indica el promedio de todos los sujetos y movimientos sobre todos los pliegues de validación cruzada.

6.0.3. Clasificación Binaria de ensayo simple

En este estudio, la clasificación se realizó usando una ventana de tiempo de tamaño $[t - t_0, t]$, donde t_0 fue evaluado en $t_0 \in \{0, 1, 0, 6, 1, 2, 1, 3, 1, 4, 1, 7, 2, 0\}$, que fue desplazada a lo largo de los ensayos de todos los sujetos de estudio, y en donde la precisión

de clasificación se alineó a la izquierda de esta ventana de tiempo (i.e., una ventana de extracción de características causal). En la [Figura 6.4](#) se muestra la precisión de gran promedio como función del tiempo para las características de muestras, en donde se puede observar que las precisiones de clasificación comienzan a saturarse con ventanas de extracción de características arriba de 1 s, y en donde la precisión máxima de 71.39 % se alcanza en $t = 0,69$ s, un tamaño de ventana de 1.3 s y usando el clasificador SVM lineal. Por otra parte, en la [Figura 6.5](#) se ilustra la precisión pero ahora usando las características morfológicas obtenidas de la ventana fija de 0.3 s, en donde la precisión de gran promedio máxima de 67.18 % en $t = 0,63$ s se obtiene también con el algoritmo de SVM lineal. Es importante resaltar que el instante del pico de precisión concuerda con aquel en donde se presentó diferencia estadística entre el MRCP de **derecha vs izquierda**, como se indica en la [Figura 6.1](#). El nivel de oportunidad para 2 clases es 50 % y el nivel de oportunidad empírico fue determinado como ~ 53 % ($\alpha = 5$ %, intervalo Wald Ajustado), el cual fue superado en ~ 0.3 s.

En la [Tabla 6.1](#) se resume los mejores resultados obtenidos de la validación cruzada para la clasificación binaria (**derecha vs izquierda**) con las características y clasificadores propuestos. Las matrices de confusión en la [Tabla 6.1](#) calculadas en el pico de la precisión de gran promedio, muestran que para las características tomadas de muestras, ambas clases son discriminables cuasi-equitativamente. Sin embargo, usando las características morfológicas se obtiene una menor precisión para la clase **derecha**. Así, de la [Tabla 6.1](#) y de la dispersión de tales precisiones indicadas en la [Figura 6.6](#), se puede observar que esta medida es mayor (en los tres clasificadores) al usar las características de muestras que usando las morfológicas, y que con el clasificador SVM lineal con $C=0.01$ se obtiene la precisión de gran promedio más alta de 71.38 % (STD= ± 3.92) en $t = 0,69$ s.

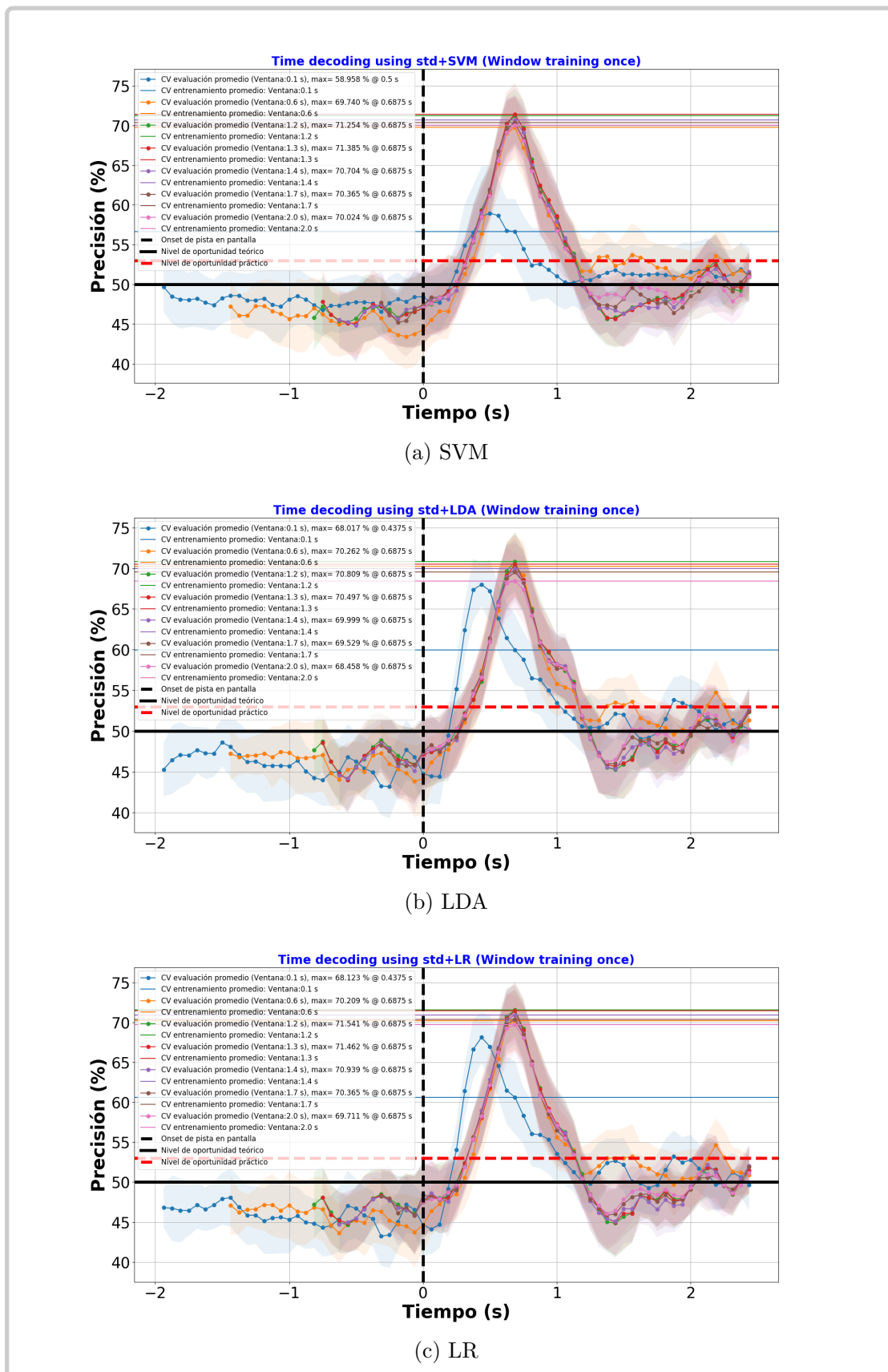


Figura 6.4: Precisión de clasificación de gran promedio usando características de muestras. Las áreas coloreadas indican la desviación estándar, mientras que cada color se asocia con un tamaño de ventana.

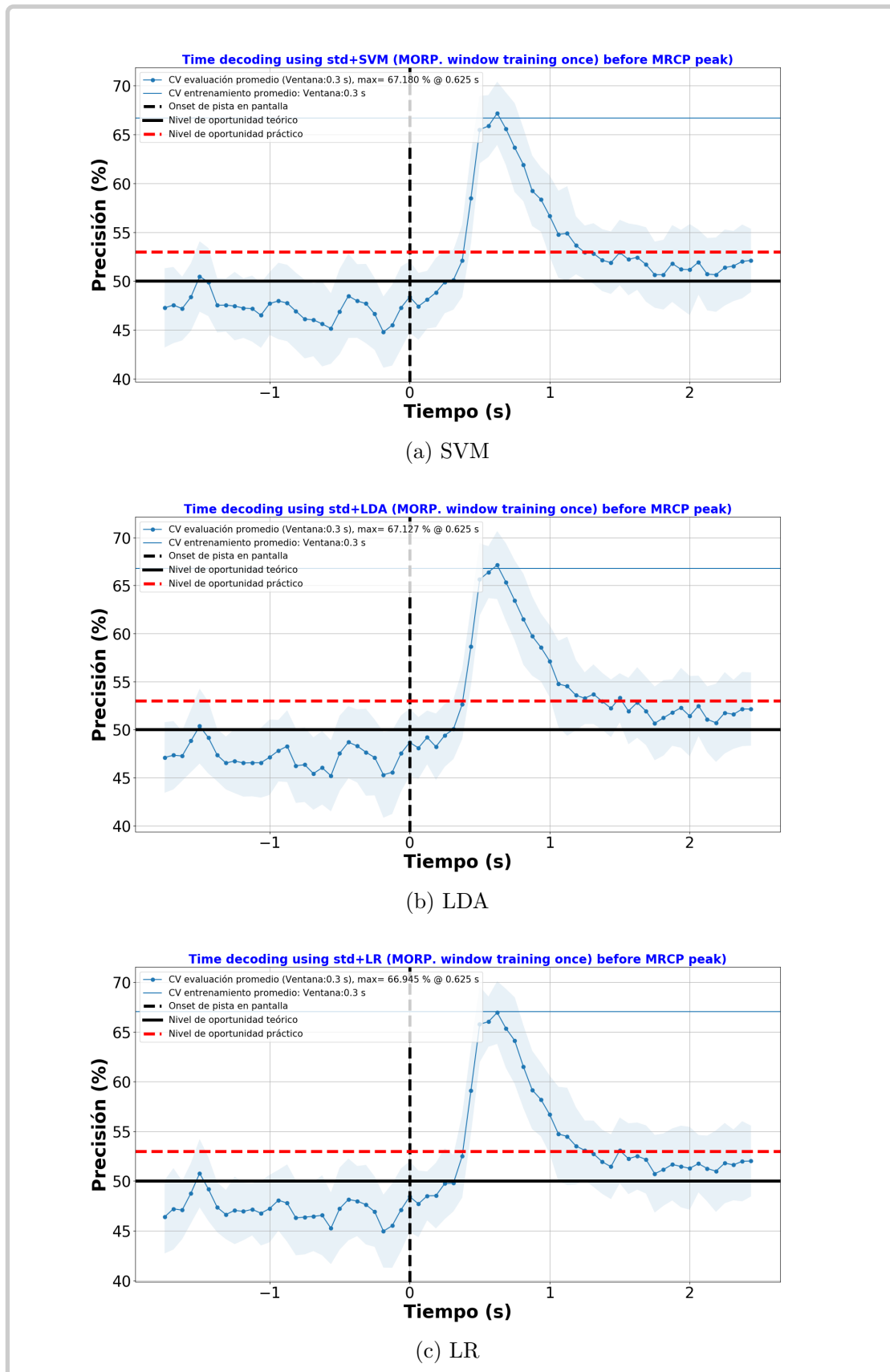


Figura 6.5: Precisión de clasificación de gran promedio usando características morfológicas. Las áreas coloreadas indican la desviación estándar.

		Clasificador																																																
		SVM	LDA	LR																																														
Características	Valores de muestras	Hiperparámetros	kernel='linear', C=0.01	solver='lsqr', shrinkage='auto'	solver='lbfgs', C=0.1																																													
		Tamaño de ventana	1.3 s	1.2 s	1.2 s																																													
		Matriz de Confusión Pico normalizada	<table border="1"> <tr><td colspan="2"></td><td colspan="2">Predicho</td></tr> <tr><td colspan="2"></td><td>I</td><td>D</td></tr> <tr><td rowspan="2">Real</td><td>I</td><td>0.73</td><td>0.27</td></tr> <tr><td>D</td><td>0.30</td><td>0.70</td></tr> </table>			Predicho				I	D	Real	I	0.73	0.27	D	0.30	0.70	<table border="1"> <tr><td colspan="2"></td><td colspan="2">Predicho</td></tr> <tr><td colspan="2"></td><td>I</td><td>D</td></tr> <tr><td rowspan="2">Real</td><td>I</td><td>0.71</td><td>0.29</td></tr> <tr><td>D</td><td>0.30</td><td>0.70</td></tr> </table>			Predicho				I	D	Real	I	0.71	0.29	D	0.30	0.70	<table border="1"> <tr><td colspan="2"></td><td colspan="2">Predicho</td></tr> <tr><td colspan="2"></td><td>I</td><td>D</td></tr> <tr><td rowspan="2">Real</td><td>I</td><td>0.72</td><td>0.28</td></tr> <tr><td>D</td><td>0.29</td><td>0.71</td></tr> </table>			Predicho				I	D	Real	I	0.72	0.28	D	0.29	0.71
				Predicho																																														
				I	D																																													
		Real	I	0.73	0.27																																													
	D		0.30	0.70																																														
			Predicho																																															
			I	D																																														
	Real	I	0.71	0.29																																														
		D	0.30	0.70																																														
			Predicho																																															
		I	D																																															
Real	I	0.72	0.28																																															
	D	0.29	0.71																																															
Precisión Pico (%)	71.38, STD= \pm 3.92	70.81, STD= \pm 3.64	71.54, STD= \pm 3.25																																															
Punto de tiempo Pico (relativo al inicio de pista en pantalla)	0.69 s	0.69 s	0.69 s																																															
Morfológicas	Hiperparámetros	kernel='linear', C=0.1	solver='lsqr', shrinkage='auto'	solver='lbfgs', C=0.1																																														
	Tamaño de ventana	0.3 s	0.3 s	0.3 s																																														
	Matriz de Confusión Pico normalizada	<table border="1"> <tr><td colspan="2"></td><td colspan="2">Predicho</td></tr> <tr><td colspan="2"></td><td>I</td><td>D</td></tr> <tr><td rowspan="2">Real</td><td>I</td><td>0.74</td><td>0.26</td></tr> <tr><td>D</td><td>0.40</td><td>0.60</td></tr> </table>			Predicho				I	D	Real	I	0.74	0.26	D	0.40	0.60	<table border="1"> <tr><td colspan="2"></td><td colspan="2">Predicho</td></tr> <tr><td colspan="2"></td><td>I</td><td>D</td></tr> <tr><td rowspan="2">Real</td><td>I</td><td>0.73</td><td>0.27</td></tr> <tr><td>D</td><td>0.39</td><td>0.61</td></tr> </table>			Predicho				I	D	Real	I	0.73	0.27	D	0.39	0.61	<table border="1"> <tr><td colspan="2"></td><td colspan="2">Predicho</td></tr> <tr><td colspan="2"></td><td>I</td><td>D</td></tr> <tr><td rowspan="2">Real</td><td>I</td><td>0.72</td><td>0.28</td></tr> <tr><td>D</td><td>0.38</td><td>0.62</td></tr> </table>			Predicho				I	D	Real	I	0.72	0.28	D	0.38	0.62	
			Predicho																																															
			I	D																																														
	Real	I	0.74	0.26																																														
D		0.40	0.60																																															
		Predicho																																																
		I	D																																															
Real	I	0.73	0.27																																															
	D	0.39	0.61																																															
		Predicho																																																
		I	D																																															
Real	I	0.72	0.28																																															
	D	0.38	0.62																																															
Precisión Pico (%)	67.18, STD= \pm 3.23	67.13, STD= \pm 3.53	66.95, STD= \pm 3.15																																															
Punto de tiempo Pico (relativo al inicio de pista en pantalla)	0.63 s	0.63 s	0.63 s																																															

