



UNIVERSIDAD ESTATAL
PENÍNSULA DE SANTA ELENA

FACSISTEL

INGENIERÍA EN ELECTRÓNICA Y
AUTOMATIZACIÓN

TRABAJO DE INTEGRACIÓN CURRICULAR

Diseño y desarrollo de un módulo de
emulación de señales cerebrales y su
integración con un modelo de brazo
robótico virtual de cuatro grados de
libertad en Simscape Multibody

Josué Francisco González Tomalá

Dirigido por:

Ing. Edison Coral, Mgtr.

La Libertad - 2026

DEDICATORIA

El presente trabajo de titulación está dedicado, con el más profundo amor y gratitud, a las personas que han sido el sostén de mi vida y la razón de cada uno de mis pasos. A Dios, por ser mi guía en cada paso, mi fortaleza en los momentos de duda y la luz que iluminó este camino. Gracias por darme la sabiduría, la salud y las fuerzas necesarias para culminar esta etapa, y por rodearme de personas maravillosas que hicieron posible este sueño. Con todo mi amor a mi madre, por ser el corazón que nunca dejó de latir por mí. Por sus noches de desvelo, sus oraciones silenciosas y ese abrazo capaz de calmar cualquier tormenta. Gracias por enseñarme con tu ejemplo lo que significa la fortaleza, la ternura y el amor sin condiciones. Cada logro mío es también tuyo, porque nada de esto habría sido posible sin tu entrega infinita. Con profunda admiración a mi padre, por ser mi ejemplo de esfuerzo, constancia y sacrificio. Por cada consejo, cada palabra de aliento y cada lección que me forjó como la persona que soy hoy. Gracias por enseñarme que los sueños se alcanzan con trabajo y perseverancia, y por sostenerme, incluso en silencio, en cada paso de este camino. A mis hermanos, por acompañarme, creer en mí y ser mi refugio en los días difíciles. De igual manera, a mi familia: Kerly Tomalá, Juan González, Kevin Condoy y Magdalena González, quienes con su cariño, su apoyo y su presencia constante formaron parte fundamental de mi formación tanto personal como académica. Gracias por sostenerme cuando más lo necesité. Con especial cariño a mis mejores amigos, Anthony Torres y Junior Figueroa, por caminar a mi lado en cada etapa, por la lealtad, las risas y la certeza de saber que nunca estuve solo. Su amistad es uno de los tesoros más grandes que me deja este camino. A todos ellos les dedico cada página de este documento, como testimonio de que su entrega, su confianza y su amor no fueron en vano.

AGRADECIMIENTO

Quiero expresar mi más sincero agradecimiento a todas las personas e instituciones que contribuyeron de manera significativa en la realización de este trabajo. En primer lugar, mi más profundo y especial agradecimiento al Ing. Edison Coral, tutor del presente proyecto, cuya guía fue esencial en cada etapa de esta investigación. Su orientación técnica, su permanente disposición y su valioso conocimiento fueron determinantes para estructurar y desarrollar cada componente del sistema propuesto. Gracias por su paciencia, por creer en este proyecto, por cada corrección oportuna y por compartir con generosidad su experiencia; sin su acompañamiento constante, este logro no habría sido posible. Es para mí un verdadero honor haber contado con su dirección. De igual manera, agradezco a la Universidad Estatal Península de Santa Elena (UPSE) y a la Facultad de Sistemas y Telecomunicaciones (FAC-SISTEL) por brindar los espacios y recursos necesarios para llevar adelante la investigación. Asimismo, reconozco la labor de los docentes del área de Electrónica y Automatización, ya que sus enseñanzas a lo largo de la carrera sentaron las bases técnicas que me permitieron abordar un proyecto de esta naturaleza. Finalmente, agradezco a mis compañeros y amigos, con quienes compartí conocimientos, experiencias y momentos de aprendizaje durante el desarrollo de este sistema.

APROBACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de titulación denominado: “**Diseño y desarrollo de un módulo de emulación de señales cerebrales y su integración con un modelo de brazo robótico virtual de cuatro grados de libertad en Simscape Multibody**”, elaborado por el estudiante **Josué Francisco González Tomalá**, de la carrera de **Ingeniería en Electrónica y Automatización** de la Universidad Estatal Península de Santa Elena, declaro que luego de haber orientado, estudiado y revisado, lo apruebo en todas sus partes y autorizo al estudiante que inicie los trámites legales correspondientes.

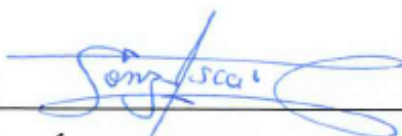


Ing. Edison Stalin Coral Salinas, Mgtr.
Tutor

TRIBUNAL DE GRADO



Ing. Ronald Rovira, Ph.D.
DIRECTOR DE CARRERA



Ing. Óscar W. Gómez Morales, Ph.D.
DOCENTE ESPECIALISTA



Ing. Edison Stalin Coral Salinas, Mgtr.
TUTOR



Ing. Luis Enrique Chuquimarca Jiménez, Ph.D.
DOCENTE GUÍA UIC



Ing. Corina Gonzabay De La A, Mgt.
SECRETARIA DEL TRIBUNAL

RESUMEN

El presente trabajo de integración curricular tiene como propósito el diseño y la validación de una plataforma de simulación y emulación de señales neurofisiológicas, orientada al control de un brazo robótico virtual de cuatro grados de libertad dentro del marco de la interfaz cerebro-computadora. Dado que la adquisición de señales electroencefalográficas reales mediante el casco OpenBCI no pudo concretarse durante el periodo de ejecución, la investigación se reorientó hacia el desarrollo de un módulo físico capaz de representar el comportamiento de las bandas neuronales que habitualmente se emplean en los sistemas BCI. Para ello, se construyó una interfaz electrónica basada en una placa de circuito impreso propia, en la cual un conjunto de potenciómetros reproduce la intensidad de los ritmos delta, theta, alfa y beta, mientras que un pulsador permite confirmar el comando que se envía al manipulador. La información generada se muestra en una pantalla SPI que indica la banda activa, el comando seleccionado y el estado general del sistema, y luego se transmite hacia MATLAB a través de comunicación serial. El brazo robótico se modeló con una configuración antropomórfica de cuatro articulaciones rotacionales, cuyos ángulos fueron corregidos e incorporaron límites máximos y mínimos coherentes con el movimiento humano, de modo que se evitaron posturas imposibles desde el punto de vista cinemático. Posteriormente, la integración del sistema se llevó a cabo en MATLAB, Simulink y Simscape Multibody, entorno en el que se reprodujo el comportamiento dinámico del manipulador frente a los comandos emulados. De esta manera, la plataforma se plantea como una herramienta de bajo costo que permite estudiar la interacción entre los sistemas BCI y la robótica asistida antes de incorporar hardware EEG real, además de constituir un recurso útil para la enseñanza y la investigación en neuroingeniería. **Palabras clave:** interfaz cerebro-computadora, emulación de señales, bandas neuronales, potenciómetros, PCB, Simscape Multibody, brazo robótico virtual, simulación.

ABSTRACT

This curricular integration project aimed to design and validate a simulation and emulation platform for neurophysiological signals oriented towards the control of a virtual four-degree-of-freedom robotic arm within the framework of brain-computer interfaces. Since the acquisition of real electroencephalographic signals using the OpenBCI headset could not be carried out during the execution period, the research was redirected towards the development of a physical module able to represent the behaviour of the neural bands commonly used in BCI systems. For this purpose, an electronic interface based on a custom printed circuit board was built, in which a set of potentiometers reproduces the intensity of the delta, theta, alpha and beta rhythms, while a pushbutton allows the user to confirm the command sent to the manipulator. The generated information is shown on a 240×320 SPI TFT screen that indicates the active band, the selected command and the general state of the system, and is then transmitted to MATLAB through serial communication. The robotic arm was modelled with an anthropomorphic configuration of four rotational joints whose angles were corrected and provided with maximum and minimum limits consistent with human movement, thus avoiding postures that are impossible from a kinematic point of view. The system was integrated in MATLAB, Simulink and Simscape Multibody, where the dynamic behaviour of the manipulator was reproduced in response to the emulated commands. In this way, the platform is proposed as a low-cost tool that makes it possible to study the interaction between BCI systems and assistive robotics before incorporating real EEG hardware, and it also constitutes a useful resource for teaching and research in neuroengineering. **Keywords:** brain-computer interface, signal emulation, neural bands, potentiometers, PCB, Simscape Multibody, virtual robotic arm, simulation.

Índice

Índice de figuras	10
Índice de tablas	12
1 Introducción	14
1.1 Justificación	14
1.2 Planteamiento del problema	15
1.3 Panorama actual	15
1.4 Objetivos	16
1.4.1 Objetivo general	16
1.4.2 Objetivos específicos	16
1.5 Fundamentos teóricos	17
1.5.1 Interfaz cerebro-computadora (BCI)	17
1.5.2 Ritmos cerebrales y bandas neuronales	18
1.5.3 Emulación de señales y plataformas de simulación de bajo costo	18
1.5.4 Cinemática y dinámica de manipuladores robóticos . .	19
1.5.5 Estrategias de control en entornos de simulación	20
1.5.6 Simscape Multibody como entorno de modelado diná- mico	21
1.6 Software utilizado	22
1.7 Bloques de MATLAB/Simulink	22
2 Métodos y diseño experimental	23
2.1 Tipo de investigación	23
2.2 Investigación aplicada	23
2.3 Investigación descriptiva	23
2.4 Investigación exploratoria	24
2.5 Investigación basada en simulación	24
2.6 Descripción general del sistema propuesto	24
2.7 Descripción general del módulo de emulación	25
2.8 Potenciómetros como representación de las bandas neuronales	26
2.8.1 Pulsador de confirmación de comando	27
2.8.2 Pantalla SPI: visualización de banda, comando y estado	28
2.8.3 Microcontrolador y acondicionamiento de señal	28
2.9 Acondicionamiento de la señal	28
2.10 Comunicación serial con MATLAB	29
3 Desarrollo experimental	31

3.1	Modelado del robot virtual	31
3.1.1	Configuración antropomórfica de cuatro grados de libertad	31
3.1.2	Diseño CAD del manipulador	31
3.1.3	Parámetros de Denavit–Hartenberg	33
3.1.4	Cinemática directa mediante la convención de Denavit–Hartenberg	33
3.1.5	Parámetros DH obtenidos del modelo CAD	34
3.1.6	Matrices de transformación homogénea	36
3.2	Cinemática inversa del manipulador	37
3.2.1	Corrección de ángulos y definición de límites articulares	40
3.2.2	Mapeo lineal del ADC al ángulo	40
3.2.3	Corrección de cero y límites articulares (<i>offset</i>)	40
3.3	Implementación en Simscape Multibody	42
3.3.1	Mapeo entre bandas emuladas y comandos articulares	43
3.4	Arquitectura de control óptimo y filtrado de la señal emulada	44
3.4.1	Cadena de procesamiento de señales	45
3.4.2	Zona muerta con arrastre (antirrebote)	45
3.4.3	Filtro 1€ (<i>One Euro</i>)	46
3.4.4	Filtro de Kalman	46
3.5	Regulador lineal-cuadrático (LQR)	46
3.5.1	Regulador lineal-cuadrático gaussiano (LQG) y estimador lineal-cuadrático (LQE)	47
3.6	Diseño CAD del brazo robótico	48
3.7	Diseño del circuito electrónico y la PCB	49
3.8	Diseño de la carcasa impresa en 3D	52
3.9	Programación del sistema de emulación	53
3.9.1	Configuración de la comunicación serial	53
3.10	Plan de validación experimental	54
3.10.1	Validación del módulo de emulación	54
3.10.2	Validación del control	55
3.10.3	Pruebas de usuario	55
3.10.4	Validación del comportamiento del manipulador	56
4	Resultados y discusión	56
4.1	Funcionamiento integral del módulo	56
4.2	Datos del módulo de emulación	57
4.3	Comunicación serial	58
4.4	Carcasa mediante impresión 3D	59
4.5	Escenarios de simulación	60
4.6	Control y simulación del manipulador	65

4.7	Discusión	69
4.8	Comparación con investigaciones previas	70
5	Conclusiones y recomendaciones	73
5.1	Conclusiones	73
5.2	Recomendaciones	74
	Referencias	76
	Referencias	76
	ANEXOS	80