Tabla 6.1: Mejores resultados de cada clasificador para las características propuestas obtenidas del MRCP

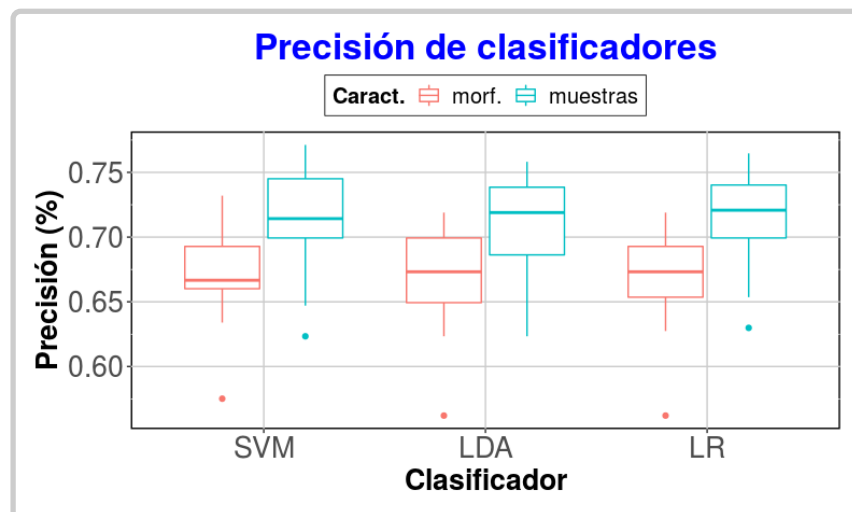


Figura 6.6: Comparación de la precisión de gran promedio entre clasificadores y características propuestas

6.0.4. Aplicación BCI

A pesar de que en este trabajo, la decodificación de lateralidad no fue evaluada en un escenario en línea, se desarrolló un primer prototipo *prueba-de-concepto* para realizar una tarea de *toma y coloca*, el cual pretende ser una inspiración para aplicaciones de asistencia en BCIs con control-compartido.

Para evaluar el sistema de manipulación asistido por visión desarrollado, se realizó la detección y localización en tiempo real de uno de dos objetos colocados al azar en ambos lados respecto de la línea central de la mesa de trabajo, dado un comando sintético que émula un estado mental asociado la lateralidad durante una intención de movimiento ($derecha=0/izquierda=1$). En <https://youtu.be/ut32Vk9I4ew> se presenta una prueba experimental de nuestro sistema, en donde se puede observar la etapa de calibración, segmentación, generación de comando y finalmente la ejecución de la tarea que primeramente se ejecuta bajo un simulador de desarrollo propio y posteriormente se logra exitosamente en el robot real con una duración promedio de 17s. Los resultados experimentales en el espacio cartesiano para una selección de objeto arbitraria se muestran en la [Figura 6.7](#), en donde se puede observar que a pesar de considerar un control cinemático el error es casi cero en las variables $x, y, roll$ y yaw , pero no en $pitch$ y z , principalmente debido al efecto de gravedad. Sin embargo, el cumplimiento de la tarea no fue afectada, por lo que se logró el objetivo de este primer prototipo experimental.

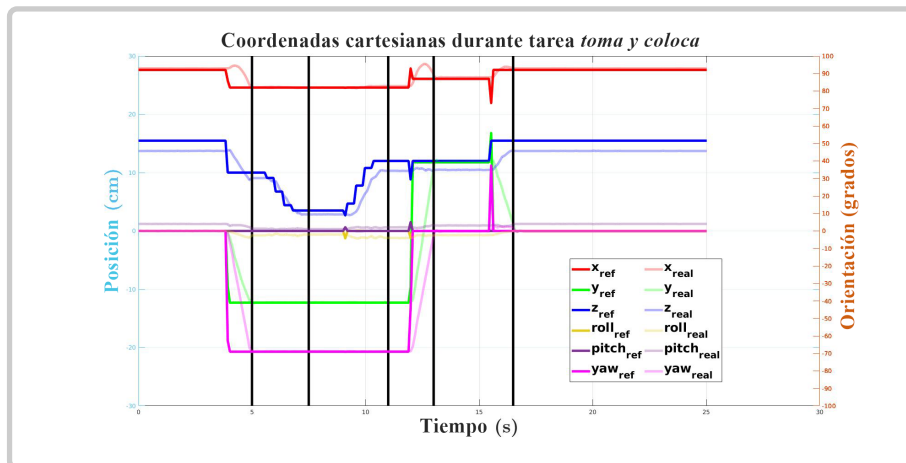


Figura 6.7: **Comparación de coordenadas cartesianas durante tarea *toma y coloca***. Las líneas oscuras representan las consignas de posición durante la tarea, mientras que las líneas claras indican los valores de posición real.

Capítulo 7

Discusión

Este capítulo aborda a detalle los resultados encontrados en el análisis neurofisiológico y la clasificación binaria. Además, se mencionan las principales limitaciones de este estudio, así como las consideraciones para la traslación a un esquema en línea. Finalmente, se discute la relevancia del sistema de manipulación propuesto y recomendaciones para proponer sistemas para escenarios más realistas en el contexto de una BCI con control-compartido.

7.0.1. Análisis Neurofisiológico

En este trabajo encontramos que los correlatos neuronales MRCP de movimientos de mano difieren en mayor proporción en áreas del hemisferio izquierdo. Este efecto de lateralización, observable en los canales sobre las áreas motoras (área motora suplementaria (SMA), corteza motora primaria (M1), y corteza somatosensorial (S1) [Kandel et al., 2000]), probablemente fue debido a la dominancia de mano derecha de la mayoría de los sujetos seleccionados de la base de datos EEGMMIDB, lo cual concuerda con los resultados obtenidos en [Bai et al., 2007].

La morfología del MRCP de Gran-Promedio se caracterizó por dos picos: uno positivo seguido de uno negativo. El positivo probablemente sea un potencial de onda lenta positiva relacionado con la evaluación del estímulo [Polich, 2007], lo cual concuerda con [Ofner et al., 2019]. El pico negativo se relaciona con el pico característico del MRCP [Shibasaki and Hallett, 2006]. Las diferencias significativas se presentaron principalmente en los canales del hemisferio izquierdo aproximadamente desde $t = 0,25$ s a $t = 0,7$ s. Es importante notar que esta diferencia se presentó alrededor de 0,3 s

antes del inicio de movimiento (véase la [Figura 6.1](#)), lo cual se explica como un LRP [[Schultze-Kraft et al., 2017](#)]. De esta manera, hacemos la suposición de que procesos relacionados con el movimiento pueden ser decodificados de señales EEG de baja frecuencia.

7.0.2. Clasificación Binaria de ensayo simple

Mostramos que participantes sanos codifican información sobre movimientos de miembros superiores asociados con la lateralidad. Esta información puede ser usada para discriminar mano derecha y mano izquierda.

Con respecto a decodificación de lateralidad de miembros superiores usando la señal de MRCP, es decir, con señales de baja frecuencia, la literatura de BCI reporta muy pocos trabajos (fuera de línea y en línea) [[Blankertz et al., 2002, 2003, 2006](#); [Krepki et al., 2007](#); [Höhne et al., 2014](#); [Schultze-Kraft et al., 2017](#)]. De esta manera, este estudio contribuye a aquellos en donde se busca decodificar la lateralidad de movimientos de manos únicamente con señales EEG de baja frecuencia (fuera de línea) en participantes sanos.

A diferencia del MRCP, las características de SMRs tales como características espectrales o espaciales han sido ampliamente estudiadas [[Yuan and He, 2014](#)] y por lo tanto trabajos en BCI para decodificar ejecución/ imaginación motora de mano unilateral (mano derecha vs mano izquierda) se han centrado principalmente en estos, como se sugiere [[Li et al., 2018a](#)]. En particular, como se reporta en [[Aljalal et al., 2020](#)], los SMRs son la señal de control endógena más usada para el control de robots basados en EEG. Sin embargo, debido a que las BCIs basadas en SMRs requieren de movimientos repetitivos de MI, los usuarios BCI necesitan aprender comandos MI no naturales, es por ello que el estado del arte actual se centra en investigaciones que permitan un control más natural, entre ellas, la decodificación continua de trayectorias desde el dominio del tiempo [[Ofner et al., 2017](#)]. Dicho lo anterior, este estudio se enfocó en la señal de MRCP, debido a que: 1) es una firma neuronal que se presenta durante la intención motora humana, 2) la latencia de detección va desde -600 a 500 ms respecto del inicio de movimiento, donde al monitorear la señal BP, el inicio del próximo movimiento puede ser predecido y se asegura la puntualidad de la neuroplasticidad durante rehabilitación [[Li et al., 2018b](#)], 3) no requiere de un entrenamiento extensivo (de semanas) como en los SMRs, y 4) se ha mostrado que codifica trayectorias de movimiento, direcciones

de movimiento, tipos de agarre, velocidad o fuerza en señales EEG y ECoG, por lo que detectar diferentes tareas de movimiento incrementa el número de posibilidades de control comparadas a las BCIs clásicas basadas en SMRs [Ofner et al., 2017].

En este trabajo, encontramos que la precisión de clasificación es superior usando características de muestras que características morfológicas de un MRCP, y que la longitud de la ventana de extracción de características de las primeras muestra un efecto de saturación cercana a 1 s, lo cual concuerda con [Ofner et al., 2017; Schwarz et al., 2017; Ofner et al., 2019], lo cual es de esperar pues la información discriminable se propaga en el tiempo [Ofner et al., 2019]. Además, la matriz de confusión sugiere que la precisión de lateralidad de miembros superiores es discriminable en tasas similares, es decir, la precisión no está sesgada, quizás debido al efecto de contralateralización. De las gráficas de precisión de gran promedio de ambas características se puede observar que la precisión supera el nivel de oportunidad práctico después del estímulo de pantalla, lo cual se puede atribuir al efecto causal de la ventana de extracción de características así como al paradigma basado en pista. Es importante notar que a pesar de que se consiguieron precisiones relativamente bajas respecto a otros estudios (véase la [Tabla 7.1](#)), incluso en el pico máximo (<75 %), la precisión supera el nivel de oportunidad práctico al menos 0.3 s antes del inicio de movimiento virtual considerado (600 ms), lo cual es prometedor para inducir plasticidad en paradigmas basados en pista.

7.0.3. Limitaciones de este estudio

El presente estudio analizó la decodificación *pseudo-online* de lateralidad de manos en una base de datos EEG basados en un paradigma basado en pista, sin embargo, algunos pasos de la canalización propuesta no son posibles usarlos en experimentos en línea, p. ej. en una BCI para propósitos de rehabilitación. Algunas de estas limitaciones se mencionan a continuación.

Una limitación de este estudio es que a pesar de que la ventana de clasificación fue causal, los filtros temporales no fueron causales, lo cual implica que se consideró también información después del instante de tiempo t . En un escenario en línea esto no sería posible, de ahí la necesidad de usar filtros causales. Otra limitación fue que debido al uso de datos con paradigma basado en pista, la parte inicial del BP reactivo estuvo combinada con un Potencial Evocado Visual (VEP) inmediatamente después del estímulo visual en pantalla, por lo que afectó la precisión del LRP. Otra limitación

Estudio	Base de datos (BD)	Tarea	Objetivo	Participantes	Canales	Protocolo de validación	Métodos	Tipo de Clasificación	Decodificación	Max. Precisión	Tiempo de detección
[Krauledat et al., 2004]	1 BD propia	ME	Mano	1(sano)	27-120	Intra-sujeto	LPR+RLDA	Clásica	Pseudo-online	95,00 %	~56 ms (relativo al inicio de mov.)
[Lu et al., 2014]	2 BD públicas + 1 BD propia	ME/MI	Mano	7(sanos)	9	Intra-sujeto	MRCP+AST	Clásica	Offline	77,17 %	-
[Abou Zeid et al., 2016]	2 BD públicas + 1 BD propia	ME	Mano	16(sanos)	9,8	Intra-sujeto	MRCP+PSTF	Clásica	Offline	74,99 % ± 10,61	-
[Abou Zeid et al., 2017]	1 BD propia	ME	Mano	14(sanos)	8	Intra-sujeto	MRCP+PSTF +DDAG	Clásica	Offline	75,08 %	-
Este trabajo	Physionet	ME	Mano	18(sanos)	16	Entre-sujetos	MRCP+SVM	Clásica	Pseudo-online	71,38 % ± 3,92	~0,69s (relativo al inicio de pista en pantalla)
[Wang et al., 2020]	BD BCI Competition II	IV ME	Mano	1(sano)	28	Intra-sujeto	MRCP+DCPM+FDA	Clásica	Offline	83,00 %	-
[Wang et al., 2020]	1 BD propia	ME	Mano	12(sanos)	60	Intra-sujeto	MRCP+DCPM+FDA	Clásica	Offline	75,88 % ± 7,75	-
[Wang et al., 2020]	BD BCI Competition II	IV ME	Mano	1(sano)	28	Intra-sujeto	DCPM+CSP+FDA	Clásica	Offline	90,00 %	-
[Lu et al., 2014]	BD BCI Competition II	IV ME	Mano	1(sano)	9	Intra-sujeto	AST+CSP	Clásica	Offline	89,00 %	-
[Karácsony et al., 2019]	Physionet	MI	Mano	109(sanos)	64	Entre-sujetos	CNN	No-Clásica	Offline	80,45 %	-
[Karácsony et al., 2019]	Physionet	MI	Mano	109(sanos)	16	Entre-sujetos	CNN	No-Clásica	Offline	72,81 %	-
[Park et al., 2013]	Physionet	MI	Mano	56(sanos)	58	Entre-sujetos	SUT-CCSP+SVM	Clásica	Offline	72,37 %	-
[Handiru and Prasad, 2016]	Physionet	MI	Mano	85(sanos)	Específicos del sujeto	Entre-sujetos	FB-CSP+SVM	Clásica	Offline	63,62 % ± 6,98	-
[Loboda et al., 2014]	Physionet	ME	Mano	103(sanos)	9	Entre-sujetos	Phase Information	Clásica	Offline	78,95 %	-
[Höhne et al., 2014]	1 BD propia	MI/MA	Mano	4(pacientes)	16	Intra-sujeto	SMR y/o SCP + aLDA	Clásica	Online	La mejor ~90 %	-

Tabla 7.1: Descripción de algunos estudios EEG reportados en la literatura de clasificación (binaria) de lateralidad de miembros superiores. Otros trabajos de decodificación online se mencionan en [Vilela and Hochberg, 2020].

fue la falta de precisión del inicio del movimiento ejecutado, pues a pesar de que se analizó un movimiento ejecutado no se contó con un canal de EMG, por lo que se calculó un inicio virtual basado en el pico mínimo de MRCP de Gran Promedio. Es importante notar que para un escenario más realista, es decir, un análisis de movimiento imaginado/intentado sería necesario considerar un paradigma distinto, tal como el sugerido en [Ofner et al., 2019]. Con respecto a los sujetos analizados, en el presente estudio consideramos únicamente datos de sujetos sanos, sin embargo para una BCI con propósitos de rehabilitación sería necesario analizar datos de pacientes, en donde se han reportado muy pocos trabajos hasta ahora como se indica en [Soekadar et al., 2015; Bockbrader, 2019; Mane et al., 2020]. Adicionalmente, como se ha mencionado en la [Subsección 7.0.1](#), las diferencias significativas en los canales del hemisferio izquierdo probablemente están asociadas con la mano dominante de los participantes analizados (sujetos diestros), por lo que en un estudio más general debería incluirse sujetos tanto diestros como zurdos.

Con respecto a la evaluación del modelo, este estudio considero únicamente la validación cruzada, motivada por el trabajo de [Schwarz et al., 2017], sin embargo una evaluación más completa se puede hacer como se sugiere en [Vega et al., 2017]. Es importante mencionar que en este trabajo únicamente se consideró la precisión del clasificador como la métrica de evaluación del modelo (como en la mayoría de trabajos de decodificación fuera de línea), sin embargo una métrica que evalúa el desempeño general de una BCI es la *tasa de transferencia de información* (ITR) [He et al., 2018], la cual considera tanto la precisión de clasificación, el número de clases y el tiempo necesario para transmitir cada comando, por lo que en posteriores estudios es necesario considerarla. Otras métricas se presentan en [Thomas et al., 2013; Thompson et al., 2014]. Adicionalmente, este estudio no consideró un análisis estadístico para evaluar diferencias significativas entre los modelos, por lo que en trabajos futuros será necesario aplicarlo.

Por último, respecto a los resultados de decodificación continua usando ventanas deslizantes, estos no se pueden comparar directamente con otros estudios debido a múltiples factores de análisis (base de datos/protocolo experimental, canalización, evaluación, etc.), sin embargo comparándola con algunos trabajos reportados en la literatura obtenemos precisiones relativamente bajas respecto a ellos (véase la [Tabla 7.1](#)). Sin embargo, para conocimiento del autor son los trabajos en donde se han obtenido los mejores resultados de precisión usando únicamente MRCP/RP y en donde algunos de ellos han

empleado estrategias adaptativas y computacionalmente más demandantes. Con respecto al factor de base de datos, es importante resaltar que algunos de estos trabajos usan también información después del inicio de movimiento. Adicionalmente, es interesante notar que incluso las mejores precisiones hasta ahora con MRCP/BP, tales como la reportada en [Wang et al., 2020], se han obtenido en un marco fuera de línea, y que hasta ahora no existe una BCI clínica basada únicamente en RP para predecir lateralidad. De esta manera, este estudio motiva a diseñar BCIs híbridas (SMRs+SCPs) basadas en EEG que permitan aumentar el desempeño de decodificación de lateralidad de miembros superiores, como se ha reportado previamente en estudios fuera de línea Lu et al. [2014]; Wang et al. [2020] y en línea [Höhne et al., 2014].

Por último pero no menos importante, una situación muy común en el análisis de base de datos públicas y no exclusiva de este trabajo, es la falta de reproducibilidad de nuevas canalizaciones, es por ello que algunos autores recomiendan usar herramientas de evaluación comparativa de algoritmos (*benchmark*) como la propuesta en [Jayaram and Barachant, 2018].

7.0.4. Transferencia a control en línea

Si bien no hay lineamientos sólidos para el diseño de una *BCI dirigida a decodificación de intención motora*, a continuación enunciamos algunos aspectos clave a considerar para una implementación en tiempo real.

Primeramente, debido a que el principal propósito de una BCI en el contexto de rehabilitación va dirigida a personas con restricciones motoras, es necesario considerar datos de *imaginación motora / movimientos intentados* de pacientes en lugar de movimientos ejecutados de sujetos sanos. Con respecto a la decodificación, es importante mencionar que un decodificador en línea, *detecta* movimientos a ritmo propio (**descanso vs movimiento**) y simultáneamente *clasifica* (i.e., **movimiento vs movimiento**), por lo que paradigmas de entrenamiento a ritmo propio deberían ser considerados en trabajos futuros, tal como el propuesto en [Ofner et al., 2019].

Con respecto a la base de datos de este estudio, de cada sujeto se obtuvieron al menos 21 ensayos, en donde la variabilidad *ensayo-a-ensayo* no permitió obtener el MRCP de ensayo simple en la mayoría de todos los sujetos, es por ello que se decidió hacer un análisis entre-sujetos. Adicionalmente, debido a que las características entre usuarios BCI (sujeto/paciente) difieren e incluso entre sesiones, se requieren métodos para

adaptar la BCI al usuario actual. Algunas de las técnicas para afrontar esto, son aquellas de *Transferencia de Aprendizaje* (TL), las cuales tienen como objetivo transferir características o clasificadores de un dominio, p. ej., sujetos o sesiones BCI, a otro dominio, p. ej., otros sujetos u otras sesiones del mismo sujeto [Lotte et al., 2018]. De esta manera, para el entrenamiento de un experimento en línea es necesario analizar cada sujeto con suficientes ensayos y considerar técnicas como TL.

Con respecto al tipo de retraso del filtro, un experimento en línea requiere del uso de filtros causales. Adicionalmente, para análisis *pseudo-online*, también se recomiendan estos filtros a fin de simular mejor las condiciones de un experimento en línea [Ranzazzo et al., 2015; Sani et al., 2016].

Relativo al rechazo de artefactos en los datos EEG, en este trabajo aplicamos el algoritmo ICA manualmente, el cual en su forma estándar está dirigido a análisis fuera de línea. Por lo que para experimentos en línea se debería considerar otra variante de ICA o usar otras técnicas, p. ej. el algoritmo de *Reconstrucción del Subespacio de Artefactos* (ASR) [Chang et al., 2020]. Otros métodos se discuten en [Kaya, 2019; Islam et al., 2020].

Con respecto a las características (de esquemas clásicos), no es nuevo que la técnica de *fusión de características* [Fazli et al., 2015] entre SMR & SCP mejora la tasa de reconocimiento en movimientos ejecutados [Liu et al., 2020], movimientos imaginados y movimientos intentados [Höhne et al., 2014] en análisis fuera de línea, sin embargo las aplicaciones en línea siguen siendo un reto que debe ser explorado en futuras investigaciones. En la última década, ha existido una tendencia de investigar si el uso de nuevos algoritmos puede mejorar las tasas de detección, principalmente con algoritmos de *aprendizaje profundo*, los cuales disminuyen o eliminan la *ingeniería de características*. Sin embargo, la mayoría de estos estudios están restringidos a resultados de análisis fuera de línea, y no son concluyentes sobre si estas técnicas pueden superar otros tipos de clasificadores [Iturrate et al., 2020], por lo tanto estudios futuros deberían también explorar métodos de *aprendizaje profundo*.

Finalmente, con respecto al desempeño de la BCI, si bien, el desempeño la BCI normalmente se asocia con el desempeño de su decodificador, este puede no reflejar que tan exitoso será el sistema completo en lograr las tareas intentadas, ni que tan satisfecho estará el usuario con tal desempeño. Por lo tanto, una evaluación completa de la BCI debería también considerar evaluar el grado de logro de objetivos relacionados a la tarea así como factores humanos [Iturrate et al., 2020], p. ej. usando el *Diseño*

Centrado en el Usuario [Höhne et al., 2014; Contreras Vidal, 2014; Kübler et al., 2014].

7.0.5. Aplicación BCI

Estudios previos, tales como [Hochberg et al., 2012; Collinger et al., 2013], han mostrado que es posible controlar continuamente sistema robóticos con BCIs invasivas, que pueden adquirir señales de mayor calidad que las BCI no invasivas. Sin embargo, las dificultades técnicas, costos económicos, y riesgos clínicos de BCIs invasivas han sido las principales limitaciones para aplicaciones reales. De esta manera, las BCIs no-invasivas, en particular basadas en EEG, han sido hasta ahora la alternativa más estudiada para aplicaciones prácticas. Sin embargo, la baja calidad de las señales EEG ha impedido hasta ahora controlar tareas complejas de robots únicamente con BCIs basadas en MI. Recientemente se ha mostrado que sujetos sanos pueden: 1) controlar un robot en un plano (2D) usando cuatro tareas mentales [Hortal et al., 2015], 2) controlar un robot en el espacio (3D) usando cuatro tareas mentales y una estrategia de movimiento secuencial [Meng et al., 2016], y 3) controlar continuamente un brazo robótico para el seguimiento aleatorio de objetivos (2D) usando cuatro tareas mentales [Edelman et al., 2019]. Es importante resaltar que estos trabajos usaron BCIs basadas en ritmos sensoriomotores.

La detección de la señal de control en usuarios objetivos de BCIs de asistencia y/o rehabilitación depende en gran medida del grado de lesión motora, por lo que esta varía de paciente a paciente. Una alternativa que puede aliviar esta situación son los esquemas de control-compartido, pues permiten disminuir la carga mental por parte del usuario en la Aplicación BCI. En este trabajo, se propuso un primer prototipo de manipulación (robot de 5 GDL) guiado por visión para ejecutar una tarea *toma y coloca* dirigido para una BCI con control-compartido. A pesar de no haber cerrado el lazo del sistema de manipulación con una BCI, la tarea se ejecutó exitosamente, dada una señal binaria virtual. En nuestro caso, la evaluación del sistema, fue únicamente el cumplimiento total de la tarea, sin embargo, en un futuro se pretende evaluar la precisión en línea con una BCI.

Con respecto al algoritmo de visión, a diferencia de otros trabajos tal como en [Wang et al., 2015], donde un sistema de visión compuesto de una cámara RGB-D y rastreo de mirada fueron usados para realizar la tarea de *toma y coloca* con una precisión de la tarea del 100 % y en un tiempo promedio de 97.8s, nuestro sistema de visión única-

mente consistió de una cámara RGB disminuyendo la carga computacional y debido al uso de un robot de pequeñas dimensiones y a que únicamente se requirió una salida binaria (virtual) desde la BCI (es decir, selección de objetivo), la tarea se completó en aprox. 15s. Con respecto al sistema de manipulación, al igual que otros trabajos como en [Zhang et al., 2017], se validó un control de posición cinemático propuesto, el cual para propósitos de este trabajo fue suficiente para llevar a cabo la tarea seleccionada. Las principales limitaciones de nuestro prototipo son: 1) Las etapas de calibración geométrica y de segmentación consumen al menos 5 min. y en el segundo caso es necesario recalibrar debido a los cambios de luz en el ambiente, 2) Únicamente se pueden localizar exactamente posiciones de puntos de superficies coplanares, y 3) Solo es posible detectar un objeto en ambos lados de la mesa respecto a la línea central.

A fin de proponer sistemas de manipulación robótica para pacientes fuera de laboratorio, creemos que es necesario proporcionar más control al paciente sin comprometer la carga mental y coadyuvar en la precisión de la tarea con la autonomía de máquina. De esta manera, como trabajo a futuro se propone abordar el problema de *alcance y agarrar* con un control continuo usando pocos estados mentales, tal como se ha mostrado recientemente en [Xia et al., 2017] para el control de cursor 2D, así como desarrollar un sistema de visión robusto que permita disminuir los tiempos de calibración, reconocer varios objetos en la escena y estimar la postura 3D. Esto podrá permitir que estos sistemas puedan asistir a pacientes para tomar cualquier objeto sobre un espacio de trabajo, contribuyendo así a romper la brecha de traslación tecnológica usando BCIs para mejorar la calidad de vida de individuos con déficit motor de miembro superior.

Capítulo 8

Conclusión y Trabajo a futuro

El presente trabajo se enfocó en el análisis de correlatos neuronales motores a partir de datos EEG, así como en el desarrollo de un sistema de manipulación experimental, los cuales tienen el propósito de ser una referencia para el futuro desarrollo de una Interfaz Cerebro-Computadora (BCI) enfocada a asistencia y/o rehabilitación.

Como parte del análisis de datos EEG, este trabajo abordó el problema de clasificación continua de lateralidad usando características del *Potencial de Preparación* (RP) durante la intención de movimientos unilaterales de miembros superiores. Los resultados obtenidos muestran que la lateralidad durante la intención de movimiento puede ser discriminada por arriba del nivel de oportunidad empírico de $\sim 53\%$ (nivel de oportunidad: 50%) en un análisis entre-sujetos ($n=18$), hasta 0.3 s antes del inicio del movimiento, y en donde se obtuvo una precisión promedio máxima de clasificación de 71.38% cercana al inicio del movimiento (~ 0.6 s), usando como características las amplitudes de muestras, una ventana de tiempo de tamaño 1.3 s y un clasificador discreto SVM. Estos resultados son prometedores en el sentido de latencia de detección durante la intención de movimiento, sin embargo la tasa de precisión obtenida fue relativamente baja, lo cual lo asociamos a la alta variabilidad ensayo-a-ensayo del MRCP.

Como trabajo a futuro en BCI motora basada en EEG, algunos puntos cruciales que se proponen abordar son: 1) Analizar datos intra-sujeto de pacientes con algún déficit motor, pues son la población objetivo en el diseño de BCIs enfocadas a rehabilitación, 2) Explorar técnicas de *Transferencia de Aprendizaje*, a fin de minimizar la calibración entre sujetos y/o sesiones, 3) Implementar técnicas de corrección de artefactos para tiempo real, 4) Evaluar la *fusión de características* (SMRs + SCPs) durante la inten-

ción de movimiento, que ya ha mostrado mejorar el desempeño del decodificador en análisis fuera de línea, 5) Explotar las capacidades del MRCP en el sentido de decodificación cinemática y grados de libertad, 6) Explorar esquemas adaptativos que permitan optimizar el diseño y rendimiento en línea de la BCI, 7) Explorar decodificadores de salida continua, que permitan mejorar el control del usuario sobre la Aplicación BCI, p. ej. el desempeño de navegación y 8) Validar las canalizaciones en tiempo real.