Índice de figuras

Figura 1.	Diagrama general del sistema propuesto, desde el módulo de emulación hasta el brazo robótico simulado (Autoría propia, 2026).	25
Figura 2.	Cinemática y dinámica de manipuladores robóticos (Autoría propia, 2025).	32
Figura 3.	Modelo virtual del manipulador en el software CAD (Ing. Edison Coral, 2025).	32
Figura 4.	Dimensiones del manipulador de cuatro grados de libertad (Coral Salinas, 2025 [31]).	33
Figura 5.	Sistemas de coordenadas asignados a las articulaciones del manipulador (Coral Salinas, 2025 [31]).	35
Figura 6.	Esquema geométrico de las coordenadas articulares del manipulador (Coral Salinas, 2025 [31]).	38
Figura 7.	Modelo de control en Simulink y Simscape Multibody (Autoría propia, 2026).	43
Figura 8.	Diseño CAD del brazo robótico de cuatro grados de libertad, vista del ensamblaje de los cuatro eslabones (Autoría propia, 2026).	48
Figura 9.	Trazado de la placa de circuito impreso (PCB) del módulo de emulación (Autoría propia, 2026).	49
Figura 10.	Diagrama de conexiones entre el microcontrolador, los potenciómetros, el pulsador y la pantalla SPI (Autoría propia, 2026).	51
Figura 11.	Diagrama de bloques de la placa de circuito impreso (PCB) del módulo de emulación (Autoría propia, 2025).	51
Figura 12.	Diseño de la tapa de la carcasa impresa en 3D para el módulo de emulación (Autoría propia, 2026).	52
Figura 13.	Diseño de la base de la carcasa impresa en 3D para el módulo de emulación (Autoría propia, 2026).	52
Figura 14.	Segunda vista del módulo de emulación ensamblado, donde se aprecian los potenciómetros, el pulsador y la pantalla SPI en funcionamiento (Autoría propia, 2026).	57
Figura 15.	Carcasa impresa en 3D del módulo de emulación, vista isométrica (Autoría propia, 2026).	60
Figura 16.	Señal de control de la articulación α (codo) para el Escenario 1, mostrando el movimiento fluido y la estabilidad en el anclaje (Autoría propia, 2026).	63

Figura 17.	Señal de control de la articulación β (mano) para el Escenario 2, evidenciando la precisión y mitigación del ruido (Autoría propia, 2026).	63
Figura 18.	Señales de control de las articulaciones δ (cintura) y θ (hombro) para el Escenario 3, mostrando el movimiento síncrono y estable (Autoría propia, 2026).	64
Figura 19.	Señales de control de todas las articulaciones (θ , δ , α , β) para el Escenario 4, mostrando la trayectoria multi-articular continua y sin sobrepicos (Autoría propia, 2026).	65
Figura 20.	Señal de control de la cintura (θ) registrada en el Scope de Simulink (Autoría propia, 2026).	66
Figura 21.	Señal de control del hombro (δ) bajo las cuatro estrategias de control (Autoría propia, 2026).	67
Figura 22.	Señal de control del codo (α) ante movimientos de alta y baja velocidad (Autoría propia, 2026).	68
Figura 23.	Señales de control (LQG, LQR, LQE y Kalman) de la mano en movimiento, registradas en el Scope de Simulink (Autoría propia, 2026).	68
Figura 24.	Señales de control (LQG, LQR, LQE y Kalman) de la muñeca en movimiento: las cuatro señales convergen al ángulo comandado (Autoría propia, 2026).	92
Figura 25.	Señales de control de la mano y el codo en movimiento alto y bajo: seguimiento simultáneo de dos articulaciones (Autoría propia, 2026).	92
Figura 26.	Señales de control de la base y el brazo en movimiento: rotación de la cintura y elevación del hombro sin sobrepicos (Autoría propia, 2026).	93
Figura 27.	Esquemático y layout de la PCB del módulo de emulación (Autoría propia, 2026).	94
Figura 28.	Planos y render de la carcasa impresa en 3D (Autoría propia, 2026).	94
Figura 29.	Certificado de antiplagio del trabajo de integración curricular.	98

Índice de tablas

Tabla 1.	Justificación cinemática de la nomenclatura griega	26
Tabla 2.	Asignación de cada potenciómetro a una banda neuronal y a una articulación del manipulador.	27
Tabla 3.	Parámetros DH del brazo robótico	34
Tabla 4.	Parámetros Denavit–Hartenberg del manipulador de cua- tro grados de libertad.	35
Tabla 5.	Validación de la cinemática inversa para tres coordenadas de prueba (Coral Salinas, 2025 [31]).	39
Tabla 6.	Límites articulares definidos para el manipulador y su justificación técnica.	41
Tabla 7.	Parámetros y variables en el código	47
Tabla 8.	Componentes principales de la PCB del módulo de emu- lación.	50
Tabla 9.	Valores registrados por el módulo de emulación para cada banda neuronal.	57
Tabla 10.	Latencia de la comunicación serial entre el módulo de emulación y MATLAB.	58
Tabla 11.	Escenarios de simulación planteados para la validación del sistema.	61
Tabla 12.	Desempeño del control del manipulador por articulación.	65
Tabla 13.	Comparación con estudios de control robótico mediante BCI con señales EEG reales.	70
Tabla 14.	Comparación con estudios basados en potenciómetros, se- ñales simuladas y plataformas de simulación.	72
Tabla 15.	Posibles errores durante la operación del sistema y sus soluciones.	83

NOMENCLATURA Y ABREVIATURAS

Símbolo	Significado
BCI	Brain-Computer Interface (Interfaz Cerebro-Computadora)
EEG	Electroencefalografía / Señal electroencefalográfica
GDL / DOF	Grados de Libertad / Degrees of Freedom
DH	Convención de Denavit–Hartenberg
PCB	Printed Circuit Board (Placa de Circuito Impreso)
SPI	Serial Peripheral Interface (interfaz serie de la pantalla gráfica)
ADC	Conversor Analógico-Digital
PWM	Pulse Width Modulation (Modulación por Ancho de Pulso)
UART	Universal Asynchronous Receiver-Transmitter
$\delta, \theta, \alpha, \beta$	Bandas neuronales delta, theta, alfa y beta
CAD	Computer-Aided Design (Diseño Asistido por Computadora)
V	Voltaje
F	Frecuencia

1. Introducción

1.1. Justificación

El interés por desarrollar tecnologías capaces de vincular la actividad del cerebro con dispositivos externos ha crecido de manera notable en los últimos años, puesto que los sistemas de cerebro-computadora abren la posibilidad de traducir la actividad eléctrica neuronal en órdenes que un sistema mecánico o virtual puede ejecutar sin necesidad de movimiento muscular [1]. Sin embargo, llevar estos sistemas a la práctica exige contar con equipos de electroencefalografía cuyo costo y disponibilidad no siempre se ajustan a las condiciones de un laboratorio universitario, de modo que con frecuencia resulta difícil que el estudiante se familiarice con su funcionamiento antes de trabajar con hardware real. Por esta razón, el presente trabajo propone una alternativa de bajo costo, reproducible y libre de riesgo biomédico: en lugar de depender de la adquisición de señales EEG reales, que en este caso no pudo concretarse porque el casco OpenBCI no llegó dentro del periodo previsto, se diseñó un módulo físico que emula el comportamiento de las bandas neuronales utilizadas habitualmente en los sistemas BCI mediante una interfaz electrónica controlada por el usuario. Esta reorientación no constituye una simple solución de contingencia, sino una herramienta de aprendizaje y desarrollo por derecho propio: al eliminar la dependencia del hardware EEG, permite estudiar de forma segura y repetible la lógica de control, la comunicación serial y la respuesta del manipulador —exactamente los mismos componentes que debería enfrentar cualquier sistema BCI real—, sin asumir los riesgos ni los costos asociados a la instrumentación biomédica. Desde el punto de vista académico, una plataforma de este tipo resulta particularmente valiosa porque, al integrarse en MATLAB, Simulink y Simscape Multibody, permite al estudiante experimentar con los parámetros de control, observar el comportamiento dinámico del manipulador y validar el funcionamiento del conjunto de forma reproducible; es decir, repitiendo exactamente el mismo estímulo de entrada cuantas veces sea necesario antes de incorporar un equipo EEG real en investigaciones posteriores. A nivel institucional, la Facultad de Sistemas y Telecomunicaciones (FACSISTEL) de la UPSE no dispone de una herramienta que permita explorar de manera práctica la relación entre los sistemas BCI y la robótica asistida, lo cual limita el desarrollo de competencias en un área emergente de la ingeniería. En consecuencia, el desarrollo de esta plataforma responde a esa necesidad y, al mismo tiempo, sienta las bases para futuros proyectos orientados a la rehabilitación motora y a la asistencia de personas con movilidad reducida, ámbitos en los que los sistemas de interfaz cerebro-computadora poseen un gran potencial [9].

1.2. Planteamiento del problema

El control de dispositivos robóticos a partir de señales cerebrales representa uno de los retos más complejos de la neuroingeniería [2], ya que requiere integrar dentro de un mismo sistema la adquisición de señales biomédicas, su procesamiento y la generación de comandos. No obstante, la implementación de estos sistemas en el contexto universitario tropieza con una dificultad concreta: la disponibilidad del equipo de adquisición. Durante el desarrollo de este proyecto, el casco OpenBCI, previsto para captar las señales EEG reales, no llegó dentro del cronograma establecido, de manera que la propuesta original, basada en la lectura directa de la actividad cortical, no pudo ejecutarse. Ante esta limitación surge la pregunta de cómo continuar estudiando el control de un brazo robótico mediante una interfaz cerebro-computadora cuando no se cuenta con el hardware EEG. Si bien existen entornos de simulación robótica ampliamente difundidos, la mayoría de los trabajos asume que las señales neuronales se encuentran disponibles, por lo que resulta poco frecuente hallar plataformas accesibles que reproduzcan el comportamiento de las bandas cerebrales con fines de prueba y enseñanza. Esta carencia dificulta que un estudiante pueda comprender y validar la cadena completa de control antes de disponer de instrumentación especializada. Por consiguiente, el problema que aborda este trabajo consiste en la ausencia de una herramienta de bajo costo que permita emular las señales asociadas a los ritmos cerebrales y, a partir de ellas, controlar y validar el comportamiento de un brazo robótico virtual de cuatro grados de libertad. Para responder a esta situación se plantea el diseño de un módulo físico de emulación junto con su integración en un entorno de simulación, de modo que la interacción entre el sistema BCI y el manipulador pueda analizarse de forma segura, reproducible y económica.

1.3. Panorama actual

Durante la última década, el campo de las interfaces cerebro-computadora ha avanzado de la mano de tres factores: la aparición de equipos EEG de bajo costo, el desarrollo de algoritmos de clasificación cada vez más eficientes y la consolidación de plataformas de simulación capaces de reproducir con fidelidad la dinámica de los sistemas robóticos. Dispositivos como Emotiv EPOC o NeuroSky acercaron la tecnología EEG a un público más amplio, mientras que proyectos de código abierto como OpenBCI ofrecieron mayor flexibilidad para la investigación [11]; aun así, el acceso a este tipo de hardware sigue siendo una barrera para muchos laboratorios, sobre todo durante las primeras etapas de aprendizaje. Frente a esa realidad, las herramientas de simulación y emulación han ganado protagonismo como una vía económica

para estudiar los sistemas BCI antes de implementar el hardware definitivo: entornos como MATLAB, Simulink y Simscape Multibody permiten modelar tanto la cinemática como la dinámica de un manipulador y, al mismo tiempo, integrar señales generadas artificialmente que cumplen el papel de las señales cerebrales [10]. De esta manera es posible reproducir el flujo completo de información desde la señal de entrada hasta el movimiento del brazo, sin necesidad de un participante ni de un casco EEG. En esta línea, diversos trabajos han recurrido a señales analógicas simuladas, generadores de funciones o potenciómetros para representar la entrada de un sistema de control y así verificar su funcionamiento. Estas estrategias, aunque sencillas, resultan especialmente útiles en el ámbito educativo porque permiten visualizar de manera tangible cómo una variación en la señal de entrada se traduce en una acción concreta del actuador. La presente investigación se inscribe en ese enfoque, dado que propone un módulo físico que emula las bandas neuronales mediante potenciómetros y las vincula con el control de un brazo robótico virtual. Por otra parte, persisten retos que la comunidad científica continúa abordando, entre ellos la variabilidad de las señales reales entre distintos usuarios, la presencia de ruido y artefactos, y las latencias propias del procesamiento en tiempo real. Si bien estos aspectos solo se manifiestan plenamente al trabajar con EEG real, una plataforma de emulación permite anticiparlos y preparar el sistema para enfrentarlos en investigaciones futuras, lo que refuerza su utilidad como paso previo a la incorporación de hardware como el casco OpenBCI [12].

1.4. Objetivos

1.4.1. Objetivo general

Diseñar y desarrollar una plataforma de bajo costo integrada por un módulo físico de emulación de señales cerebrales y un modelo de brazo robótico virtual de cuatro grados de libertad, implementados en MATLAB, Simulink y Simscape Multibody, permitiendo validar la comunicación, el control óptimo y la respuesta del manipulador frente a comandos inspirados en el funcionamiento de las interfaces cerebro-computadora.

1.4.2. Objetivos específicos

1. Diseñar y construir un módulo físico de emulación de señales cerebrales, mediante una placa de circuito impreso propia, que represente por medio de potenciómetros la intensidad de las bandas neuronales delta, theta, alfa y beta.

2. Modelar un brazo robótico virtual antropomórfico de cuatro grados de libertad, corrigiendo los ángulos articulares e incorporando límites coherentes con el movimiento humano.
3. Implementar una arquitectura de control óptimo (LQR/LQG con filtro de Kalman) que traduzca las señales emuladas en el movimiento de las articulaciones del manipulador dentro de Simscape Multibody.
4. Validar la comunicación, el control y el comportamiento integral del sistema mediante pruebas experimentales sobre el módulo físico, el manipulador simulado y su interacción conjunta.

1.5. Fundamentos teóricos

El sistema desarrollado en este trabajo se sitúa en la intersección de tres campos que se complementan entre sí: por un lado, la neurociencia aporta los conceptos relacionados con las señales cerebrales y sus bandas; por otro, la ingeniería de control proporciona las estrategias para traducir esas señales en órdenes de movimiento y, finalmente, la robótica ofrece el modelo matemático del manipulador. A continuación, se presentan los conceptos teóricos que sustentan la propuesta.

1.5.1. Interfaz cerebro-computadora (BCI)

Una interfaz cerebro-computadora constituye un sistema capaz de establecer una vía de comunicación directa entre el cerebro humano y un dispositivo externo sin recurrir a las vías neuromusculares convencionales [1]. Su funcionamiento descansa en cuatro etapas: la adquisición de la señal, su preprocesamiento, la extracción de características y, por último, la clasificación que permite asociar la señal con una intención determinada. Cuando el dispositivo controlado es un manipulador robótico, a estas etapas se añade un lazo de control que convierte la intención reconocida en movimiento [2]. Las señales electroencefalográficas representan la modalidad más empleada en los sistemas BCI no invasivos, debido a su buena resolución temporal, su accesibilidad y su ausencia de riesgo para el usuario. Estas señales registran las pequeñas variaciones de potencial generadas por la actividad sincronizada de grupos de neuronas, con amplitudes que oscilan típicamente entre 1 y 100 microvoltios. Aunque el presente trabajo no adquiere señales reales, comprender este principio resulta indispensable, ya que el módulo de emulación busca reproducir de forma controlada el comportamiento de dichas señales.

1.5.2. Ritmos cerebrales y bandas neuronales

Los ritmos cerebrales corresponden a oscilaciones de la actividad eléctrica de la corteza que se agrupan en bandas de frecuencia, cada una asociada a determinados estados funcionales del cerebro. Si bien su comportamiento real es complejo y variable, su clasificación en bandas constituye la base sobre la que operan la mayoría de los sistemas BCI, motivo por el cual resulta necesario conocerlas para diseñar un módulo capaz de representarlas [3]. La banda delta, comprendida aproximadamente entre 0,5 y 4 Hz, es la de menor frecuencia y mayor amplitud, y se relaciona con el sueño profundo. La banda theta, situada entre 4 y 8 Hz, aparece durante la somnolencia, la relajación profunda y algunos procesos de memoria. Por su parte, la banda alfa, ubicada entre 8 y 13 Hz, se asocia con estados de relajación con los ojos cerrados y con la inhibición de áreas que no participan en una tarea, de manera que su variación resulta especialmente útil en los paradigmas de imaginación motora. Finalmente, la banda beta, que abarca de 13 a 30 Hz, se vincula con la concentración, la atención activa y la planificación de movimientos [4]. Dado que cada una de estas bandas refleja un estado distinto, los sistemas BCI suelen analizar cómo cambia su potencia para inferir la intención del usuario. En el presente trabajo esta idea se traslada al plano de la emulación, puesto que el módulo desarrollado permite ajustar de manera independiente la intensidad de cada banda y observar cómo dicha variación se traduce en un comando de movimiento, reproduciendo así la lógica que seguiría un sistema BCI real.

1.5.3. Emulación de señales y plataformas de simulación de bajo costo

La emulación de señales consiste en reproducir, mediante un dispositivo o un modelo, el comportamiento que tendría una señal real sin necesidad de adquirirla directamente de su fuente. En el ámbito de las interfaces cerebro-computadora, esta técnica resulta muy útil, ya que permite generar entradas con características controladas, como su amplitud o su banda dominante, y emplearlas para probar la cadena de procesamiento y de control. A diferencia de las señales reales, las señales emuladas son repetibles y se encuentran libres de artefactos, lo que facilita el análisis sistemático y reduce de forma considerable los costos. Conviene precisar en qué sentido resulta válido emplear un potenciómetro como sustituto de una señal EEG. En la práctica, ningún sistema BCI opera directamente sobre la forma de onda cruda del electroencefalograma; la etapa de extracción de características reduce cada banda de interés a un único valor escalar que varía lentamente en el tiempo,

habitualmente la potencia de la banda, es decir, la energía media contenida en la ventana de frecuencia correspondiente, obtenida mediante densidad espectral de potencia o un banco de filtros seguido de un promediado. Es precisamente ese valor, y no la oscilación original, el que se compara entre bandas o se entrega al clasificador [4, 5]. Dado que la potencia de banda es, por construcción, una magnitud escalar acotada y de variación lenta frente al ritmo oscilatorio subyacente, un voltaje analógico generado manualmente mediante un potenciómetro reproduce fielmente esa característica esencial: un nivel continuo entre un mínimo y un máximo, sin necesidad de sintetizar la oscilación de la que procede. La emulación propuesta en este trabajo se sitúa, por tanto, en la etapa posterior al preprocesamiento de la señal dentro del esquema clásico de un sistema BCI [2]: no pretende imitar la fisiología del electroencefalograma, sino reproducir, como aproximación de primer orden, la variable de control que el resto del sistema utilizaría en cualquier caso, lo cual resulta suficiente para validar la comunicación, el mapeo banda-articulación y la arquitectura de control antes de introducir hardware EEG real. Las plataformas de simulación complementan este enfoque al ofrecer un entorno en el que el manipulador y su control pueden evaluarse de manera virtual. Herramientas como Simscape Multibody reproducen la física del brazo robótico con un nivel de detalle suficiente para anticipar su comportamiento real, mientras que MATLAB y Simulink permiten implementar la lógica de control y la comunicación con el módulo de emulación. De la combinación de ambos elementos surge una plataforma de bajo costo en la que es posible estudiar, depurar y validar el sistema completo antes de incorporar instrumentación biomédica.

1.5.4. Cinemática y dinámica de manipuladores robóticos

El modelado de un brazo robótico se sustenta en dos componentes que se complementan: por un lado, la cinemática describe la geometría del movimiento sin considerar las fuerzas que lo originan; por otro, la dinámica relaciona los torques aplicados en las articulaciones con el movimiento resultante [6]. Para un manipulador de cuatro grados de libertad, la cinemática directa permite determinar la posición y la orientación del efector final a partir de los ángulos de cada articulación, para lo cual se emplea habitualmente la convención de Denavit–Hartenberg, que asigna un sistema de referencia a cada eslabón y describe la relación entre ellos mediante un conjunto reducido de parámetros [7]. La dinámica, por su parte, suele expresarse a través de la formulación de Euler–Lagrange, que el propio entorno Simscape Multibody resuelve de manera automática durante la simulación.

1.5.5. Estrategias de control en entornos de simulación

Una vez que el manipulador recibe un comando de movimiento, se necesita un sistema de control que asegure que cada articulación alcance el ángulo deseado con precisión y estabilidad. Para ello, el control se ejecuta en lazo cerrado, de modo que la posición real de cada junta se compara de forma continua con la referencia y la diferencia entre ambas se emplea para corregir la acción aplicada sobre la articulación [8]. Entre las estrategias disponibles para cerrar este lazo, la familia de controladores clásicos de tipo PID continúa siendo la más extendida en robótica de manipuladores por su sencillez de sintonía [6, 7]; sin embargo, su ajuste resulta esencialmente empírico y no ofrece ninguna garantía formal sobre el compromiso entre la velocidad de respuesta y el esfuerzo de control, aspecto que se vuelve relevante cuando, como en este trabajo, la señal de entrada proviene de un conversor analógico-digital de bajo costo y arrastra ruido eléctrico no despreciable. Frente a esta limitación, el regulador lineal-cuadrático (LQR) plantea el problema de control como la minimización de un índice de coste cuadrático que pondera de forma explícita tanto el error de seguimiento como el esfuerzo aplicado, cuya solución óptima se obtiene de forma sistemática resolviendo la ecuación algebraica de Riccati asociada al modelo en espacio de estados de la planta [30]; esta formulación sustituye el ajuste empírico de las ganancias por una elección razonada de las matrices de ponderación del error y del esfuerzo, lo que resulta especialmente conveniente al tener que sintonizar de manera consistente cuatro articulaciones con dinámicas distintas. La conveniencia de esta elección frente al PID cuenta con precedentes directos en la literatura: Batista et al. [27] compararon ambos controladores sobre la articulación de un manipulador robótico y reportaron resultados favorables al LQR en términos de error de posición y esfuerzo de control, mientras que Asghar et al. [28] evaluaron el LQR frente a un controlador robusto H-infinito en un manipulador de cuatro grados de libertad modelado en Simulink, un escenario prácticamente equivalente al de esta investigación; esto constituye un precedente directo para justificar la elección de un controlador óptimo de tipo LQR frente a alternativas de mayor complejidad de diseño y ajuste, como el control robusto H-infinito. El LQR clásico asume que el vector de estado completo está disponible para realimentación, condición que rara vez se cumple cuando la única medición disponible es una lectura ruidosa del conversor analógico-digital. Por ello, se incorpora un estimador lineal-cuadrático (LQE), que corresponde a la aplicación del filtro de Kalman [25] al problema de estimación óptima del estado a partir de mediciones ruidosas: el filtro pondera de forma recursiva la predicción que ofrece el modelo interno del sistema frente a la información que aporta cada nueva medición, en función

de las covarianzas de proceso y de medición que se le asignen, y entrega en cada instante la estimación de mínima varianza del estado real. La combinación resultante de un LQR alimentado por un estimador de Kalman se conoce como regulador lineal-cuadrático gaussiano (LQG) [30], arquitectura cuya robustez frente al ruido de medición en articulaciones flexibles ha sido reportada por Doina [29]. Se adoptó esta arquitectura combinada, en lugar de un LQR con realimentación directa del valor crudo del ADC, precisamente porque permite separar el problema de filtrado del ruido eléctrico del problema de control óptimo, de modo que cada uno puede ajustarse de forma independiente sin comprometer la estabilidad del lazo. En este trabajo dicho lazo se resolvió, por tanto, mediante una arquitectura de control óptimo basada en el regulador lineal-cuadrático (LQR), su versión gaussiana (LQG) y el estimador lineal-cuadrático (LQE), apoyada en un filtro de Kalman que depura la señal antes de accionar cada junta. Esta estrategia calcula la acción de control minimizando de forma explícita un índice de coste que pondera a la vez el error de seguimiento y el esfuerzo aplicado, lo que ofrece un compromiso equilibrado entre la precisión y la suavidad del movimiento y aporta robustez frente al ruido que arrastran las señales emuladas. De esta manera, cada articulación se corrige de forma continua hasta lograr el movimiento esperado.