Con respecto a Aplicaciones BCI con control-compartido para manipulación de objetos, este trabajo propuso un primer prototipo de manipulación guiado por visión, conformado de una plataforma experimental y un algoritmo de visión artificial, para ejecutar una tarea *toma y coloca* de un objeto en una mesa de trabajo. Para fines de este trabajo, la evaluación del sistema fue únicamente en términos del cumplimiento de la tarea, es decir del seguimiento de posición, la cual se validó satisfactoriamente. Como primer prototipo, este pretende establecer las bases para prototipos dirigidos a escenarios más realistas. De esta manera, como trabajo a futuro se propone desarrollar un sistema que contemple: 1) Técnicas robustas para reconocimiento de objetos, 2) Estimar posturas 3D de objetos, 3) Validar el desempeño de la arquitectura del robot y del controlador en simuladores, a fin de agilizar el desarrollo y 4) Considerar Máquinas de Estados Finitos (MEFs) para colaborar con un control continuo de la BCI.

8.1. Contribuciones

1. Este trabajo contribuye a la literatura de BCIs basado en Potencial de Preparación (BP), en particular en decodificación *pseudo-online*, en donde se ha mostrado que usando únicamente características simples tales como las amplitudes de muestras y clasificadores discretos es posible decodificar lateralidad de movimientos de miembros superiores con precisiones de $\sim <75\%$ antes del inicio del movimiento.
2. Se propuso un algoritmo de visión artificial para obtener la postura 2D de objetos, en donde se ha mostrado que con datos obtenidos de una cámara RGB es suficiente para obtener la posición de objetos de la misma altura en un espacio de trabajo.
3. Finalmente, en este trabajo se desarrolló una prototipo *prueba-de-concepto* de un brazo robot, que para propósitos de este trabajo permitió evaluar un algoritmo de visión artificial para ejecutar una tarea *toma y coloca*, sin embargo, también podrá servir como una plataforma de aprendizaje en diversas áreas como robótica, teoría de control y visión artificial.

Bibliografía

- About Zeid, E., Rezazadeh Sereshkeh, A., Schultz, B., and Chau, T. (2017). A Ternary Brain-Computer Interface Based on Single-Trial Readiness Potentials of Self-initiated Fine Movements: A Diversified Classification Scheme. *Frontiers in Human Neuroscience*, 11:254.
- About Zeid, E., Sereshkeh, A. R., and Chau, T. (2016). A pipeline of spatio-temporal filtering for predicting the laterality of self-initiated fine movements from single trial readiness potentials. *Journal of neural engineering*, 13(6):066012.
- Adair, J. (2018). *Improving the Generalisability of Brain Computer Interface Applications via Machine Learning and Search-Based Heuristics*. PhD thesis, University of Stirling.
- Al-Quraishi, M. S., Elamvazuthi, I., Daud, S. A., Parasuraman, S., and Borboni, A. (2018). EEG-based control for upper and lower limb exoskeletons and prostheses: A systematic review. *Sensors*, 18(10):3342.
- Aljalal, M., Ibrahim, S., Djemal, R., and Ko, W. (2020). Comprehensive review on brain-controlled mobile robots and robotic arms based on electroencephalography signals. *Intelligent Service Robotics*.
- Alomari, M. H., Baniyounes, A. M., and Awada, E. A. (2014). Eeg-based classification of imagined fists movements using machine learning and wavelet transform analysis. *Int. J. Adv. Electron. Electr. Eng*, 3(3):83–87.
- Alomari, M. H., Samaha, A., and AlKamha, K. (2013). Automated classification of L/R hand movement EEG signals using advanced feature extraction and machine learning. *arXiv preprint arXiv:1312.2877*.

- Bai, O., Lin, P., Vorbach, S., Li, J., Furlani, S., and Hallett, M. (2007). Exploration of computational methods for classification of movement intention during human voluntary movement from single trial EEG. *Clinical Neurophysiology*, 118(12):2637–2655.
- Bai, O., Rathi, V., Lin, P., Huang, D., Battapady, H., Fei, D.-Y., Schneider, L., Houdayer, E., Chen, X., and Hallett, M. (2011). Prediction of human voluntary movement before it occurs. *Clinical Neurophysiology*, 122(2):364–372.
- Barthélemy, Q., Mayaud, L., Renard, Y., Kim, D., Kang, S.-W., Gunkelman, J., and Congedo, M. (2017). Online denoising of eye-blinks in electroencephalography. *Neurophysiologie Clinique*, 47(5-6):371–391.
- Bashashati, A., Fatourech, M., Ward, R. K., and Birch, G. E. (2007). A survey of signal processing algorithms in brain–computer interfaces based on electrical brain signals. *Journal of Neural Engineering*, 4(2):R32–R57.
- Bell, A. J. and Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159.
- Bertrand, O., Perrin, F., and Pernier, J. (1985). A theoretical justification of the average reference in topographic evoked potential studies. *Electroencephalography and Clinical Neurophysiology/Evoked Potentials Section*, 62(6):462–464.
- Bhagat, N. A., Venkatakrishnan, A., Abibullaev, B., Artz, E. J., Yozbatiran, N., Blank, A. A., French, J., Karmonik, C., Grossman, R. G., and O’Malley, M. K. (2016). Design and optimization of an EEG-based brain machine interface (BMI) to an upper-limb exoskeleton for stroke survivors. *Frontiers in neuroscience*, 10:122.
- Bhuyan, M. K. (2019). *Computer Vision and Image Processing: Fundamentals and Applications*. CRC Press.
- Bibian, C., Lopez-Larraz, E., Irastorza-Landa, N., Birbaumer, N., and Ramos-Murguialday, A. (2017). Evaluation of filtering techniques to extract movement intention information from low-frequency EEG activity. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2960–2963, Seogwipo. IEEE.

- Billinger, M., Daly, I., Kaiser, V., Jin, J., Allison, B. Z., Müller-Putz, G. R., and Brunner, C. (2012). Is it significant? Guidelines for reporting BCI performance. In *Towards Practical Brain-Computer Interfaces*, pages 333–354. Springer.
- Birbaumer, N., Elbert, T., Canavan, A. G., and Rockstroh, B. (1990). Slow potentials of the cerebral cortex and behavior. *Physiological Reviews*, 70(1):1–41.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. springer.
- Blankertz, B., Curio, G., and Müller, K.-R. (2002). Classifying single trial EEG: Towards brain computer interfacing. In *Advances in Neural Information Processing Systems*, pages 157–164.
- Blankertz, B., Dornhege, G., Lemm, S., Krauledat, M., Curio, G., and Müller, K.-R. (2006). The Berlin brain-computer interface: Machine learning based detection of user specific brain states. *J. UCS*, 12(6):581–607.
- Blankertz, B., Dornhege, G., Schafer, C., Krepki, R., Kohlmorgen, J., Müller, K.-R., Kunzmann, V., Losch, F., and Curio, G. (2003). Boosting bit rates and error detection for the classification of fast-paced motor commands based on single-trial EEG analysis. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2):127–131.
- Blankertz, B., Lemm, S., Treder, M., Haufe, S., and Müller, K.-R. (2011). Single-trial analysis and classification of ERP components — A tutorial. *NeuroImage*, 56(2):814–825.
- Bockbrader, M. (2019). Upper limb sensorimotor restoration through brain-computer interface technology in tetraparesis. *Current Opinion in Biomedical Engineering*, 11:85–101.
- Brunner, C., Naeem, M., Leeb, R., Graimann, B., and Pfurtscheller, G. (2007). Spatial filtering and selection of optimized components in four class motor imagery EEG data using independent components analysis. *Pattern Recognition Letters*, 28(8):957–964.
- Carvalhaes, C. and de Barros, J. A. (2014). The Surface Laplacian Technique in EEG: Theory and Methods. *arXiv:1406.0458 [physics]*. Comment: 43 pages, 8 figures.

- Chang, C.-Y., Hsu, S.-H., Pion-Tonachini, L., and Jung, T.-P. (2020). Evaluation of Artifact Subspace Reconstruction for Automatic Artifact Components Removal in Multi-Channel EEG Recordings. *IEEE Transactions on Biomedical Engineering*, 67(4):1114–1121.
- Chang, W. H. and Kim, Y.-H. (2013). Robot-assisted Therapy in Stroke Rehabilitation. *Journal of Stroke*, 15(3):174.
- Choreño-Parra, J. A., Carnalla-Cortés, M., and Guadarrama-Ortíz, P. (2019). Enfermedad vascular cerebral isquémica: Revisión extensa de la bibliografía para el médico de primer contacto. *Medicina interna de México*, 35(1):61–79.
- Cohen, M. X. (2014). *Analyzing Neural Time Series Data: Theory and Practice*. MIT press.
- Colebatch, J. G. (2007). Bereitschaftspotential and movement-related potentials: Origin, significance, and application in disorders of human movement. *Movement Disorders*, 22(5):601–610.
- Collinger, J. L., Wodlinger, B., Downey, J. E., Wang, W., Tyler-Kabara, E. C., Weber, D. J., McMorland, A. J., Velliste, M., Boninger, M. L., and Schwartz, A. B. (2013). 7 degree-of-freedom neuroprosthetic control by an individual with tetraplegia. *Lancet*, 381(9866):557–564.
- Combrisson, E. and Jerbi, K. (2015). Exceeding chance level by chance: The caveat of theoretical chance levels in brain signal classification and statistical assessment of decoding accuracy. *Journal of neuroscience methods*, 250:126–136.
- Contreras Vidal, J. L. (2014). Identifying engineering, clinical and patient’s metrics for evaluating and quantifying performance of brain-machine interface (BMI) systems. In *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1489–1492. IEEE.
- Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.
- Costa, J. C. G. D., Da-Silva, P. J. G., Almeida, R. M. V. R., and Infantosi, A. F. C. (2014). Validation in Principal Components Analysis Applied to EEG Data. *Computational and Mathematical Methods in Medicine*, 2014:1–10.

- Cramer, S. C., Sur, M., Dobkin, B. H., O'Brien, C., Sanger, T. D., Trojanowski, J. Q., Rumsey, J. M., Hicks, R., Cameron, J., Chen, D., Chen, W. G., Cohen, L. G., deCharms, C., Duffy, C. J., Eden, G. F., Fetz, E. E., Filart, R., Freund, M., Grant, S. J., Haber, S., Kalivas, P. W., Kolb, B., Kramer, A. F., Lynch, M., Mayberg, H. S., McQuillen, P. S., Nitkin, R., Pascual-Leone, A., Reuter-Lorenz, P., Schiff, N., Sharma, A., Shekim, L., Stryker, M., Sullivan, E. V., and Vinogradov, S. (2011). Harnessing neuroplasticity for clinical applications. *Brain*, 134(6):1591–1609.
- Daly, J. J. and Wolpaw, J. R. (2008). Brain–computer interfaces in neurological rehabilitation. *The Lancet Neurology*, 7(11):1032–1043.
- de Cheveigné, A. and Arzounian, D. (2018). Robust detrending, rereferencing, outlier detection, and inpainting for multichannel data. *NeuroImage*, 172:903–912.
- de Cheveigné, A. and Simon, J. Z. (2008). Denoising based on spatial filtering. *Journal of Neuroscience Methods*, 171(2):331–339.
- Deecke, L., Grözing, B., and Kornhuber, H. H. (1976). Voluntary finger movement in man: Cerebral potentials and theory. *Biological cybernetics*, 23(2):99–119.
- Derambure, P., Defebvre, L., Bourriez, J., Cassim, F., and Guieu, J. (1999). Désynchronisation et synchronisation liées à l'événement Étude de la réactivité des rythmes électrocorticaux en relation avec la planification et l'exécution du mouvement volontaire. *Neurophysiologie Clinique/Clinical Neurophysiology*, 29(1):53–70.
- Edelman, B. J., Meng, J., Suma, D., Zurn, C., Nagarajan, E., Baxter, B. S., Cline, C. C., and He, B. (2019). Noninvasive neuroimaging enhances continuous neural tracking for robotic device control. *Science Robotics*, 4(31).
- Esch, L., Dinh, C., Larson, E., Engemann, D., Jas, M., Khan, S., Gramfort, A., and Hämäläinen, M. S. (2019). MNE: Software for Acquiring, Processing, and Visualizing MEG/EEG Data. *Magnetoencephalography: From Signals to Dynamic Cortical Networks*, pages 355–371.
- Fazli, S., Dähne, S., Samek, W., Bießmann, F., and Mueller, K.-R. (2015). Learning from more than one data source: Data fusion techniques for sensorimotor rhythm-based brain–computer interfaces. *Proceedings of the IEEE*, 103(6):891–906.

- Garipelli, G., Chavarriaga, R., and del R Millán, J. (2013). Single trial analysis of slow cortical potentials: A study on anticipation related potentials. *Journal of neural engineering*, 10(3):036014.
- Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220.
- Graimann, B., Allison, B. Z., and Pfurtscheller, G. (2010). *Brain-Computer Interfaces: Revolutionizing Human-Computer Interaction*. Springer Science & Business Media.
- Gupta, S. K. (2012). The relevance of confidence interval and P-value in inferential statistics. *Indian journal of pharmacology*, 44(1):143.
- Gutiérrez, I. D. and González, M. E. M. (2014). Rehabilitación del paciente con enfermedad vascular cerebral (EVC). page 15.
- Hamano, T., Lüders, H. O., Ikeda, A., Collura, T. F., Comair, Y. G., and Shibasaki, H. (1997). The cortical generators of the contingent negative variation in humans: A study with subdural electrodes. *Electroencephalography and Clinical Neurophysiology/Evoked Potentials Section*, 104(3):257–268.
- Han, C.-H., Müller, K.-R., and Hwang, H.-J. (2020). Brain-Switches for Asynchronous Brain–Computer Interfaces: A Systematic Review. *Electronics*, 9(3):422.
- Han Yuan and Bin He (2014). Brain–Computer Interfaces Using Sensorimotor Rhythms: Current State and Future Perspectives. *IEEE Transactions on Biomedical Engineering*, 61(5):1425–1435.
- Handiru, V. S. and Prasad, V. A. (2016). Optimized bi-objective eeg channel selection and cross-subject generalization with brain–computer interfaces. *IEEE Transactions on Human-Machine Systems*, 46(6):777–786.
- Hartley, R. and Zisserman, A. (2003). *Multiple View Geometry in Computer Vision*. Cambridge university press.