1.5.6. Simscape Multibody como entorno de modelado dinámico

Simscape Multibody es un entorno de modelado físico integrado en Simulink que permite representar sistemas mecánicos como un conjunto de cuerpos rígidos interconectados por articulaciones, en lugar de exigir al usuario la derivación manual de las ecuaciones de movimiento [10]. Cada cuerpo se define mediante su geometría, su masa y su tensor de inercia, con parámetros que pueden importarse directamente desde un modelo CAD conservando las propiedades físicas reales de cada eslabón, mientras que cada articulación se representa con un bloque primitivo (Revolute Joint en el caso de las juntas rotacionales de este manipulador) que fija los grados de libertad permitidos entre los cuerpos que conecta y admite sensores de posición o velocidad, así como actuadores de par o de movimiento prescrito. A partir de esta descripción topológica, el motor de Simscape Multibody ensambla automáticamente las ecuaciones de movimiento del sistema completo mediante formulaciones multicuerpo equivalentes a las de Newton–Euler o Euler–Lagrange [6], y las integra numéricamente durante la simulación, de modo que tanto la cinemática (posición y orientación de cada eslabón en función de los ángulos articulares) como la dinámica (los pares que resultan de las inercias, la gravedad y los pares aplicados) se obtienen sin necesidad de programar explícitamente.

te las matrices de transformación homogénea ni las ecuaciones de Lagrange, aunque estas últimas, desarrolladas conceptualmente mediante la convención de Denavit–Hartenberg [7], siguen siendo el marco teórico que explica el comportamiento que el solver reproduce internamente. Para este trabajo, dicha automatización resultó particularmente conveniente porque permitió concentrar el esfuerzo de modelado en la corrección de los límites articulares y en el diseño del lazo de control, delegando en el entorno la resolución numérica de la dinámica del manipulador de cuatro grados de libertad.

1.6. Software utilizado

El desarrollo de la plataforma se apoyó en un conjunto de herramientas que se integran entre sí. MATLAB constituyó el entorno central, puesto que en él se implementó la lectura de los datos provenientes del módulo de emulación a través del puerto serial, así como el escalado de dichos valores hacia los ángulos de cada articulación. Por otro lado, Simulink permitió construir el sistema de control de forma visual mediante diagramas de bloques, lo que facilitó la conexión entre la señal emulada y el modelo del manipulador. Simscape Multibody se utilizó para representar el brazo robótico como un conjunto de cuerpos rígidos unidos por articulaciones rotacionales, con propiedades de masa, inercia y geometría que pueden importarse desde el modelo CAD [10]; de esta manera, el entorno resolvió automáticamente las ecuaciones de movimiento y reprodujo la respuesta dinámica del manipulador. Para el diseño tridimensional del brazo se empleó un software CAD, mientras que la programación del microcontrolador del módulo de emulación se realizó en el entorno Arduino IDE.

1.7. Bloques de MATLAB/Simulink

La implementación en Simulink hizo uso de un conjunto de bloques con funciones definidas dentro del flujo de control. En primer lugar, el bloque de recepción serial se encargó de leer las tramas enviadas por el módulo de emulación, las cuales contienen la banda activa, el valor del potenciómetro y el comando seleccionado. A continuación, los bloques de ganancia y saturación escalaron dichos valores hacia el rango angular válido de cada articulación, de modo que se respetaran los límites establecidos en el modelado y se evitaran movimientos fuera de lo permitido. Posteriormente, el bloque del controlador óptimo (LQR/LQG con filtro de Kalman) cerró el lazo de posición de cada junta, comparando el ángulo de referencia con el ángulo real entregado por el sensor de la articulación. Finalmente, los bloques Revolute Joint y Rigid Transform de Simscape Multibody [10] modelaron las cuatro articulaciones

rotacionales y las transformaciones geométricas asociadas a los parámetros de Denavit–Hartenberg, mientras que los bloques de visualización permitieron observar el comportamiento del manipulador y registrar las variables de interés durante la simulación.

2. Métodos y diseño experimental

2.1. Tipo de investigación

La presente investigación se enmarca dentro de un enfoque que combina cuatro tipos complementarios: la investigación aplicada, descriptiva, exploratoria y basada en simulación [32]. Esta combinación responde a la naturaleza del proyecto, dado que se busca resolver un problema concreto mediante el diseño de una plataforma, describir con detalle cada uno de sus componentes, explorar distintas alternativas de implementación y, finalmente, validar su comportamiento a través de la simulación. Cabe recalcar que en ningún momento se contempló la experimentación con sujetos, la adquisición de señales EEG reales ni la validación clínica del sistema, de manera que todo el trabajo se desarrolla en un entorno controlado.

2.2. Investigación aplicada

El carácter aplicado de la investigación [32] se manifiesta en que el conocimiento teórico sobre la interfaz cerebro-computadora, las bandas neuronales y el control robótico se utilizó para construir una solución concreta y funcional, en lugar de limitarse al plano conceptual. El trabajo dio lugar a un módulo físico de emulación y a un modelo simulado del manipulador, los cuales pueden emplearse de inmediato como herramientas de estudio. Por consiguiente, la investigación no solo genera conocimiento, sino que también produce un resultado tangible orientado a una necesidad real del laboratorio.

2.3. Investigación descriptiva

La componente descriptiva [32] se orientó a caracterizar con precisión todos los elementos del sistema, sin manipular variables ni establecer relaciones de causa y efecto. En este sentido, se documentaron las especificaciones de los componentes electrónicos del módulo de emulación, la función de cada potenciómetro y de cada pulsador, las características del modelo del brazo robótico y la arquitectura de comunicación entre el hardware y MATLAB. De esta forma se obtuvo una descripción completa del funcionamiento de la

plataforma, que sirve además como base para el manual de usuario incluido más adelante.

2.4. Investigación exploratoria

La metodología exploratoria [32] se aplicó durante las fases iniciales de diseño, momento en el que se evaluaron distintas alternativas antes de definir la configuración final. Por ejemplo, se analizó la conveniencia de emplear potenciómetros o pulsadores para representar la entrada del sistema, así como la mejor manera de mostrar la información en la pantalla SPI y de estructurar las tramas de comunicación serial. De igual modo, se exploraron diferentes valores para las ganancias del controlador y distintas formas de asociar cada banda con una articulación, todo ello mediante un ciclo de prueba, ajuste y evaluación que permitió afinar la propuesta de manera progresiva.

2.5. Investigación basada en simulación

El componente basado en simulación constituye el eje central del proyecto, ya que el comportamiento del manipulador se estudió íntegramente dentro del entorno MATLAB, Simulink y Simscape Multibody. La simulación permitió reproducir la cinemática del brazo robótico, observar su respuesta frente a los comandos emulados y verificar el cumplimiento de los límites articulares, todo ello sin riesgo y sin necesidad de construir físicamente el manipulador. Gracias a este enfoque, resultó posible repetir las pruebas cuantas veces fuera necesario y ajustar los parámetros del sistema hasta obtener un comportamiento adecuado.

2.6. Descripción general del sistema propuesto

El sistema propuesto se organiza en tres bloques que trabajan de forma encadenada. El primero corresponde al módulo físico de emulación, en el cual el usuario ajusta los potenciómetros que representan las bandas delta, theta, alfa y beta, selecciona un comando mediante el pulsador y observa la información en la pantalla SPI. El segundo bloque es la comunicación serial, encargada de transmitir hacia MATLAB la banda activa, el valor de la señal y el comando elegido. El tercer bloque es el entorno de simulación, donde MATLAB y Simulink interpretan la información recibida, la convierten en referencias angulares y, a través del controlador óptimo (LQR/LQG con filtro de Kalman), accionan el modelo del brazo robótico en Simscape Multibody. De esta manera, la señal emulada recorre todo el sistema hasta producir un movimiento concreto del manipulador, lo que reproduce de forma controlada

el flujo que tendría un sistema BCI real. La **Figura 1** representa este flujo general desde el ajuste de las bandas en el módulo físico hasta el movimiento del brazo en el entorno de simulación.

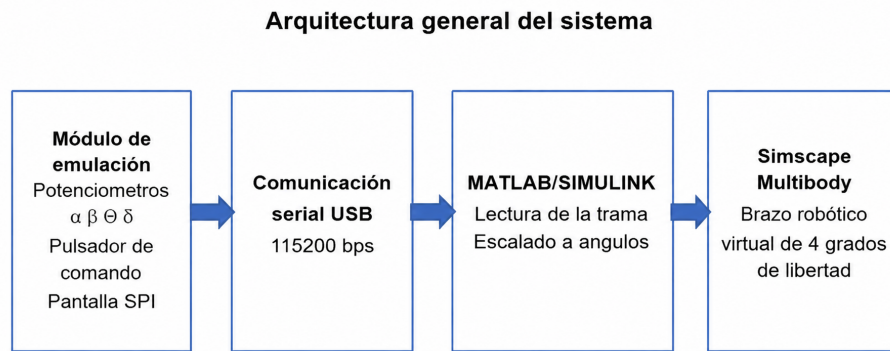


Figura 1

Diagrama general del sistema propuesto, desde el módulo de emulación hasta el brazo robótico simulado (Autoría propia, 2026).

2.7. Descripción general del módulo de emulación

El módulo de emulación constituye el corazón de la propuesta, puesto que reemplaza la función que cumpliría el casco OpenBCI dentro de un sistema BCI real. En lugar de captar la actividad cerebral, el módulo permite que el usuario reproduzca de forma manual la intensidad de cada banda neuronal, de modo que se obtiene una señal de entrada con características conocidas y repetibles. El dispositivo no se montó sobre una protoboard, sino que se concibió como una pequeña caja terminada con una placa de circuito impreso propia y una carcasa impresa en 3D, lo cual le otorga un aspecto profesional y facilita su manejo. En la parte frontal de la caja se ubicaron cuatro potenciómetros, cada uno rotulado con la banda que representa, junto con un pulsador que permite confirmar el comando y una pantalla SPI que muestra en todo momento la banda activa, el comando enviado y el estado del sistema. De esta forma, el usuario cuenta con una interfaz clara e intuitiva que hace visible en la parte física el proceso que ocurre dentro del sistema.

Tabla 1*Justificación cinemática de la nomenclatura griega*

Símbolo	Frecuencia (Hz)	Articulación	Justificación en cinemática
δ (Delta)	0,5 – 4	Cintura	Variable de articulación prismática o rotacional base. En DH, δ representa el desplazamiento o rotación inicial que fija el sistema de coordenadas global.
θ (Theta)	4 – 8	Hombro	Ángulo de articulación rotacional estándar. En DH, θ es la variable principal para articulaciones giratorias, ideal para el hombro, que proporciona elevación.
α (Alfa)	8 – 13	Codo	Ángulo de torsión entre ejes Z. En DH, α mide la desviación entre ejes consecutivos, aplicable al codo, donde cambia la dirección del antebrazo.
β (Beta)	13 – 30	Muñeca	Ángulo de orientación del efector final. Aunque no es estándar en DH, β se usa comúnmente en notación de Euler para describir la rotación final de la muñeca.

2.8. Potenciómetros como representación de las bandas neuronales

El elemento más representativo del módulo son los cuatro potenciómetros, ya que cada uno se asocia a una banda neuronal y permite ajustar su intensidad de manera independiente al girar la perilla. El potenciómetro entrega un voltaje variable entre 0 y 5 V que el microcontrolador convierte en

un valor digital mediante su conversor analógico-digital; ese valor representa, de forma análoga, la potencia que tendría la banda correspondiente en una señal real, dado que un giro mayor equivale a una banda más activa [4]. La asociación entre cada banda y una articulación del brazo se definió buscando una relación clara y fácil de recordar para el usuario; conviene destacar que esta asignación es configurable, por lo que puede modificarse según los requerimientos de futuras prácticas. La **Tabla 2** resume la correspondencia adoptada en este trabajo.

Tabla 2

Asignación de cada potenciómetro a una banda neuronal y a una articulación del manipulador.

Potenciómetro neuronal	Banda	Rango de grados en radianes	Frecuencia de muestreo (Hz)	Articulación asociada
P1	Delta (δ)	-0,10 a 4,09	100	Cintura
P2	Theta (θ)	-3,40 a 0,30	100	Hombro
P3	Alfa (α)	5,47 a 0,31	100	Codo
P4	Beta (β)	-0,10 a 3,37	100	Muñeca

La delimitación de las bandas por frecuencia sigue la clasificación clásica de los ritmos corticales descrita por Pfurtscheller y Neuper [3] y sistematizada por Sanei y Chambers [4], de modo que las bandas delta, theta, alfa y beta se corresponden con los rangos de frecuencia aceptados en la literatura de electroencefalografía. La asignación de cada banda a una articulación es una decisión de diseño propia y configurable, orientada a facilitar el aprendizaje del sistema.

2.8.1. Pulsador de confirmación de comando

Además de los potenciómetros, el módulo incorpora un pulsador cuya función es complementar el control analógico con una acción de confirmación: mientras que los potenciómetros definen la intensidad de cada banda y, con ello, el ángulo de destino de la articulación asociada, el pulsador permite confirmar el envío del comando hacia MATLAB. De este modo se ofrece una interfaz sencilla en la que el usuario ajusta la banda deseada y, mediante el pulsador, indica el momento en que la acción debe transmitirse al manipulador. De este modo, el sistema reproduce, dentro de su simplicidad, la lógica de un BCI en la que la detección de una banda dominante se traduce en una orden concreta. Por ejemplo, al ajustar el potenciómetro de la banda

alfa y presionar el pulsador de confirmación, el módulo envía el comando correspondiente a la flexión del codo, el cual se ejecuta posteriormente en la simulación.

2.8.2. Pantalla SPI: visualización de banda, comando y estado

Con el fin de que el módulo resulte más llamativo e informativo, se incorporó una pantalla SPI que despliega en tiempo real la información más relevante del sistema. La pantalla muestra tres datos principales: la banda que se encuentra activa en ese momento, el comando que se está enviando y el estado general del sistema. Por esto, el usuario no necesita recurrir a la computadora para saber qué ocurre dentro del módulo, ya que toda la información se encuentra disponible en la propia caja. Por ejemplo, cuando se ajusta el potenciómetro de la banda alfa, la pantalla puede indicar “Banda: ALFA”, “Comando: Codo” y “Estado: Enviando”, de manera que el proceso queda claramente representado. Esta retroalimentación visual no solo facilita el uso del dispositivo, sino que también lo convierte en una herramienta didáctica, puesto que permite observar de forma directa la relación entre la señal emulada y la acción enviada al manipulador.

2.8.3. Microcontrolador y acondicionamiento de señal

El microcontrolador es el encargado de coordinar el funcionamiento de todo el módulo. Por una parte, lee de forma continua el valor de cada potenciómetro a través de sus entradas analógicas y detecta la pulsación del botón mediante sus entradas digitales; por otra parte, gestiona la información que se muestra en la pantalla SPI y compone las tramas que se transmiten hacia MATLAB. Para garantizar una lectura estable, las señales de los potenciómetros se acondicionan mediante elementos de filtrado que atenúan el ruido y evitan variaciones bruscas en los valores registrados. Dado que el conversor analógico-digital del microcontrolador trabaja habitualmente con una resolución de 10 bits, cada lectura se representa con un valor entre 0 y 1023, el cual se asocia luego a la intensidad de la banda correspondiente. Este valor se normaliza dentro del firmware para que pueda interpretarse de manera coherente en MATLAB, donde finalmente se traduce en una referencia angular para la articulación asignada.

2.9. Acondicionamiento de la señal

El acondicionamiento de la señal comprende cuatro operaciones sucesivas: la generación de la tensión en el potenciómetro, su filtrado analógico, la con-

versión analógico-digital y la normalización dentro del firmware. **Divisor de tensión del potenciómetro:** Cada potenciómetro se conecta como divisor de tensión de tres terminales. Si $x \in [0, 1]$ es la posición fraccional del cursor, la tensión entregada al ADC es

$$V_p = V_{cc} \frac{R_{cursor}}{R_{total}} = V_{cc}x, \quad V_{cc} = 5 \text{ V}, \quad (1)$$

de modo que el giro de la perilla barre linealmente la tensión entre 0 y 5 V. **Filtro pasa-bajos de primer orden:** La red resistiva de acondicionamiento, formada por resistencias de 4,7 k Ω y 10 k Ω , conforma junto con la capacidad de entrada un filtro pasa-bajos que atenúa el ruido eléctrico. Su función de transferencia y su frecuencia de corte son:

$$H(s) = \frac{1}{1 + sRC}, \quad f_c = \frac{1}{2\pi RC}. \quad (2)$$

En su implementación discreta a periodo T_e , este filtro equivale a un promediador exponencial:

$$y[k] = \alpha x[k] + (1 - \alpha)y[k - 1], \quad \alpha = \frac{T_e}{RC + T_e} = \frac{2\pi f_c T_e}{1 + 2\pi f_c T_e}, \quad (3)$$

expresión que se retoma en el filtro 1 € (*One Euro*). **Conversión analógico-digital (10 bits):** El ADC del microcontrolador cuantifica V_p en N niveles con resolución

$$N = \left\lfloor \frac{V_p}{V_{ref}}(2^n - 1) \right\rfloor, \quad q = \frac{V_{ref}}{2^n - 1} = \frac{5 \text{ V}}{1023} \approx 4,88 \text{ mV/LSB}, \quad (4)$$

con $n = 10$ y $V_{ref} = 5 \text{ V}$, de forma que $N \in \{0, 1, \dots, 1023\}$ [4]. **Normalización:** Finalmente, el valor digital se normaliza en el firmware al intervalo unitario para su interpretación uniforme en MATLAB:

$$v_{norm} = \frac{N}{1023} \in [0, 1]. \quad (5)$$

Este v_{norm} representa la intensidad (potencia de banda emulada, (1)) que posteriormente se mapea al ángulo de la articulación asignada.

2.10. Comunicación serial con MATLAB

La comunicación entre el módulo y la computadora se realiza a través del puerto serial, aprovechando la conexión USB del microcontrolador. El firmware compone una trama binaria que contiene los cuatro valores de los

potenciómetros, uno por cada articulación, y la envía de forma periódica por el puerto serial COM5. En el extremo de la computadora, el modelo de Simulink recibe estos bytes mediante el bloque Serial Receive (puerto COM5, tipo uint16, tamaño de 4 articulaciones, cabecera 170 y 85, equivalentes a 0xAA y 0x55, formato “Column major” y tiempo de muestreo de 0,1 s), y el bloque Byte Unpack los reagrupa en cuatro enteros de 16 bits (uint16), uno por articulación, para utilizarlos dentro del lazo de control. La trama se organizó de forma compacta: los datos se agrupan en cuatro enteros de 16 bits (uint16), uno por cada articulación, con los valores del conversor analógico-digital comprendidos entre 0 y 1023. Un ejemplo de la distribución de los bytes es el siguiente, en el que cada par de bytes codifica el ángulo de una articulación, según el orden angle1, angle2, angle3 y angle4. Byte 0-1 → angle1 (cintura) | Byte 2-3 → angle2 (hombro) Byte 4-5 → angle3 (codo) | Byte 6-7 → angle4 (muñeca) Gracias a este formato, MATLAB puede separar cada campo y asignar el valor recibido a la articulación correspondiente, lo que asegura una interpretación clara y ordenada de la información proveniente del módulo de emulación.

3. Desarrollo experimental

El desarrollo del proyecto se llevó a cabo siguiendo una secuencia de etapas que abarcan desde el diseño de los componentes hasta la validación del comportamiento del manipulador. A continuación, se describe cada una de ellas, de modo que se evidencie el proceso completo que permitió integrar la plataforma.

3.1. Modelado del robot virtual

3.1.1. Configuración antropomórfica de cuatro grados de libertad

El manipulador se modeló siguiendo una configuración antropomórfica de cuatro grados de libertad, con el fin de que cada articulación rotacional reproduzca un movimiento característico del brazo humano. La primera articulación corresponde a la cintura, que permite el giro de la base sobre un eje vertical; la segunda representa el hombro, donde se realiza la flexión y la extensión; la tercera equivale al codo y, asimismo, ejecuta la flexión del antebrazo; por último, la cuarta corresponde a la muñeca, que aporta un movimiento de giro o flexión final [6]. Esta disposición ofrece un buen compromiso entre simplicidad y capacidad de movimiento, lo que la hace adecuada para una plataforma de estudio.

3.1.2. Diseño CAD del manipulador

El brazo robótico se diseñó en un entorno CAD, donde se definió la geometría de cada eslabón y se ensamblaron las cuatro articulaciones respetando la configuración antropomórfica. Durante el modelado se cuidó que las proporciones de los eslabones resultaran coherentes con un brazo real, de modo que el comportamiento del manipulador en la simulación se asemejara al de un sistema físico. Una vez validado el ensamblaje, el modelo se preparó para su exportación hacia el entorno de simulación, conservando las propiedades de masa, inercia y geometría necesarias para reproducir su dinámica [10].

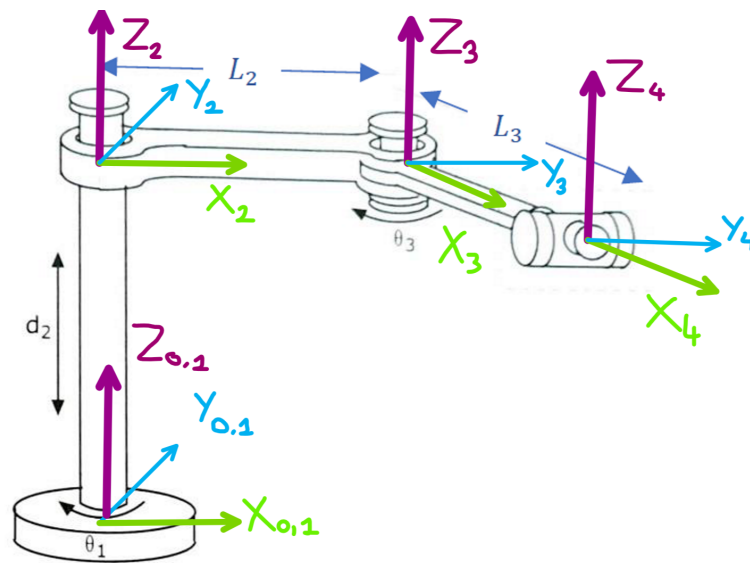


Figura 2
Cinemática y dinámica de manipuladores robóticos (Autoría propia, 2025).

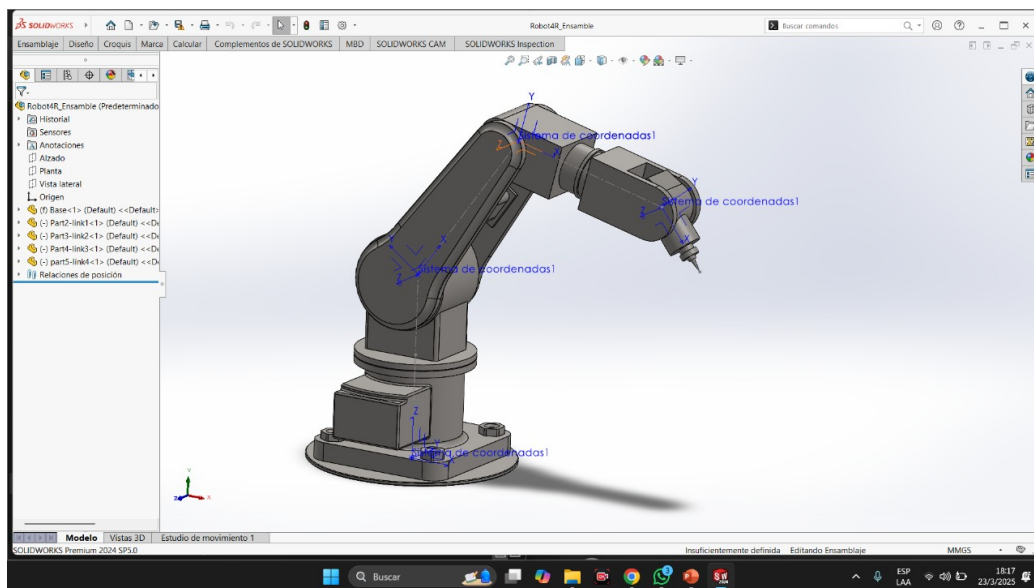


Figura 3
Modelo virtual del manipulador en el software CAD (Ing. Edison Coral, 2025).