- Hassan, A., Ghani, U., Riaz, F., Rehman, S., Jochumsen, M., Taylor, D., and Niazi, I. K. (2015). Using a portable device for online single-trial MRCP detection and classification. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 527–534. Springer.
- Haykin, S. (2010). *Neural Networks and Learning Machines, 3/E*. Pearson Education India.
- Haynes, J.-D. (2011). Decoding and predicting intentions. *Annals of the New York Academy of Sciences*, 1224(1):9–21.
- He, B., Gao, S., Yuan, H., and Wolpaw, J. R. (2013). Brain–computer interfaces. In *Neural Engineering*, pages 87–151. Springer.
- He, Y., Eguren, D., Azorín, J. M., Grossman, R. G., Luu, T. P., and Contreras-Vidal, J. L. (2018). Brain–machine interfaces for controlling lower-limb powered robotic systems. *Journal of neural engineering*, 15(2):021004.
- Hochberg, L. R., Bacher, D., Jarosiewicz, B., Masse, N. Y., Simeral, J. D., Vogel, J., Haddadin, S., Liu, J., Cash, S. S., and Van Der Smagt, P. (2012). Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature*, 485(7398):372–375.
- Höhne, J., Holz, E., Staiger-Sälzer, P., Müller, K.-R., Kübler, A., and Tangermann, M. (2014). Motor imagery for severely motor-impaired patients: Evidence for brain–computer interfacing as superior control solution. *PloS one*, 9(8):e104854.
- Hortal, E., Planelles, D., Costa, A., Iáñez, E., Úbeda, A., Azorín, J., and Fernández, E. (2015). SVM-based Brain–Machine Interface for controlling a robot arm through four mental tasks. *Neurocomputing*, 151:116–121.
- Huong, N. T. M. and Linh, H. Q. (2017). Classification of left/right hand movement EEG signals using event related potentials and advanced features. In *International Conference on the Development of Biomedical Engineering in Vietnam*, pages 209–215. Springer.
- Hyvärinen, A. and Oja, E. (1997). A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492.

- Ibáñez, J., Serrano, J. I., del Castillo, M. D., Monge-Pereira, E., Molina-Rueda, F., Alguacil-Diego, I., and Pons, J. L. (2014). Detection of the onset of upper-limb movements based on the combined analysis of changes in the sensorimotor rhythms and slow cortical potentials. *Journal of Neural Engineering*, 11(5):056009.
- Islam, M. K., Rastegarnia, A., and Sanei, S. (2020). Signal Artifacts and Techniques for Artifacts and Noise Removal. In *Signal Processing Techniques for Computational Health Informatics*, pages 23–79. Springer.
- Islam, M. K., Rastegarnia, A., and Yang, Z. (2016). Methods for artifact detection and removal from scalp EEG: A review. *Neurophysiologie Clinique/Clinical Neurophysiology*, 46(4-5):287–305.
- Iturrate, I., Chavarriaga, R., and Millán, J. D. R. (2020). General principles of machine learning for brain-computer interfacing. *Handbook of Clinical Neurology*, 168:311–328.
- Jahanshahi, M. and Hallett, M. (2003). *The Bereitschaftspotential: Movement-Related Cortical Potentials*. Springer Science & Business Media.
- Jahanshahi, M., Jenkins, I. H., Brown, R. G., Marsden, C. D., Passingham, R. E., and Brooks, D. J. (1995). Self-initiated versus externally triggered movements: I. An investigation using measurement of regional cerebral blood flow with PET and movement-related potentials in normal and Parkinson’s disease subjects. *Brain*, 118(4):913–933.
- James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning with Applications in R*, volume 112. Springer.
- Jankelowitz, S. and Colebatch, J. (2002). Movement-related potentials associated with self-paced, cued and imagined arm movements. *Experimental brain research*, 147(1):98–107.
- Jayaram, V. and Barachant, A. (2018). MOABB: Trustworthy algorithm benchmarking for BCIs. *Journal of Neural Engineering*, 15(6):066011.
- Jiang, N., Mrachacz-Kersting, N., Xu, R., Dremstrup, K., and Farina, D. (2014). An Accurate, Versatile, and Robust Brain Switch for Neurorehabilitation. In Guger,

- C., Vaughan, T., and Allison, B., editors, *Brain-Computer Interface Research*, pages 47–61. Springer International Publishing, Cham.
- Jochumsen, M., Khan Niazi, I., Samran Navid, M., Nabeel Anwar, M., Farina, D., and Dremstrup, K. (2015). Online multi-class brain-computer interface for detection and classification of lower limb movement intentions and kinetics for stroke rehabilitation. *Brain-Computer Interfaces*, 2(4):202–210.
- Jochumsen, M., Navid, M. S., Wiberg, R., Anwar, M. N., and Niazi, I. K. (2014a). Online detection and classification of movement kinetics. page 4.
- Jochumsen, M., Niazi, I. K., Rovsing, H., Rovsing, C., Nielsen, G. A. R., Andersen, T. K., Dong, N. P. T., Sørensen, M. E., Mrachacz-Kersting, N., Jiang, N., Farina, D., and Dremstrup, K. (2014b). Detection of Movement Intentions through a Single Channel of Electroencephalography. In Jensen, W., Andersen, O. K., and Akay, M., editors, *Replace, Repair, Restore, Relieve – Bridging Clinical and Engineering Solutions in Neurorehabilitation*, volume 7, pages 465–472. Springer International Publishing, Cham.
- Jović, A., Brkić, K., and Bogunović, N. (2015). A review of feature selection methods with applications. In *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1200–1205. Ieee.
- Jung, P., Baumgärtner, U., Bauermann, T., Magerl, W., Gawehn, J., Stoeter, P., and Treede, R.-D. (2003). Asymmetry in the human primary somatosensory cortex and handedness. *NeuroImage*, 19(3):913–923.
- Kandel, E. R., Schwartz, J. H., Jessell, T. M., of Biochemistry, D., Jessell, M. B. T., Siegelbaum, S., and Hudspeth, A. J. (2000). *Principles of Neural Science*, volume 4. McGraw-hill New York.
- Karácsony, T., Hansen, J. P., Iversen, H. K., and Puthusserypady, S. (2019). Brain Computer Interface for Neuro-rehabilitation With Deep Learning Classification and Virtual Reality Feedback. In *Proceedings of the 10th Augmented Human International Conference 2019*, pages 1–8.

- Karimi, F., Kofman, J., Mrachacz-Kersting, N., Farina, D., and Jiang, N. (2017). Detection of Movement Related Cortical Potentials from EEG Using Constrained ICA for Brain-Computer Interface Applications. *Frontiers in Neuroscience*, 11.
- Kawashima, R., Yamada, K., Kinomura, S., Yamaguchi, T., Matsui, H., Yoshioka, S., and Fukuda, H. (1993). Regional cerebral blood flow changes of cortical motor areas and prefrontal areas in humans related to ipsilateral and contralateral hand movement. *Brain Research*, 623(1):33–40.
- Kaya, I. (2019). A Brief Summary of EEG Artifact Handling. *arXiv:2001.00693 [eess, q-bio]*.
- Klamroth-Marganska, V. (2018). Stroke Rehabilitation: Therapy Robots and Assistive Devices. In Kerkhof, P. L. M. and Miller, V. M., editors, *Sex-Specific Analysis of Cardiovascular Function*, volume 1065, pages 579–587. Springer International Publishing, Cham.
- Kowalczyk, A. (2017). Support vector machines succinctly. *Syncfusion Inc.*
- Krauledat, M., Dornhege, G., Blankertz, B., Losch, F., Curio, G., and Müller, K.-R. (2004). Improving speed and accuracy of brain-computer interfaces using readiness potential features. In *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 2, pages 4511–4515. IEEE.
- Krepki, R., Blankertz, B., Curio, G., and Müller, K.-R. (2007). The Berlin Brain-Computer Interface (BBCI)—towards a new communication channel for online control in gaming applications. *Multimedia Tools and Applications*, 33(1):73–90.
- Kropotov, J. D. (2009). Methods of Analysis of Background EEG. In *Quantitative EEG, Event-Related Potentials and Neurotherapy*, pages 116–160. Elsevier.
- Kübler, A., Holz, E. M., Riccio, A., Zickler, C., Kaufmann, T., Kleih, S. C., Staiger-Sälzer, P., Desideri, L., Hoogerwerf, E.-J., and Mattia, D. (2014). The User-Centered Design as Novel Perspective for Evaluating the Usability of BCI-Controlled Applications. *PLoS ONE*, 9(12):e112392.
- Langhorne, P., Bernhardt, J., and Kwakkel, G. (2011). Stroke rehabilitation. *The Lancet*, 377(9778):1693–1702.

- Langhorne, P., Coupar, F., and Pollock, A. (2009a). Motor recovery after stroke: A systematic review. *The Lancet Neurology*, 8(8):741–754.
- Langhorne, P., Sandercock, P., and Prasad, K. (2009b). Evidence-based practice for stroke. *The Lancet Neurology*, 8(4):308–309.
- Ledoit, O. and Wolf, M. (2000). A well conditioned estimator for large dimensional covariance matrices.
- Lee, T.-W., Girolami, M., and Sejnowski, T. J. (1999). Independent component analysis using an extended infomax algorithm for mixed subgaussian and supergaussian sources. *Neural computation*, 11(2):417–441.
- Lew, E. (2012). Detection of self-paced reaching movement intention from EEG signals. *Frontiers in Neuroengineering*, 5.
- Lew, E. Y. L., Chavarriaga, R., Silvoni, S., and Millán, J. d. R. (2014). Single trial prediction of self-paced reaching directions from EEG signals. *Frontiers in Neuroscience*, 8.
- Li, H., Huang, G., Lin, Q., Zhao, J.-L., Lo, W.-L. A., Mao, Y.-R., Chen, L., Zhang, Z.-G., Huang, D.-F., and Li, L. (2018a). Combining movement-related cortical potentials and event-related desynchronization to study movement preparation and execution. *Frontiers in neurology*, 9:822.
- Li, M., Xu, G., Xie, J., and Chen, C. (2018b). A review: Motor rehabilitation after stroke with control based on human intent. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, 232(4):344–360.
- Liu, T., Huang, G., Jiang, N., Yao, L., and Zhang, Z. (2020). Reduce brain computer interface inefficiency by combining sensory motor rhythm and movement-related cortical potential features. *Journal of Neural Engineering*, 17(3):035003.
- Loboda, A., Margineanu, A., Rotariu, G., and Lazar, A. M. (2014). Discrimination of EEG-based motor imagery tasks by means of a simple phase information method. *International Journal of Advanced Research in Artificial Intelligence*, 3(10).

- López-Larraz, E., Montesano, L., Gil-Agudo, Á., and Minguez, J. (2014). Continuous decoding of movement intention of upper limb self-initiated analytic movements from pre-movement EEG correlates. *Journal of NeuroEngineering and Rehabilitation*, 11(1):153.
- López-Larraz, E., Trincado-Alonso, F., Rajasekaran, V., Pérez-Nombela, S., Del-Ama, A. J., Aranda, J., Minguez, J., Gil-Agudo, A., and Montesano, L. (2016). Control of an ambulatory exoskeleton with a brain-machine interface for spinal cord injury gait rehabilitation. *Frontiers in neuroscience*, 10:359.
- Lotte, F. (2008). *Study of Electroencephalographic Signal Processing and Classification Techniques towards the Use of Brain-Computer Interfaces in Virtual Reality Applications*. PhD Thesis.
- Lotte, F. (2014). A Tutorial on EEG Signal-processing Techniques for Mental-state Recognition in Brain-Computer Interfaces. In Miranda, E. R. and Castet, J., editors, *Guide to Brain-Computer Music Interfacing*, pages 133–161. Springer London, London.
- Lotte, F., Bougrain, L., Cichocki, A., Clerc, M., Congedo, M., Rakotomamonjy, A., and Yger, F. (2018). A review of classification algorithms for EEG-based brain-computer interfaces: A 10 year update. *Journal of Neural Engineering*, 15(3):031005.
- Lotte, F., Bougrain, L., and Clerc, M. (2015). *Electroencephalography (EEG)-Based Brain-Computer Interfaces*, pages 1–20. John Wiley & Sons, Inc., Hoboken, NJ, USA.
- Lu, J., McFarland, D. J., and Wolpaw, J. R. (2013). Adaptive Laplacian filtering for sensorimotor rhythm-based brain-computer interfaces. *Journal of Neural Engineering*, 10(1):016002.
- Lu, J., Xie, K., and McFarland, D. J. (2014). Adaptive spatio-temporal filtering for movement related potentials in EEG-based brain-computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(4):847–857.
- Mak, J. and Wolpaw, J. (2009). Clinical Applications of Brain-Computer Interfaces: Current State and Future Prospects. *IEEE Reviews in Biomedical Engineering*, 2:187–199.

- Makeig, S., Westerfield, M., Jung, T.-P., Covington, J., Townsend, J., Sejnowski, T. J., and Courchesne, E. (1999). Functionally Independent Components of the Late Positive Event-Related Potential during Visual Spatial Attention. *The Journal of Neuroscience*, 19(7):2665–2680.
- Mane, R., Chouhan, T., and Guan, C. (2020). BCI for stroke rehabilitation: Motor and beyond. *Journal of Neural Engineering*, 17(4):041001.
- Mantini, D., Perrucci, M., Cugini, S., Ferretti, A., Romani, G., and Del Gratta, C. (2007). Complete artifact removal for EEG recorded during continuous fMRI using independent component analysis. *NeuroImage*, 34(2):598–607.
- Maris, E. and Oostenveld, R. (2007). Nonparametric statistical testing of EEG-and MEG-data. *Journal of neuroscience methods*, 164(1):177–190.
- McFarland, D. J., McCane, L. M., David, S. V., and Wolpaw, J. R. (1997). Spatial filter selection for EEG-based communication. *Electroencephalography and Clinical Neurophysiology*, 103(3):386–394.
- Meng, J., Zhang, S., Bekyo, A., Olsoe, J., Baxter, B., and He, B. (2016). Noninvasive Electroencephalogram Based Control of a Robotic Arm for Reach and Grasp Tasks. *Scientific Reports*, 6(1):38565.
- Mrachacz-Kersting, N., Jiang, N., Stevenson, A. J. T., Niazi, I. K., Kostic, V., Pavlovic, A., Radovanovic, S., Djuric-Jovicic, M., Agosta, F., and Dremstrup, K. (2016). Efficient neuroplasticity induction in chronic stroke patients by an associative brain-computer interface. *Journal of neurophysiology*, 115(3):1410–1421.
- Mühl, C., Allison, B., Nijholt, A., and Chanel, G. (2014). A survey of affective brain computer interfaces: Principles, state-of-the-art, and challenges. *Brain-Computer Interfaces*, 1(2):66–84.
- Müller-Putz, G., Scherer, R., Brunner, C., Leeb, R., and Pfurtscheller, G. (2008). Better than random: A closer look on BCI results. *International Journal of Bioelectromagnetism*, 10(ARTICLE):52–55.
- Nam, C. S., Nijholt, A., and Lotte, F. (2018). *Brain-Computer Interfaces Handbook: Technological and Theoretical Advances*. CRC Press.