3.1.3. Parámetros de Denavit–Hartenberg

Para describir la cinemática del manipulador se emplearon los parámetros de Denavit–Hartenberg, los cuales relacionan los sistemas de referencia de cada eslabón mediante cuatro magnitudes [7]. Las dimensiones físicas del manipulador que sustentan estos parámetros, tomadas del informe de referencia de Coral Salinas [31] para este mismo manipulador de cuatro grados de libertad, se muestran en la Figura 4.

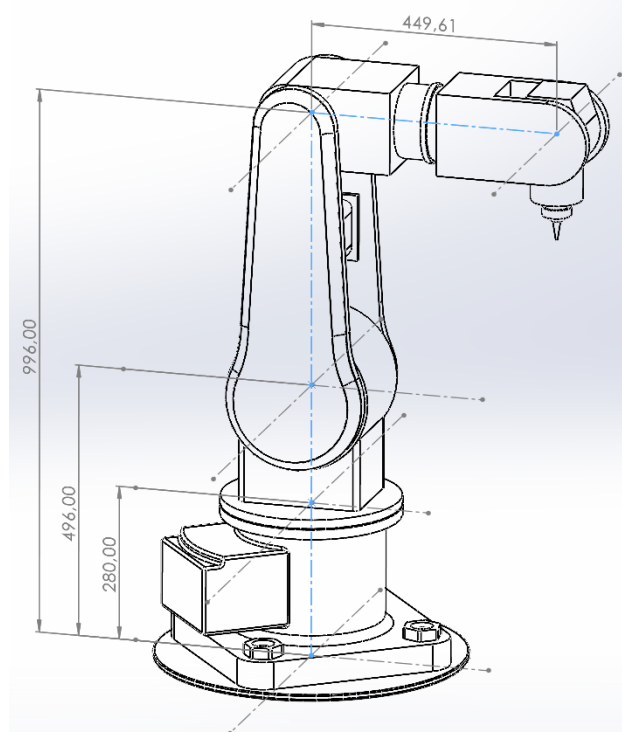


Figura 4

Dimensiones del manipulador de cuatro grados de libertad (Coral Salinas, 2025 [31]).

3.1.4. Cinemática directa mediante la convención de Denavit–Hartenberg

La convención de Denavit–Hartenberg (DH) asigna a cada eslabón un sistema de referencia y describe la relación entre dos sistemas consecutivos mediante cuatro parámetros: el ángulo de articulación θ_i , el desplazamiento d_i a lo largo del eje z_{i-1} , la longitud a_i del eslabón a lo largo de x_i y el ángulo de torsión α_i entre z_{i-1} y z_i [6, 7]. La transformación homogénea entre el

sistema $i - 1$ y el sistema i se obtiene como:

$${}^{i-1}T = \text{Rot}_z(\theta_i) \text{Trans}_z(d_i) \text{Trans}_x(a_i) \text{Rot}_x(\alpha_i)$$

$$= \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

$$T = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 = \begin{bmatrix} R_{3 \times 3} & P \\ 0 & 1 \end{bmatrix}, \quad (7)$$

donde R es la matriz de orientación del efector final y $P = [p_x, p_y, p_z]^T$ su posición.

3.1.5. Parámetros DH obtenidos del modelo CAD

Los parámetros DH obtenidos del modelo CAD se resumen en la Tabla 3. Se observa que $\alpha_1 = +90^\circ$ (por eso 0T_1 contiene $\sin \alpha_1 = 1$, $\cos \alpha_1 = 0$): el

Tabla 3

Parámetros DH del brazo robótico

Articulación	θ_i (rad)	d_i (mm)	a_i (mm)	α_i (grados)	Rango
1 (Cintura)	q_1	496	0	90	-180° a 180°
2 (Hombro)	q_2	0	500	0	-90° a 90°
3 (Codo)	q_3	0	449,61	0	-135° a 135°
4 (Muñeca)	q_4	0	195,43	0	-180° a 180°

eje de la base es vertical mientras que el del hombro es horizontal, formando 90° . En las tres articulaciones siguientes los ejes permanecen paralelos ($\alpha_i = 0$). Estos parámetros se emplean para verificar la coherencia geométrica del modelo en Simscape Multibody y para calcular analíticamente la posición del efector final mediante

$${}^0T_4 = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 = \begin{bmatrix} \mathbf{R}_{3 \times 3} & \mathbf{p}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

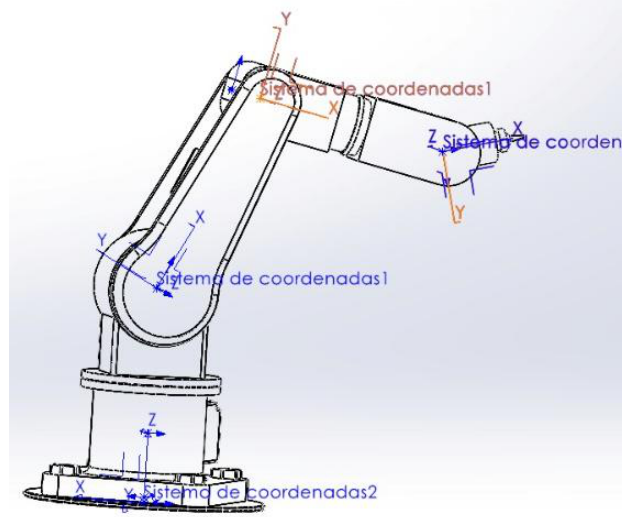
En la **Tabla 4** se presentan dichos parámetros, donde θ representa el ángulo de la articulación, d el desplazamiento a lo largo del eje, a la longitud del eslabón y α el ángulo de torsión entre ejes consecutivos.

Tabla 4

Parámetros Denavit–Hartenberg del manipulador de cuatro grados de libertad.

Articulación				
i	θ_i	d_i (mm)	a_i (mm)	α_i
1	q_1	496	0	$+90^\circ$
2	q_2	0	500	0°
3	q_3	0	449,61	0°
4	q_4	0	195,43	0°

Los parámetros de Denavit–Hartenberg de la Tabla 4 se construyeron siguiendo la convención estándar formalizada por Craig [7] y por Spong et al. [6], que asigna un sistema de referencia a cada eslabón mediante cuatro magnitudes. Los rangos de operación que acompañan a cada articulación corresponden a la calibración real del sistema y se detallan en la Tabla 4. La Figura 5 —también tomada del informe de referencia [31]— ilustra la asignación de los sistemas de coordenadas (X_i, Y_i, Z_i) a cada articulación del manipulador, siguiendo esta convención.

**Figura 5**

Sistemas de coordenadas asignados a las articulaciones del manipulador (Coral Salinas, 2025 [31]).

A partir de estos parámetros se construye, para cada eslabón, la matriz de transformación homogénea general de Denavit–Hartenberg, la cual relaciona

el sistema de referencia del eslabón i con el del eslabón $i - 1$ mediante una rotación y una traslación compuestas [6, 7]:

$${}^{i-1}A_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

Siguiendo el procedimiento desarrollado por Coral Salinas [31] para este mismo manipulador de cuatro grados de libertad, al sustituir en la Ecuación (9) los valores de d_i , a_i y α_i de la Tabla 4 se obtienen las cuatro matrices individuales, cada una en función únicamente del ángulo articular correspondiente:

3.1.6. Matrices de transformación homogénea

A continuación, se presentan las matrices de transformación homogénea para cada articulación, obtenidas al sustituir los parámetros DH de la Tabla 3 en la Ecuación (6).

$$A_0^1(\theta_1) = \begin{bmatrix} \cos \theta_1 & 0 & \sin \theta_1 & 0 \\ \sin \theta_1 & 0 & -\cos \theta_1 & 0 \\ 0 & 1 & 0 & 496 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

Para la articulación 1 (cintura), se tiene $d_1 = 496$ mm, $a_1 = 0$ y $\alpha_1 = 90^\circ$. Al sustituir $\sin \alpha_1 = 1$ y $\cos \alpha_1 = 0$ en la matriz DH general, se obtiene la matriz anterior, que representa una rotación alrededor del eje z y una traslación vertical de 496 mm.

$$A_1^2(\theta_2) = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & 500 \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & 500 \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

Para la articulación 2 (hombro), se tiene $d_2 = 0$, $a_2 = 500$ mm y $\alpha_2 = 0^\circ$. Al sustituir $\sin \alpha_2 = 0$ y $\cos \alpha_2 = 1$, la matriz se reduce a una rotación en z y una traslación horizontal de 500 mm a lo largo de x .

$$A_2^3(\theta_3) = \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & 449,61 \cos \theta_3 \\ \sin \theta_3 & \cos \theta_3 & 0 & 449,61 \sin \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$

Para la articulación 3 (codo), se tiene $d_3 = 0$, $a_3 = 449,61$ mm y $\alpha_3 = 0^\circ$. Nuevamente, al ser $\alpha_3 = 0$, la matriz corresponde a una rotación en z con traslación horizontal de 449,61 mm.

$$A_3^4(\theta_4) = \begin{bmatrix} \cos \theta_4 & -\sin \theta_4 & 0 & 195,43 \cos \theta_4 \\ \sin \theta_4 & \cos \theta_4 & 0 & 195,43 \sin \theta_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

Para la articulación 4 (muñeca), se tiene $d_4 = 0$, $a_4 = 195,43$ mm y $\alpha_4 = 0^\circ$. Al igual que las anteriores, la matriz describe una rotación en z y una traslación horizontal de 195,43 mm. La transformación total que ubica el efector final respecto a la base se obtiene multiplicando en cadena las cuatro matrices anteriores [31]:

$${}^0A_4 = A_0^1(\theta_1) A_1^2(\theta_2) A_2^3(\theta_3) A_3^4(\theta_4) \quad (14)$$

cuya submatriz de rotación 3×3 describe la orientación del efector final y cuya última columna entrega su posición (p_x, p_y, p_z) en función de los cuatro ángulos articulares. Dado que el entorno Simscape Multibody resuelve internamente estas operaciones a través de los bloques Revolute Joint y Rigid Transform, las ecuaciones anteriores se utilizan en este trabajo principalmente para verificar la coherencia geométrica del modelo y como base teórica del comportamiento que reproduce la simulación.

3.2. Cinemática inversa del manipulador

Como complemento al modelo de cinemática directa, Coral Salinas [31] desarrolló también la formulación de cinemática inversa para este mismo manipulador de cuatro grados de libertad, la cual permite obtener los ángulos articulares q_1, q_2, q_3, q_4 a partir de una posición cartesiana deseada del efector final (X_0, Y_0, Z_0) y de un ángulo de orientación de muñeca (*pitch*) q definido como parámetro de diseño. Aunque el sistema desarrollado en este trabajo comanda directamente los ángulos articulares a partir de las bandas emuladas y no requiere resolver esta cinemática inversa en tiempo real, las ecuaciones se incluyen aquí porque constituyen el fundamento analítico del modelo geométrico del brazo y sientan la base para una eventual extensión futura de la plataforma hacia un control por coordenadas cartesianas. La posición inicial o de referencia (*home*) del manipulador es aquella a partir de la cual se trazan los ejes de referencia para determinar los ángulos; el procedimiento de obtención de las ecuaciones depende de la configuración del robot.

La Figura 6 —tomada del informe de referencia [31]— muestra el esquema geométrico empleado para la resolución sistemática, con las distancias M y h y los ángulos auxiliares que se emplean en las Ecuaciones (15) a (19).

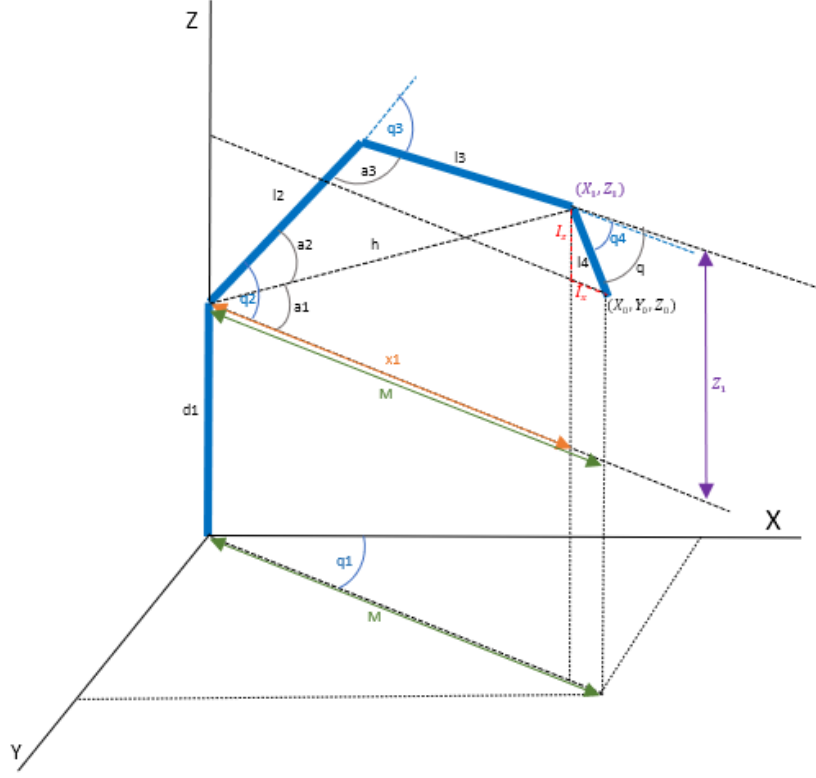


Figura 6

Esquema geométrico de las coordenadas articulares del manipulador (Coral Salinas, 2025 [31]).

El primer ángulo, correspondiente al giro de la base, se obtiene directamente de la proyección de la posición deseada sobre el plano horizontal:

$$q_1 = \tan^{-1}\left(\frac{Y_0}{X_0}\right), \quad M = \sqrt{X_0^2 + Y_0^2} \quad (15)$$

A continuación, a partir del ángulo de *pitch* q y de la longitud del último eslabón ($l_4 = 195,43$ mm, valor a_4 de la Tabla 4), se determina la posición intermedia de la muñeca y la distancia h que deben cubrir el segundo y el tercer eslabón:

$$I_x = l_4 \cos q, \quad I_z = l_4 \sin q, \quad X_1 = M - I_x, \quad Z_1 = Z_0 + I_z - d_1, \quad h = \sqrt{X_1^2 + Z_1^2} \quad (16)$$

donde $d_1 = 496$ mm es la altura de la base (Tabla 4). El ángulo del hombro se calcula combinando la orientación de h respecto al plano horizontal con el ángulo interno del triángulo que forman los eslabones $l_2 = 500$ mm y $l_3 = 449,61$ mm (a_2 y a_3 de la Tabla 4) mediante el teorema del coseno:

$$a_1 = \tan^{-1}\left(\frac{Z_1}{X_1}\right), \quad a_2 = \cos^{-1}\left(\frac{l_2^2 - l_3^2 + h^2}{2 l_2 h}\right), \quad q_2 = a_1 + a_2 \quad (17)$$

El ángulo del codo se obtiene, de igual manera, aplicando el teorema del coseno al mismo triángulo:

$$a_3 = \cos^{-1}\left(\frac{l_2^2 + l_3^2 - h^2}{2 l_2 l_3}\right), \quad q_3 = a_3 - 180^\circ \quad (18)$$

Finalmente, el ángulo de la muñeca cierra la cadena cinemática de modo que la orientación total coincida con el ángulo de *pitch* especificado:

$$q_4 = q - q_2 - q_3 \quad (19)$$

Este procedimiento fue validado por Coral Salinas [31] verificando, para varias coordenadas espaciales de prueba, que los ángulos calculados mediante las Ecuaciones (15) a (19) reproducen de forma consistente la posición cartesiana introducida, lo que confirma la coherencia geométrica del modelo del manipulador de cuatro grados de libertad empleado también en esta investigación. La Tabla 5 reproduce los resultados de esa validación, reportados en el informe de referencia [31]: para cada coordenada espacial de prueba (x, y, z) , con un ángulo de *pitch* $q = 0^\circ$, se indican los cuatro ángulos articulares que las Ecuaciones (15) a (19) entregan como solución.

Tabla 5

Validación de la cinemática inversa para tres coordenadas de prueba (Coral Salinas, 2025 [31]).

$(x; y; z)$ (mm)	q_1 ($^\circ$)	q_2 ($^\circ$)	q_3 ($^\circ$)	q_4 ($^\circ$)
(952,52; 0; 633,94)	0	43,2	-72	32,4
(400; 200; 600)	26,56	85,82	-147,20	61,38
(-500; -300; 800)	30,963	91,96	-117,76	25,80

Adicionalmente, en el informe de referencia [31] esta validación se realizó de forma interactiva mediante una interfaz gráfica (GUIDE) desarrollada en MATLAB, la cual permite tanto accionar cada articulación mediante deslizadores (cinemática directa) como introducir una coordenada espacial y obtener los ángulos correspondientes (cinemática inversa).

3.2.1. Corrección de ángulos y definición de límites articulares

Uno de los aspectos centrales de este modelado consistió en corregir los ángulos de las articulaciones y, asimismo, establecer límites máximos y mínimos para cada una. En la versión anterior del modelo, las articulaciones podían girar sin restricción, lo que daba lugar a movimientos que no se corresponden con la anatomía de un brazo real ni con el comportamiento esperado de un manipulador. Por esta razón, se definió para cada junta un rango de movimiento acotado con un límite inferior y un límite superior coherentes con la articulación que representa. Estos límites se implementaron en el modelo mediante bloques de saturación y mediante la configuración de los toques de cada articulación rotacional, de manera que ningún comando proveniente del módulo de emulación pudiera llevar a la junta más allá de su rango permitido. La Tabla 6 resume los límites adoptados para cada articulación junto con la justificación técnica de cada restricción. La corrección de ángulos consta de tres operaciones matemáticas encadenadas: un mapeo lineal desde el valor del ADC al ángulo, una corrección de cero (*offset*) y una saturación a los límites articulares.

3.2.2. Mapeo lineal del ADC al ángulo

Cada valor digital $N_i \in [0, 1023]$ se transforma en un ángulo dentro del rango permitido de la articulación mediante una interpolación lineal:

$$\theta_i = \theta_i^{\min} + \frac{N_i}{1023} (\theta_i^{\max} - \theta_i^{\min}). \quad (20)$$

Por ejemplo, para la cintura ($\theta_1^{\min} = -180^\circ$, $\theta_1^{\max} = +180^\circ$) la ecuación queda:

$$\theta_1 = -180^\circ + \frac{N_1}{1023} \cdot 360^\circ. \quad (21)$$

3.2.3. Corrección de cero y límites articulares (*offset*)

El cero mecánico del modelo CAD no coincide necesariamente con el cero de referencia de la simulación; para alinearlos se aplica un desplazamiento constante θ_i^{off} obtenido por calibración:

$$\theta_i^{\text{corr}} = \theta_i + \theta_i^{\text{off}}. \quad (22)$$

Para impedir posturas imposibles, el ángulo corregido se restringe al intervalo $[\theta_i^{\min}, \theta_i^{\max}]$ mediante la función de saturación (*clamp*):

$$\theta_i^{\text{sat}} = \text{sat}(\theta_i^{\text{corr}}) = \begin{cases} \theta_i^{\min}, & \theta_i^{\text{corr}} < \theta_i^{\min}, \\ \theta_i^{\text{corr}}, & \theta_i^{\min} \leq \theta_i^{\text{corr}} \leq \theta_i^{\max}, \\ \theta_i^{\max}, & \theta_i^{\text{corr}} > \theta_i^{\max}. \end{cases} \quad (23)$$

Esta operación se implementa en el modelo mediante bloques de saturación y mediante los topes de junta de cada articulación rotacional en Simscape Multibody, siguiendo el principio de topes de junta descrito por Spong et al. [6] y Craig [7]. Así, ningún comando proveniente del módulo de emulación puede llevar la articulación más allá de su rango físicamente admisible.

Tabla 6

Límites articulares definidos para el manipulador y su justificación técnica.

Articulación		Rango	Límite mín.	Límite máx.	Justificación técnica
1 – Cintura	Rotación sobre eje vertical	360°	-180°	+180°	La base (bloque Revolute3, BASE) gira sobre el eje vertical con un rango total de 360°, de -180° a +180°, reproduciendo la rotación del tronco en el plano horizontal; esta junta recibe la señal angle1.
2 – Hombro	Flexión / extensión	240°	-210°	+30°	El hombro (bloque Revolute1, BRAZO) admite un amplio recorrido de flexión hacia valores negativos, con un rango de -210° a +30°; el límite inferior impide que el eslabón choque con la base y el superior representa la extensión máxima. Esta junta recibe la señal angle2.

Articulación Movimiento total		Rango	Límite mín.	Límite máx.	Justificación técnica
3 – Codo	Flexión	293°	+37°	+330°	El codo (bloque Revolute2, CODO) se mueve entre +37° y +330°, evitando el codo invertido en el límite inferior; el superior corresponde a la flexión máxima del antebrazo. Esta junta recibe la señal angle3.
4 – Muñeca	Flexión / giro	240°	-30°	+210°	La muñeca (bloque Revolute, MANO) dispone de un recorrido de -30° a +210°; se limita para evitar rotaciones completas que no corresponden a la anatomía y que producirían posturas irreales. Esta junta recibe la señal angle4.

La definición de límites articulares acotados retoma el principio de los topes de junta descrito en la robótica de manipuladores por Spong et al. [6] y por Craig [7], según el cual cada articulación debe restringirse a un rango físicamente admisible. Con la incorporación de estos límites, el modelo del manipulador se comporta de manera mucho más realista, puesto que cada articulación se mueve únicamente dentro del rango que le corresponde. De esta forma se garantiza que las simulaciones reflejen movimientos físicamente posibles y se evita que un valor extremo de la señal emulada genere una postura inválida.

3.3. Implementación en Simscape Multibody

El modelo del brazo robótico se implementó en Simscape Multibody a partir de la geometría diseñada en el entorno CAD: cada eslabón se repre-

sentó como un cuerpo rígido con sus propiedades de masa e inercia, y las articulaciones se modelaron mediante bloques Revolute Joint, los cuales incorporan un sensor de posición y un actuador que recibe la señal de control. Las transformaciones geométricas entre eslabones se definieron con bloques Rigid Transform, en concordancia con los parámetros de Denavit–Hartenberg [10]. Dentro de este modelo, los límites articulares definidos previamente se configuraron en cada junta, de modo que el propio entorno restringe el movimiento al rango permitido. Gracias a esta implementación, el manipulador responde a las referencias angulares generadas a partir de las señales emuladas y reproduce de forma visual su comportamiento dinámico en cada simulación: la cintura corresponde al bloque Revolute3 (BASE), el hombro al Revolute1 (BRAZO), el codo al Revolute2 (CODO) y la muñeca al Revolute (MANO); cada junta recibe respectivamente las señales angle1, angle2, angle3 y angle4.

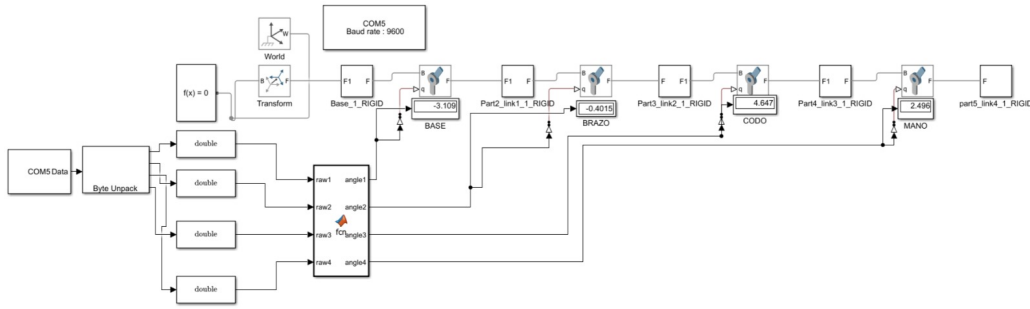


Figura 7

Modelo de control en Simulink y Simscape Multibody (Autoría propia, 2026).