- Nicolas-Alonso, L. F. and Gomez-Gil, J. (2012). Brain computer interfaces, a review. *Sensors*, 12(2):1211–1279.
- Nunez, P. L. and Srinivasan, R. (2006). *Electric Fields of the Brain: The Neurophysics of EEG*. Oxford University Press, USA.
- Ofner, P., Schwarz, A., Pereira, J., and Müller-Putz, G. R. (2017). Upper limb movements can be decoded from the time-domain of low-frequency EEG. *PLOS ONE*, 12(8):e0182578.
- Ofner, P., Schwarz, A., Pereira, J., Wyss, D., Wildburger, R., and Müller-Putz, G. R. (2019). Attempted arm and hand movements can be decoded from low-frequency EEG from persons with spinal cord injury. *Scientific reports*, 9(1):1–15.
- Oweiss, K. G. (2010). *Statistical Signal Processing for Neuroscience and Neurotechnology*. Academic Press.
- Paris, S., Kornprobst, P., Tumblin, J., and Durand, F. (2008). Bilateral Filtering: Theory and Applications. *Foundations and Trends® in Computer Graphics and Vision*, 4(1):1–75.
- Park, C., Took, C. C., and Mandic, D. P. (2013). Augmented complex common spatial patterns for classification of noncircular EEG from motor imagery tasks. *IEEE Transactions on neural systems and rehabilitation engineering*, 22(1):1–10.
- Pereira, F., Mitchell, T., and Botvinick, M. (2009). Machine learning classifiers and fMRI: A tutorial overview. *Neuroimage*, 45(1):S199–S209.
- Pfurtscheller, G., Guger, C., Müller, G., Krausz, G., and Neuper, C. (2000). Brain oscillations control hand orthosis in a tetraplegic. *Neuroscience Letters*, 292(3):211–214.
- Pfurtscheller, G. and Lopes da Silva, F. (1999). Event-related EEG/MEG synchronization and desynchronization: Basic principles. *Clinical Neurophysiology*, 110(11):1842–1857.
- Pfurtscheller, G., Neuper, C., and Berger, J. (1994). Source localization using event-related desynchronization (ERD) within the alpha band. *Brain Topography*, 6(4):269–275.

- Polich, J. (2007). Updating P300: An integrative theory of P3a and P3b. *Clinical neurophysiology*, 118(10):2128–2148.
- Ramadan, R. A. and Vasilakos, A. V. (2017). Brain computer interface: Control signals review. *Neurocomputing*, 223:26–44.
- Randazzo, L., Iturrate, I., Chavarriaga, R., Leeb, R., and Millán, J. d. R. (2015). Detecting intention to grasp during reaching movements from EEG. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1115–1118. IEEE.
- Rao, R. P. N. (2013). *Brain-Computer Interfacing: An Introduction*. Cambridge University Press.
- Ren Xu, Ning Jiang, Chuang Lin, Mrachacz-Kersting, N., Dremstrup, K., and Farina, D. (2014). Enhanced Low-Latency Detection of Motor Intention From EEG for Closed-Loop Brain-Computer Interface Applications. *IEEE Transactions on Biomedical Engineering*, 61(2):288–296.
- Rodpongpun, S., Janyalikit, T., and Ratanamahatana, C. A. (2020). Influential Factors of an Asynchronous BCI for Movement Intention Detection. *Computational and Mathematical Methods in Medicine*, 2020:1–12.
- Roots, K. (2020). *Development of EEG-Based BCI Application Using Machine Learning to Classify Motor Movement and Imagery*. Bachelor’s thesis, UNIVERSITY OF TARTU, Estonia.
- Ruchkin, D. S., Sutton, S., Mahaffey, D., and Glaser, J. (1986). Terminal CNV in the absence of motor response. *Electroencephalography and Clinical Neurophysiology*, 63(5):445–463.
- Sani, O. G., Chavarriaga, R., Shamsollahi, M. B., and del R Millan, J. (2016). Detection of movement related cortical potential: Effects of causal vs. non-causal processing. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 5733–5736, Orlando, FL. IEEE.
- Savić, A. M., Lontis, E. R., Mrachacz-Kersting, N., and Popović, M. B. (2020). Dynamics of movement-related cortical potentials and sensorimotor oscillations during palmar grasp movements. *European Journal of Neuroscience*, 51(9):1962–1970.

- Schalk, G. and Allison, B. Z. (2018). Noninvasive Brain–Computer Interfaces. In *Neuromodulation*, pages 357–377. Elsevier.
- Schalk, G. and Mellinger, J. (2010). *A Practical Guide to Brain–Computer Interfacing with BCI2000: General-Purpose Software for Brain-Computer Interface Research, Data Acquisition, Stimulus Presentation, and Brain Monitoring*. Springer Science & Business Media.
- Schultze-Kraft, M., Neumann, M., Lundfall, M., Wagner, P., Birman, D., Haynes, J.-D., and Blankertz, B. (2017). Predicting Motor Intentions With Closed-Loop Brain-Computer Interfaces. In *Brain-Computer Interface Research*, pages 79–90. Springer.
- Schwarz, A., Ofner, P., Pereira, J., Sburlea, A. I., and Mueller-Putz, G. R. (2017). Decoding natural reach-and-grasp actions from human EEG. *Journal of neural engineering*, 15(1):016005.
- Semmlow, J. L. and Griffel, B. (2014). *Biosignal and Medical Image Processing*. CRC press.
- Shakeel, A., Navid, M. S., Anwar, M. N., Mazhar, S., Jochumsen, M., and Niazi, I. K. (2015). A Review of Techniques for Detection of Movement Intention Using Movement-Related Cortical Potentials. *Computational and Mathematical Methods in Medicine*, 2015:1–13.
- Shibasaki, H. and Hallett, M. (2006). What is the Bereitschaftspotential? *Clinical Neurophysiology*, 117(11):2341–2356.
- Soekadar, S. R., Birbaumer, N., Slutzky, M. W., and Cohen, L. G. (2015). Brain–machine interfaces in neurorehabilitation of stroke. *Neurobiology of disease*, 83:172–179.
- Spong, M. W., Hutchinson, S., and Vidyasagar, M. (2006). *Robot Modeling and Control*.
- Studer, P., Kratz, O., Gevensleben, H., Rothenberger, A., Moll, G. H., Hautzinger, M., and Heinrich, H. (2014). Slow cortical potential and theta/beta neurofeedback training in adults: Effects on attentional processes and motor system excitability. *Frontiers in Human Neuroscience*, 8.

- Subasi, A. (2019). *Practical Guide for Biomedical Signals Analysis Using Machine Learning Techniques: A MATLAB Based Approach*. Academic Press.
- Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Springer Science & Business Media.
- Tayeb, Z., Fedjaev, J., Ghaboosi, N., Richter, C., Everding, L., Qu, X., Wu, Y., Cheng, G., and Conradt, J. (2019). Validating Deep Neural Networks for Online Decoding of Motor Imagery Movements from EEG Signals. *Sensors*, 19:210.
- Thickbroom, G. W. (2007). Transcranial magnetic stimulation and synaptic plasticity: Experimental framework and human models. *Experimental Brain Research*, 180(4):583–593.
- Thomas, E., Dyson, M., and Clerc, M. (2013). An analysis of performance evaluation for motor-imagery based BCI. *Journal of neural engineering*, 10(3):031001.
- Thompson, D. E., Quitadamo, L. R., Mainardi, L., Gao, S., Kindermans, P.-J., Simeral, J. D., Fazel-Rezai, R., Matteucci, M., Falk, T. H., and Bianchi, L. (2014). Performance measurement for brain–computer or brain–machine interfaces: A tutorial. *Journal of neural engineering*, 11(3):035001.
- Tonin, L. and Millán, J. d. R. (2021). Noninvasive Brain–Machine Interfaces for Robotic Devices. *Annual Review of Control, Robotics, and Autonomous Systems*, 4(1):annurev-control-012720-093904.
- van Dokkum, L., Ward, T., and Laffont, I. (2015). Brain computer interfaces for neurorehabilitation – its current status as a rehabilitation strategy post-stroke. *Annals of Physical and Rehabilitation Medicine*, 58(1):3–8.
- Vega, R., Sajed, T., Mathewson, K. W., Khare, K., Pilarski, P. M., Greiner, R., Sanchez-Ante, G., and Antelis, J. M. (2017). Assessment of feature selection and classification methods for recognizing motor imagery tasks from electroencephalographic signals. *Artif. Intell. Research*, 6(1):37–51.
- Venkatakrishnan, A., Francisco, G. E., and L. Contreras-Vidal, J. (2014). Applications of Brain–Machine Interface Systems in Stroke Recovery and Rehabilitation. *Current Physical Medicine and Rehabilitation Reports*, 2(2):93–105.

- Vilela, M. and Hochberg, L. R. (2020). Applications of brain-computer interfaces to the control of robotic and prosthetic arms. In *Handbook of Clinical Neurology*, volume 168, pages 87–99. Elsevier.
- Volkman, J., Schnitzler, A., Witte, O. W., and Freund, H.-J. (1998). Handedness and Asymmetry of Hand Representation in Human Motor Cortex. *Journal of Neurophysiology*, 79(4):2149–2154.
- Wairagkar, M., Hayashi, Y., and Nasuto, S. J. (2018). Exploration of neural correlates of movement intention based on characterisation of temporal dependencies in electroencephalography. *PloS one*, 13(3):e0193722.
- Walter, W. G., Cooper, R., Aldridge, V. J., McCallum, W. C., and Winter, A. L. (1964). Contingent negative variation: An electric sign of sensori-motor association and expectancy in the human brain. *Nature*, 203(4943):380–384.
- Wang, H., Arceo, R., Chen, S., Ding, L., Jia, J., and Yao, J. (2019). Effectiveness of interventions to improve hand motor function in individuals with moderate to severe stroke: A systematic review protocol. *BMJ open*, 9(9):e032413.
- Wang, H., Dong, X., Chen, Z., and Shi, B. E. (2015). Hybrid gaze/EEG brain computer interface for robot arm control on a pick and place task. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1476–1479, Milan. IEEE.
- Wang, K., Xu, M., Wang, Y., Zhang, S., Chen, L., and Ming, D. (2020). Enhance decoding of pre-movement EEG patterns for brain-computer interfaces. *Journal of Neural Engineering*, 17(1):016033.
- Widmann, A., Schröger, E., and Maess, B. (2015). Digital filter design for electrophysiological data – a practical approach. *Journal of Neuroscience Methods*, 250:34–46.
- Wolpaw, J. and Wolpaw, E. W. (2012). *Brain-Computer Interfaces: Principles and Practice*. OUP USA.
- Wright, D. J., Holmes, P. S., and Smith, D. (2011). Using the Movement-Related Cortical Potential to Study Motor Skill Learning. *Journal of Motor Behavior*, 43(3):193–201.

- Xia, B., Cao, L., Maysam, O., Li, J., Xie, H., Su, C., and Birbaumer, N. (2017). A binary motor imagery tasks based brain-computer interface for two-dimensional movement control. *Journal of Neural Engineering*, 14(6):066009.
- Xu, B., Song, A., Zhao, G., Xu, G., Pan, L., Yang, R., Li, H., and Cui, J. (2015). Design and evaluation of a motor imagery electroencephalogram-controlled robot system. *Advances in Mechanical Engineering*, 7(3):168781401557360.
- Xu, R., Jiang, N., Lin, C., Mrachacz-Kersting, N., Dremstrup, K., and Farina, D. (2013). Enhanced low-latency detection of motor intention from EEG for closed-loop brain-computer interface applications. *IEEE Transactions on Biomedical Engineering*, 61(2):288–296.
- Xu, R., Jiang, N., Mrachacz-Kersting, N., Dremstrup, K., and Farina, D. (2016). Factors of Influence on the Performance of a Short-Latency Non-Invasive Brain Switch: Evidence in Healthy Individuals and Implication for Motor Function Rehabilitation. *Frontiers in Neuroscience*, 9.
- Xu, Y., Ding, C., Shu, X., Gui, K., Bezsudnova, Y., Sheng, X., and Zhang, D. (2019). Shared control of a robotic arm using non-invasive brain-computer interface and computer vision guidance. *Robotics and Autonomous Systems*, 115:121–129.
- Yuan, H. and He, B. (2014). Brain-computer interfaces using sensorimotor rhythms: Current state and future perspectives. *IEEE Transactions on Biomedical Engineering*, 61(5):1425–1435.
- Zadeh, Elham Akhond, N., Bijan, R., Amir, A., and H, Behrouz, F. (2017). *Handbook of Research on Data Science for Effective Healthcare Practice and Administration*. IGI Global.
- Zaki, M. J. and Meira Jr, W. (2019). *Data Mining and Machine Learning: Fundamental Concepts and Algorithms*. Cambridge University Press.
- Zannatha, J. M. I. (2012). *Vision Artificial Para Robots*. Eae Editorial Academia Espanola, Place of publication not identified.
- Zhang, X., Vialatte, F.-B., Chen, C., Rathi, A., and Dreyfus, G. (2015). Embedded Implementation of Second-Order Blind Identification (SOBI) for Real-Time Applications in Neuroscience. *Cognitive Computation*, 7(1):56–63.

-
- Zhang, Z., Huang, Y., Chen, S., Qu, J., Pan, X., Yu, T., and Li, Y. (2017). An intention-driven semi-autonomous intelligent robotic system for drinking. *Frontiers in neurorobotics*, 11:48.
- Zhou, B., Wu, X., Zhang, L., Lv, Z., and Guo, X. (2014). Robust Spatial Filters on Three-Class Motor Imagery EEG Data Using Independent Component Analysis. *Journal of Biosciences and Medicines*, 02(02):43–49.

Anexo A

Artículo de conferencia COMROB 2019

Development of a Vision-Assisted Robotic Manipulation System

A.D. Hernández-Rojas¹, J.M. Ibarra-Zannatha² and J. Cantillo-Negrete³

Abstract—In recent years, Brain-Computer Interface (BCI) research has increased considerably, however, real-life applications are still challenging due to the legal framework that this implies coupled with technical difficulties. Therefore, it is necessary to develop versatile and low-cost systems for these technologies to solve real-life problems, such as aiding disabled people in improving daily life movement-related tasks. This article presents the development of an experimental platform for the manipulation of objects, where the position of the objects to be manipulated is obtained through a vision algorithm. The robot uses Dynamixel actuators and a gripper as end-effector, while the vision system uses an RGB-camera that can be located in any place from which the worktable can be seen completely. We present the artificial vision process, the kinematic model of the robot, as well as its implementation through Robot Operating System (ROS). This system is intended to be used in conjunction with a Motor-Imagery (MI) based BCI, in order to perform a pick & place task of a desired object, using patient's MI. However, it could be used for other tasks as well, such as a testbed for learning ROS, control theory or robotics.

I. INTRODUCTION

The implementation of a brain-computer-interface (BCI) for rehabilitation is basically comprised by the recording and decoding of local brain signals generated by the patient, as he/she tries to perform a particular task, or during a mental imagery task [1]. Motor imagery (MI) is a state in which a subject imagines a specific movement, without executing it. For motor recovery, the kinaesthetic imagery is an alternative as it activates the motor network of the brain. The *BCI-Rob* project is comprised by the development of these types of systems. This project's objective consists of recognizing between two mental states (LEFT/RIGHT movement) of a subject through MI-based BCI and using this command to control a Vision-Assisted Robotic Manipulation System (VARMS). A related research to the *BCI-Rob* project is presented in [2], which presents an MI-based BCI used to control a robot arm in order to reach and grasp an object in a 3D space. This movement was achieved through a sequential movement strategy, but the time required to perform the complete task was significant. Regarding the use of artificial vision techniques to obtain the position of the objects, in [3] an artificial vision algorithm is proposed to obtain the position (x, y) of objects using a Homography matrix. More recent research has put forward new strategies for manipulation, such as in the one proposed in [4], in which

¹CINVESTAV, Department of Automatic Control, 07360 Mexico City, Mexico alex.io_c@hotmail.com

²CINVESTAV, Department of Automatic Control, 07360 Mexico City, Mexico jibarra@cinvestav.mx

³Instituto Nacional de Rehabilitación, Division of Medical Engineering Research, 14389 Mexico City, Mexico jessica.cantillo.negrete@gmail.com

a shared control strategy (BCI & computer vision) is used to perform a reach and grasp task in a 3D space. In this system, the subjects can move a robot arm in the surrounding area of the target by performing two different mental states, but once the end-effector of the robot enters a vision-guided region, the robot arm grasps the target autonomously.

In this paper we propose the development of the VARMS for the *BCI-Rob* project, and it works as follows: Given a pair of objects and a basket, all of them placed randomly on a worktable, a vision-guided robot takes the object corresponding to the side that the patient thought in order to place it in the basket.

The basic requirements for the robot of the VARMS are:

- 1) *Workspace*: Cover the space of a $40\text{cm} \times 40\text{cm}$ worktable with min. height: 2.5cm (height to take the object) and max. height: 10cm (height during the transportation of the targeted object).
- 2) *Mechanical structure*: 5-DOF (enough to take polyhedral objects (straight prisms) on a work table), three for positioning and two for orientation of the gripper.
- 3) *Actuation*. Enough torque to perform the proposed task.

The rest of the paper is organized as follows: In [Sec. II](#) each part of the ROS-based framework is explained. The [Sec. III](#) is dedicated to the results obtained from simulation and experimentally. Finally [Sec. IV](#) includes the conclusions.

II. METHODS

A. System Overview

The general architecture of the system for the *BCI-Rob* project is shown in [Fig. 1](#). It consists of two subsystems: the BCI system and the VARMS (i.e. robot system and the vision system). Once the command signal from BCI is generated, the vision system sent the pose data of the desired object and the basket to the robot system where a planning strategy is computed to perform the corresponding pick & place task. We used Robot-Operating-System (ROS) as a knowledge base to exchange data of all subsystems. Below we explain the subsystems corresponding to VARMS.

B. Vision system

An RGB camera with a resolution of 800×600 pixels at a frame rate of 30 Hz was chosen for the vision system. The camera could be located in an arbitrary position as long as the entire surface of the worktable was visible. We placed regular square prisms with a width of 2cm and height of 2.9cm with its upper face colored at the corners of the worktable as markers for a geometric calibration phase. The objects to be manipulated were as the previously described but with

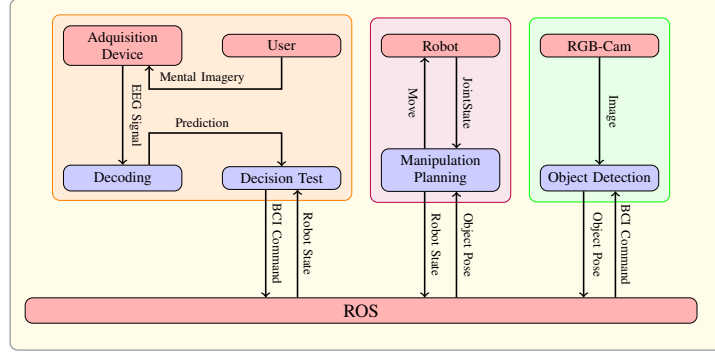


Fig. 1: Overview of our framework. The BCI system (orange box) generates the command signal through the motor imagery paradigm. When the vision system (green box) detects it, the pose of the selected object and the basket are sent to the robot system (pink box) to generate the path corresponding to the pick & place task which is executed by the robot.

different upper face color, while the basket had a height of 3.5cm with its upper face of another color.

Our system was implemented under Ubuntu 16.04 and ROS using OpenCV-C++ and Qt libraries. It has two fundamental stages: *processing* and *analysis*. In the *processing* stage two algorithms were implemented, one for filtering and one for segmentation. The first one allows to eliminate lighting problems such as shadows and excessive brightness on the worktable, as well as ignoring the grid lines on the worktable surface. The second one allows discriminating by color criteria and extracting the areas of interest from the images, i.e., the area of the object to be manipulated and the basket. As for the *analysis* stage, it includes two phases, one off-line corresponding to the geometric calibration, which consists in the identification of the Homography matrix, used to correct the perspective distortion, as well as to transform the pixel locations of an object in the image into its corresponding position and orientation on the worktable in centimeters. The second phase is applied online and includes the measurement of the position and orientation of the objects on the worktable, for which it is required to have the Homography matrix previously obtained, since it must be done in an image in which the perspective distortion has been corrected. The basic theory of the homography matrix is explained below.

The perspective projection model allows obtaining the projection of a 3D point, \mathbf{M}_0 vector, onto a (2D) image point, \mathbf{w} vector, according to the equation:

$$\begin{aligned} \lambda \mathbf{w} &= \mathbf{P} \mathbf{M}_0 \\ \lambda \mathbf{w} &= \mathbf{K} [\mathbf{R} \quad \mathbf{t}] \mathbf{M}_0 \\ \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \begin{bmatrix} k_x & k_\theta & C_x \\ 0 & k_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \end{aligned} \quad (1)$$

where \mathbf{P} is called projection matrix, which is a function of the calibration matrix \mathbf{K} of the camera and the matrix that represents the position, \mathbf{t} vector, and the orientation,

\mathbf{R} matrix, of the camera with respect to the world reference frame R_0 ; \mathbf{w} and \mathbf{M}_0 are expressed in homogeneous form; λ is a non-zero scale factor, commonly called projective depth. In the special case where the points of interest of the scene lie on a plane ($Z = 0$), without loss of generality, the perspective projection model is reduced to:

$$\begin{aligned} \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \begin{bmatrix} k_x & k_\theta & C_x \\ 0 & k_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \\ \lambda \mathbf{w} &= \mathbf{H} \mathbf{M}'_0 \end{aligned} \quad (2)$$

where the point of the scene, \mathbf{M}'_0 vector, is now two-dimensional and is projected onto the image according to the so-called Homography \mathbf{H} matrix. In general, this matrix has 8-DOF and therefore it can be identified with at least four known point correspondences. In our case, we used four points, the outer corners of the markers placed on the worktable. As what is required is to transform the image's points (in pixels) into points on the plane parallel to the worktable (in centimeters), it is necessary to find \mathbf{H}^{-1} . Once this matrix is identified, it is possible to make the measurements required by the robot, regardless of the distortion in perspective.

We summarize the steps of the vision algorithm proposed as follows:

1) Image Analysis: Geometric calibration

- a) Four points corresponding to the 4 external corners of markers placed on the worktable are manually selected, whose positions in the world reference correspond to: (0,-20) (0,20) (40,20) (40,-20) in cm units.
- b) The inverse homography matrix is computed using the OpenCV function `findHomography()`.

- 2) Perspective correction image is obtained through Homography matrix
- 3) Image Processing
 - a) A bilateral filter is applied to soften the image, without losing edges.
 - b) Segmentation process
 - i) Image conversion from RGB to HSV.
 - ii) Binarization in HSV space of color manually selected in frame.
 - iii) Application of a median filter to remove noise.
 - iv) Combination of segmented color images into a single image.
- 4) Image Analysis: 2D Pose estimation process
 - a) In order to locate the objects (selected object and basket), the contours in the binary image were obtained and discriminated according to their area and the number of vertices of an approximate contour.
 - b) To determine the centroid of each object, we used the property of *centre of mass* of image moments of the approximate contours. The center of mass (\bar{x}, \bar{y}) of an image is defined as the weighted average of the pixels and computed from the zeroth and first moments [5]:

$$\begin{aligned} (\bar{x}, \bar{y}) &= \frac{1}{\sum_{x,y} f(x,y)} \left(\sum_{x,y} xf(x,y), \sum_{x,y} yf(x,y) \right) \\ &= \begin{pmatrix} m_{10} & m_{01} \\ m_{00} & m_{00} \end{pmatrix}. \end{aligned} \quad (3)$$
 - c) The orientation of the object to be taken was also obtained. This angle was obtained using the OpenCV function `minRect.angle()`. This function allows obtaining the orientation of the edge that goes from the lowest vertex of the polygon (base of the prism) towards the vertex located at the top right. This angle could have been found using image moments as indicated in [6], however, this path was chosen because with this angle we can determine the *yaw* angle of the gripper.
 - d) So that the robot takes the objects with the minimum orientation with respect to the center of the worktable, we applied the following rule:

Rule 1: Orientation of the objective (1 or 2) object

Input: Angle obtained with OpenCV
function `minRect.angle()`: δ

Output: *yaw* angle for robot: ϕ

if $y_{objective} < 0$ **then**
 | $\phi = - (90 - |\delta|)$
else
 | $\phi = |\delta|$
end

where $y_{objective}$ is the y-position of the objective object.

C. Robot system

This system consists of the robot arm and the algorithm for planning the pick & place task. We build a 5-DOF robot for this project because the task-space for taking the object laterally is defined by $x, y, z, pitch, yaw$ ($T \subset \mathbb{R}^3 \times \mathbb{S}^2$), i.e., to get any position in the task space, 5-DOF are required. Dynamixel servomotors were chosen for robot actuation. The maximum workspace radius of the robot is 45cm. The gripper used has a stroke up to 32mm. The robot system ran on Ubuntu 16.04 and ROS using Python and Qt libraries.

• Robot Design

The criterion to select the lengths of the links was that the stall torque of the servomotors was less than or equal to 1/5 of its maximum, as the manufacturer recommends. For this, a moment calculation was made. At the same time, we made the CAD design of each robot part, shown in Fig. 2. Then, 3D printing of the longest links was made, in addition to aluminum cutting to build the robot structure, which was finally assembled as it is shown in Fig. 3.

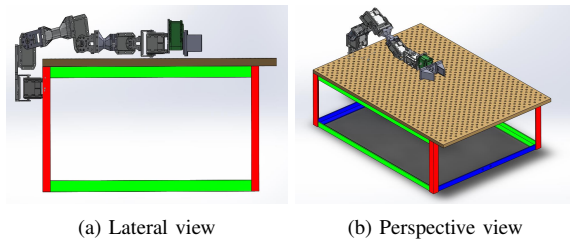


Fig. 2: CAD Design

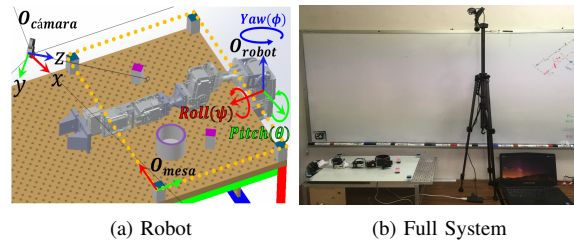


Fig. 3: Developed system

For an n -degree of mobility manipulator the reachable workspace are the points that can be achieved by considering the forward kinematics only for the position part, i.e.:

$$\mathbf{x}_p = f(\mathbf{q}), \quad q_{im} \leq q_i \leq q_{iM}, i = 1, \dots, n \quad (4)$$

where $q_{im}(q_{iM})$ denotes the minimum (maximum) limit at joint i [7]. In order to verify that the proposed robot performs the desired task, the reachable workspace was determined by forward kinematics. Fig. 4 shows the reachable workspace of the robot obtained with the MATLAB Robotics toolbox, where the workspace is the red volume, the worktable is the rectangular area in yellow, and in pink the task-space.

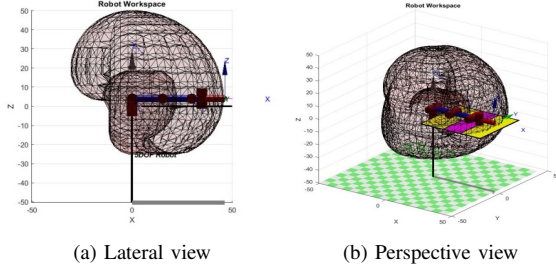


Fig. 4: Robot's workspace

• Kinematic Modeling

We compute the forward kinematics of the robot, which consists of determining the relationship between the pose of the final effector of the robot given its articular coordinates. The most common mathematical tool to express the position of a referential is the matrix (4x4) of homogeneous transformation. A commonly used convention for selecting frames of reference in robotic applications is the Denavit-Hartenberg or DH convention. In this convention, each homogeneous transformation is represented as a product of four basic transformations:

$$T_{i-1}^i = Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i} \quad (5)$$

where $\theta_i, a_i, d_i, \alpha_i$ are parameters associated with link i , named *joint angle*, *link length*, *link offset* and *link twist*, respectively [8]. Finally, the transformation matrix from link 0 to link n is obtained by composition of transformations between consecutive frames. For our robot, the frame assignment according to the DH convention is established as shown in Fig. 5.

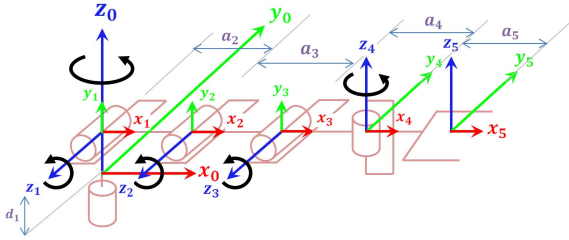


Fig. 5: Robot's frame assignment

Using the D-H parameters of the manipulator, shown in Tab. I, each of the homogeneous transformations between the references was determined, finally obtaining the one that relates the position of the final effector with the fixed reference system of the base, denoted by Eq. 6.

We decided to use a minimal representation of the pose to control the robot. For this, the rotation was parameterized at Cardan XYZ angles, also known as *roll-pitch-yaw* angles, which defines an orientation in space by three successive rotations around the fixed base reference: first a rotation around x_0 an angle ψ (*roll*), then a rotation around y_0 an angle θ (*pitch*), and finally a rotation around z_0 an angle ϕ (*yaw*), i.e., $R_{ZYX} = R_{z,\phi}R_{y,\theta}R_{x,\psi}$. The inverse problem, that is, calculating *roll-pitch-yaw* angles given the rotation matrix R_{ZYX} , has two possible cases as indicated in [9].

TABLE I: DH parameters of the robot

Link	a_i	α_i	d_i	θ_i
1	0cm	90°	4cm	θ_1^*
2	13.0cm	0°	0cm	θ_2^*
3	11.7cm	0°	0cm	θ_3^*
4	5.0cm	-90°	0cm	θ_4^*
5	11.5cm	0°	0cm	θ_5^*

The problem of inverse kinematics consists of finding the joint variables that generate a given position and orientation of the end effector. Because it is required to take the object laterally, the problem of inverse kinematics can be posed as: Given the pose of the end-effector, such as:

$$\mathbf{X} = [x \ y \ z \ 0 \ 0 \ \phi_{z0}]^T, \quad (7)$$

a closed solution is sought such that it generates physically realizable movements, without colliding with the worktable. The first three joints allow the robot to be positioned in space. With respect to joint 4, it is sought to establish its articular variable such that its referential remains parallel to the worktable maintaining *pitch* in zero, this because the joint 4 and 5 do not intersect as in a spherical wrist. Finally, joint 5 generates the orientation needed to take the object. In this way, the problem of inverse kinematics can be divided into two sub-problems, since the solution for the position is decoupled from the orientation. Next, we present each of the equations of the joint variables based on the geometric approach, according to the following steps:

- Calculation of wrist position 2 (\mathbf{p}_{w2})
- Solve inverse kinematics for (q_1, q_2, q_3)
- Solve inverse kinematics for (q_4)
- Solve inverse kinematics for orientation (q_5)

From the vectors in the reference frame of the gripper, the position of the reference frame of the joint 5 is:

$$\mathbf{p}_{w2} = \mathbf{p}_0^5 - a_5 \mathbf{a} \quad (8)$$

From Fig. 6a it is observed that θ_1 is defined by

$$\theta_1 = \text{atan2} \left(\frac{p_{w2y}}{p_{w2x}} \right). \quad (9)$$

For θ_2 and θ_3 we solved as a planar case. Fig. 6b shows the projection of the center of the wrist 1 on the $x_0 - y_0$ plane, from which we can define $r = \sqrt{p_{w1x}^2 + p_{w1y}^2}$ and Fig. 6c shows the previous position but in the plane $x_0 - z_0$, from which it is obtained $s = p_{w1z} - d_1$. Using the law of cosines $h^2 = a_2^2 + a_3^2 - 2a_2a_3$, and the trigonometric properties $\cos(a-b) = \cos a \cos b + \sin a \sin b$ and $\tan \frac{\theta}{2} = \pm \sqrt{\frac{1-\cos \theta}{1+\cos \theta}}$ it follows that θ_2 and θ_3 are defined by:

$$\theta_2 = \alpha \pm \beta = \text{atan2} \left(\frac{s}{r} \right) \pm \text{atan2} \left(\frac{a_3 \sin \theta_3}{a_2 + a_3 \cos \theta_3} \right) \quad (10)$$

$$T_0^5 = T_0^1 T_1^2 T_2^3 T_3^4 T_4^5 = \begin{bmatrix} -s_1 s_5 + c_1 c_5 c_{234} & -s_1 c_5 - s_5 c_1 c_{234} & -s_{234} c_1 & -11.5 s_1 s_5 + 13.0 c_1 c_2 + 11.5 c_1 c_5 c_{234} + 11.7 c_1 c_{23} + 5 c_1 c_{23} \\ s_1 c_5 c_{234} + s_5 c_1 & -s_1 s_5 c_{234} + c_1 c_5 & -s_1 s_{234} & 13.0 s_1 c_2 + 11.5 s_1 c_5 c_{234} + 11.7 s_1 c_{234} + 5 s_1 c_{234} + 11.5 s_5 c_1 \\ s_{234} c_5 & -s_5 s_{234} & c_{234} & 13.0 s_2 + 11.7 s_{23} + 11.5 s_{234} c_5 + 5 s_{234} + 4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

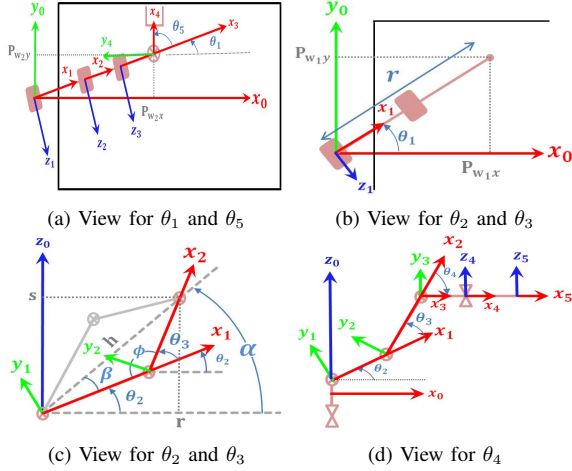


Fig. 6: Robot positions to solve inverse kinematics

$$\theta_3 = \pm 2 \operatorname{atan2} \left(\frac{\sqrt{(a_2 + a_3)^2 - (r^2 + s^2)}}{\sqrt{(r^2 + s^2) - (a_2 - a_3)^2}} \right) \quad (11)$$

respectively, where we select '-' sign for *elbow up*. As we are considering that there is only orientation in the *yaw* angle, the *x*-axis of joint 4 must remain parallel to the worktable, therefore θ_4 is defined by (see Fig. 6d):

$$\theta_4 = -(\theta_2 + \theta_3). \quad (12)$$

Finally, from Fig. 6a, θ_5 is expressed by:

$$\theta_5 = \phi - \theta_1 \quad (13)$$

Note: $\operatorname{atan2}(y, x)$ is a four-quadrant arc tangent function that computes $\tan^{-1}(y/x)$ but uses the sign of both x and y to determine the quadrant in which the resulting angle lies.

• Control and planning

To control the robot arm we proposed a kinematic control architecture, as shown in Fig. 7, in which there is an independent control for each joint, which for Dynamixel actuators is a PID controller already included in its structure. Next, we briefly describe the process of planning and control used. First, due to the 2D pose received from the vision system is in the referential frame of the worktable in mm, see Fig. 3, we transform it to cm. Then, we transform it to the referential

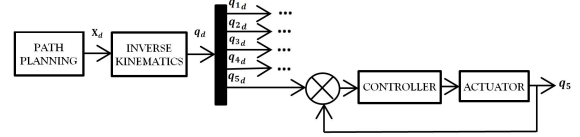


Fig. 7: Control architecture for manipulator

frame of the robot through the TH matrix:

$$T_{robot}^{worktable} = \begin{bmatrix} 0 & -1 & 0 & 55 \\ -1 & 0 & 0 & 20 \\ 0 & 0 & -1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

Then, we defined the pose of the points corresponding to the pick & place task given by:

- 1) Go to $(x_{obj}, y_{obj}, 15, yaw_{obj})$
- 2) Go to $(x_{obj}, y_{obj}, 3.5, yaw_{obj})$ and close gripper
- 3) Go to $(x_{obj}, y_{obj}, 15, yaw_{obj})$
- 4) Go to $(x_{goal}, y_{goal}, 15, 0)$ and open gripper
- 5) Go to home position of the robot

and we transform each of these points into a homogeneous transformation. Next, the generation of trajectory in the cartesian space between each pose of the task was achieved using the spherical linear interpolation (SLERP) algorithm coded in `ctrjaj` function from Python Robotics Toolbox. To obtain the trajectory points in joint coordinates, inverse kinematics was applied to the cartesian trajectory points. Finally, the position control was achieved by using the *joint mode* control of Dynamixel actuators.

D. ROS Integration

The Robot Operating System (ROS) is an open-source framework targeted for writing robot software. ROS has the ability to divide the structure in small modules called "nodes", which can communicate between them through a master node. A node publishes data under a topic and other nodes can use this data by subscribing to that topic. We used ROS-Kinetic over Ubuntu 16.04 to implement our system, which consists of four packages. The robot package with a corresponding node to establish the parameters of each actuator. The vision system package to send the pose of the objects to the robot through a custom message. The planning and control package to receive the data from the vision system and generate the trajectory points necessary for the task and sending the corresponding commands to the robot system. To achieve this, we used a Joint Trajectory

Action Controller (JTAC), which is designed to accept an entire joint trajectory as input. A joint trajectory is a list or sequence of joint positions, velocities, accelerations, and efforts for each joint over a period of time. Finally, we created the BCI system package that sends a synthetic command to indicate LEFT/RIGHT movement, as long as the robot is not busy.

III. RESULTS

The theory presented together with the implementation resulted in the successful execution of the pick & place task given a synthetic signal emulating the BCI command. Fig. 8 shows frames of the main stages of our artificial vision process explained in Subsec. II-B. From left to right, the input frame, perspective correction, segmentation and the estimation of the 2D pose of the objects on the image.

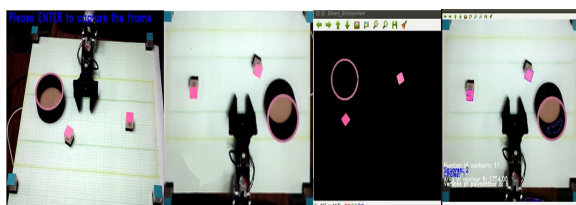


Fig. 8: Main stages of vision algorithm

We validated the movement to be executed by the manipulator through a 5-DOF robot simulator designed using PyQt, shown in Fig. 9. So, if the robot could kinematically execute the task, then it performs it. Also, we compared the desired and real cartesian trajectory corresponding to an object pose of (25cm, -12.4cm, 75°) and a basket pose of (26.5cm, 11.5cm, 0°) as shown in Fig. 10, where we can observe that despite the use of a kinematic control, the error is almost zero in $x, y, roll$ and yaw variables but not in $pitch$ and z . Despite that, the pick & place task was not affected at its completion.

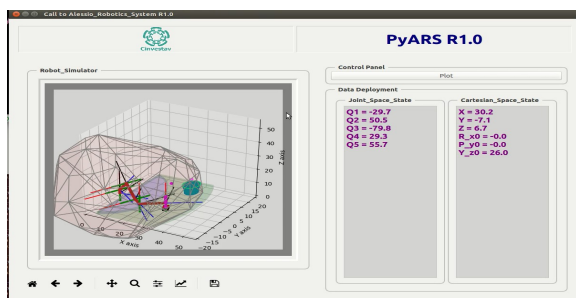


Fig. 9: Robot Simulator

IV. CONCLUSIONS

This article addressed the development of a ROS-based object manipulation system in a 2D space, to control a 5-DOF robot integrated with a vision system. Experimental

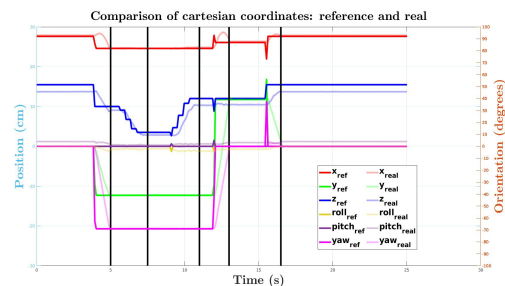


Fig. 10: Comparison of desired and real cartesian trajectory

results were satisfactory as the system could perform the pick & place task of a chosen object. Our vision system is not free to errors, for example, light changes degenerate the segmentation affecting the later stages, which is why as future work we propose to use robust techniques for the localization process. Concerning the robot, we propose to use a spherical wrist to simplify the kinematic modeling, in addition to using a dynamic model so that to get zero error in the full state vector as well as to demonstrate system stability. As mentioned before, this system will be integrated with an MI-based BCI, to finally allow a subject/patient to manipulate objects through mental commands. However, it will also be possible to use it as a testbed for learning ROS, robotics, control theory, as well as artificial vision.

SUPPLEMENTARY MATERIAL

We provide a video where we show a test of the developed system. <https://youtu.be/ut32Vvk9I4ew>

REFERENCES

- [1] L. E. H. Van Dokkum, T. Ward, and I. Laffont, "Brain computer interfaces for neurorehabilitation its current status as a rehabilitation strategy post-stroke," *Annals of physical and rehabilitation medicine*, vol. 58, no. 1, pp. 3–8, 2015.
- [2] J. Meng, S. Zhang, A. Bekyo, J. Olsoe, B. Baxter, and B. He, "Noninvasive Electroencephalogram Based Control of a Robotic Arm for Reach and Grasp Tasks," *Scientific Reports*, vol. 6, no. 1, p. 38565, Dec. 2016.
- [3] M. A. Ramirez-Moreno, S. M. Orozco-Soto, J. M. Ibarra-Zannatha, and D. Gutierrez, "Artificial Vision Algorithm for Object Manipulation with a Robotic Arm in a Semi-Autonomous Brain-Computer Interface," in *Wearable Robotics: Challenges and Trends*, Springer International Publishing, 2019, pp. 187–191.
- [4] Y. Xu, C. Ding, X. Shu, K. Gui, Y. Bezsudnova, X. Sheng, and D. Zhang, "Shared control of a robotic arm using non invasive brain computer interface and computer vision guidance," *Robotics and Autonomous Systems*, vol. 115, pp. 121–129, May 2019.
- [5] S. Birchfield, *Image processing and analysis*. Cengage Learning, 2016.

- [6] S. X. Liao, "Image Analysis by Moments," Ph.D. dissertation, The University of Manitoba, Manitoba, Canada, Jun. 1993.
- [7] L. Sciavicco and B. Siciliano, *Modelling and control of robot manipulators*. Springer Science & Business Media, 2012.
- [8] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. Wiley, Nov. 2005, ISBN: 978-0-471-64990-8.
- [9] H. D. Taghirad, *Parallel Robots: Mechanics and Control*. CRC Press, Feb. 2013.