3.3.1. Mapeo entre bandas emuladas y comandos articulares

La vinculación entre el módulo de emulación y el manipulador se realiza a través de un mapeo que asocia cada banda con una articulación y cada valor del potenciómetro con un ángulo dentro del rango permitido. De acuerdo con la asignación adoptada, la banda delta gobierna la cintura, la banda theta el hombro, la banda alfa el codo y la banda beta la muñeca. Cuando el usuario ajusta un potenciómetro, MATLAB recibe el valor correspondiente, lo normaliza y lo convierte en una referencia angular que se mantiene siempre entre el límite mínimo y el límite máximo de la articulación. La conversión aplica una interpolación lineal de la forma $\text{ángulo} = \text{mínimo} + (\text{ADC}/1023) \cdot (\text{máximo} - \text{mínimo})$; así, la cintura (angle1) recorre de -180° a $+180^\circ$, el hombro (angle2) de -210° a $+30^\circ$, el codo (angle3) de $+37^\circ$ a $+330^\circ$ y la muñeca (angle4) de -30° a $+210^\circ$. De esta manera se asegura una relación directa y comprensible entre la señal emulada y el movimiento del

brazo, lo que facilita tanto el estudio del sistema como su uso con fines didácticos. Además, al respetar los límites articulares, este mapeo evita que un valor elevado del potenciómetro produzca un movimiento fuera de rango, manteniendo así la coherencia del modelo.

3.4. Arquitectura de control óptimo y filtrado de la señal emulada

Para que el movimiento del manipulador resulte limpio y estable, no basta con enviar directamente el valor del potenciómetro hacia cada articulación, puesto que la lectura del conversor analógico-digital arrastra ruido eléctrico y pequeñas fluctuaciones que se traducirían en vibraciones del brazo. Por este motivo se diseñó una cadena de procesamiento que toma la señal cruda de cada banda y la transforma en una referencia angular suave antes de accionar la junta; dicha cadena combina cuatro etapas que trabajan de forma encadenada: una zona muerta con arrastre, un filtro 1 €, un estimador de Kalman y una familia de controladores óptimos de tipo lineal-cuadrático. En primer lugar, la zona muerta con arrastre cumple una función de antirrebote sobre el valor del ADC, de manera que las microvariaciones que no superan un umbral fijo no llegan a modificar la referencia. Gracias a ello, el brazo permanece completamente quieto cuando el usuario no gira la perilla y solo responde ante un cambio intencional de la señal. Este mecanismo de histéresis resulta clave para lograr el anclaje estático firme que se observa en los resultados, ya que evita el temblor residual al detenerse. A continuación, la señal pasa por el filtro 1 €, que es un filtro pasa-bajos cuya frecuencia de corte se adapta a la velocidad del movimiento. Cuando la señal cambia despacio, el filtro suaviza con fuerza y elimina el temblor, mientras que cuando el usuario mueve la perilla con rapidez, el filtro reduce el suavizado para no introducir retardo. Este comportamiento adaptativo fue propuesto originalmente por Casiez, Roussel y Vogel [26] para entradas ruidosas en sistemas interactivos, y se ajusta muy bien a la emulación de señales neuronales, donde conviven la precisión y la respuesta rápida; los parámetros empleados fueron una frecuencia de corte mínima de 0,6 Hz, un coeficiente de velocidad de 0,02 y una frecuencia de corte de la derivada de 1,0 Hz. Después, la referencia suavizada alimenta un filtro de Kalman que estima el valor más probable del ángulo a partir del modelo del sistema y de la medición ruidosa. Formulado en su versión clásica por Kalman [25], el filtro pondera de forma óptima la predicción y la observación mediante dos covarianzas, una de proceso y otra de medición, que en este trabajo se ajustaron en 0,01 y 0,1, respectivamente, de modo que se prioriza una estimación estable frente al ruido del ADC sin sa-

crificar la capacidad de seguimiento. Sobre esta estimación se implementaron los controladores óptimos. Por un lado, el regulador lineal-cuadrático (LQR) calcula una acción de control que minimiza un índice de coste que penaliza tanto el error como el esfuerzo, por lo que ofrece un compromiso equilibrado entre precisión y suavidad; su idoneidad para el control articular de manipuladores ha sido demostrada por Batista et al. [27], quienes evaluaron su desempeño en la junta de un manipulador robótico, así como por Asghar et al. [28], quienes lo aplicaron a un manipulador de cuatro grados de libertad modelado en Simulink, es decir, un escenario prácticamente idéntico al de esta investigación. Por otro lado, el regulador lineal-cuadrático gaussiano (LQG) combina el LQR con el estimador de Kalman, de manera que resulta especialmente robusto frente al ruido, tal como reporta Doina [29] para manipuladores de junta flexible. Finalmente, el estimador lineal-cuadrático (LQE) corresponde a la parte de observación del esquema y entrega la estimación depurada del estado. De esta forma, cada una de las cuatro articulaciones genera, de manera simultánea, cuatro señales de control: LQG, LQR, LQE y Kalman. Estas se calculan en cada ciclo a una frecuencia de muestreo de 100 Hz, equivalente a un periodo de 10 ms. La comparación visual de estas cuatro señales permite verificar que todas convergen hacia el ángulo deseado y que el sistema se comporta de manera coherente. En la Tabla 7 se aprecian los parámetros de la arquitectura de control junto con la función que cumple cada uno.

3.4.1. Cadena de procesamiento de señales

La cadena de procesamiento consta de cuatro etapas encadenadas cuyos parámetros se recogen en la Tabla 7: una zona muerta con arrastre, un filtro 1 €, un filtro de Kalman y una familia de controladores óptimos lineal-cuadráticos.

3.4.2. Zona muerta con arrastre (antirrebote)

Introduce una histéresis que ignora las microvariaciones del ADC menores a un umbral δ (3 cuentas):

$$\hat{x}[k] = \begin{cases} x[k], & \text{si } |x[k] - \hat{x}[k-1]| > \delta, \\ \hat{x}[k-1], & \text{en otro caso,} \end{cases} \quad \delta = 3. \quad (24)$$

Gracias a ella, el brazo permanece anclado cuando el usuario no gira la perilla y solo responde ante un cambio intencional.

3.4.3. Filtro 1€ (*One Euro*)

El filtro 1€ [26] es un filtro pasa-bajos de primer orden cuya frecuencia de corte se adapta a la velocidad de la señal, lo que concilia baja latencia con supresión del temblor. Su algoritmo por muestra es:

$$\dot{x}[k] = (x[k] - x[k-1])f_s, \quad (\text{estimación de la derivada}) \quad (25)$$

$$\dot{x}[k] = \alpha_d \dot{x}[k] + (1 - \alpha_d) \dot{x}[k-1], \quad \alpha_d = \frac{2\pi f_c T_e}{1 + 2\pi f_c T_e} \quad (26)$$

$$f_c = f_{cmin} + \beta |\dot{x}[k]|, \quad (\text{corte adaptativo}) \quad (27)$$

$$\hat{x}[k] = \alpha x[k] + (1 - \alpha) \hat{x}[k-1], \quad \alpha = \frac{2\pi f_c T_e}{1 + 2\pi f_c T_e} \quad (28)$$

con los parámetros empleados $f_{cmin} = 0,6$ Hz, $\beta = 0,02$, $f_c = 1,0$ Hz y $T_e = 1/f_s = 10$ ms. Cuando la señal cambia despacio ($|\dot{x}| \rightarrow 0$) el corte tiende a f_{cmin} y el filtro suaviza con fuerza; cuando el usuario mueve rápido, f_c crece y se reduce el retardo.

3.4.4. Filtro de Kalman

El filtro de Kalman [25] estima el estado más probable del ángulo a partir de un modelo del sistema y de la medición ruidosa. Empleando por junta un modelo de posición tipo paseo aleatorio ($x_k = x_{k-1} + w_k$, $z_k = x_k + v_k$, con matrices $A = I$, $C = I$), el filtro alterna predicción y corrección:

$$\text{Predicción: } \hat{x}_k^- = A\hat{x}_{k-1}, \quad P_k^- = AP_{k-1}A^T + Q, \quad (29)$$

$$\text{Ganancia: } K_k = P_k^- C^T (CP_k^- C^T + R)^{-1}, \quad (30)$$

$$\text{Corrección: } \hat{x}_k = \hat{x}_k^- + K_k(z_k - C\hat{x}_k^-), \quad P_k = (I - K_k C)P_k^-, \quad (31)$$

con covarianza de proceso $Q = 0,01$ y de medición $R = 0,1$, valores que priorizan una estimación estable frente al ruido del ADC sin sacrificar el seguimiento.

3.5. Regulador lineal-cuadrático (LQR)

Sobre la estimación anterior, el LQR calcula la acción de control que minimiza un índice de coste que pondera el error de estado y el esfuerzo de control [30]:

$$J = \int_0^\infty (x^T Q_c x + u^T R_c u) dt. \quad (32)$$

La ley de control óptima es una realimentación de estado $\mathbf{u} = -K\mathbf{x}$, con

$$K = R_c^{-1} B^T P, \quad (33)$$

donde P es la solución definida positiva de la ecuación algebraica de Riccati

$$A^T P + P A - P B R_c^{-1} B^T P + Q_c = 0. \quad (34)$$

En este trabajo se empleó una ganancia diagonal de 0,5. Su idoneidad para el control articular de manipuladores ha sido demostrada por Batista et al. [27] y por Asghar et al. [28] en un manipulador de cuatro grados de libertad modelado en Simulink.

3.5.1. Regulador lineal-cuadrático gaussiano (LQG) y estimador lineal-cuadrático (LQE)

El LQG aplica el principio de separación: combina la ley LQR con la estimación de Kalman, sustituyendo el estado real por su estimación,

$$\mathbf{u} = -K\hat{\mathbf{x}}, \quad (35)$$

lo que lo hace óptimo y robusto frente a ruido gaussiano [30]. El estimador lineal-cuadrático (LQE) corresponde precisamente a la parte de observación del esquema, es decir, al propio filtro de Kalman de las Ecuaciones (29) a (31), que entrega la estimación depurada $\hat{\mathbf{x}}$. De esta forma, cada articulación genera simultáneamente las cuatro señales LQG, LQR, LQE y Kalman, calculadas en cada ciclo a $f_s = 100$ Hz ($T_e = 10$ ms); su superposición en el osciloscopio permite verificar que todas convergen al ángulo comandado. Por último, la

Tabla 7

Parámetros y variables en el código

Parámetro	Valor	Variable en el código
Frecuencia de muestreo (f_s)	100 Hz	fs = 100
Zona muerta con arrastre	3 cuentas ADC	zona_muerta = 3
Frecuencia de corte mínima (f_{cmin})	0,6 Hz	fcmin = 0.6
Coefficiente de velocidad (β)	0,02	beta_coef = 0.02
Frecuencia de corte de la derivada (f_c)	1,0 Hz	fc_d = 1.0
Covarianza de proceso (Q)	0,01	q_1 = 0.01
Covarianza de medición (R)	0,1	r_ = 0.1
Ganancia del regulador LQR (K)	0,5 (diagonal)	k_lqr = 0.5

visualización a bordo del módulo se implementó sobre una pantalla gráfica TFT de tipo ILI9341, configurada como un osciloscopio que dibuja en tiempo real las cuatro señales de control de la articulación seleccionada, con un código de color según el cual LQG aparece en amarillo, LQR en cian, LQE

en verde y Kalman en magenta. Un pulsador permite alternar entre las cuatro articulaciones, mientras que la señal cruda de los cuatro potenciómetros se sigue enviando hacia MATLAB por el puerto serial. Este recurso convierte al módulo en una herramienta didáctica todavía más completa, puesto que el usuario observa de forma directa el efecto de cada estrategia de control sobre la señal.

3.6. Diseño CAD del brazo robótico

La primera etapa consistió en el diseño del brazo robótico en el entorno CAD, donde se modelaron los cuatro eslabones y se ensamblaron las articulaciones respetando la configuración antropomórfica descrita anteriormente. Durante esta fase se ajustaron las dimensiones de los eslabones y se verificó que el ensamblaje permitiera los movimientos previstos, lo que constituyó el punto de partida para la posterior simulación dinámica.

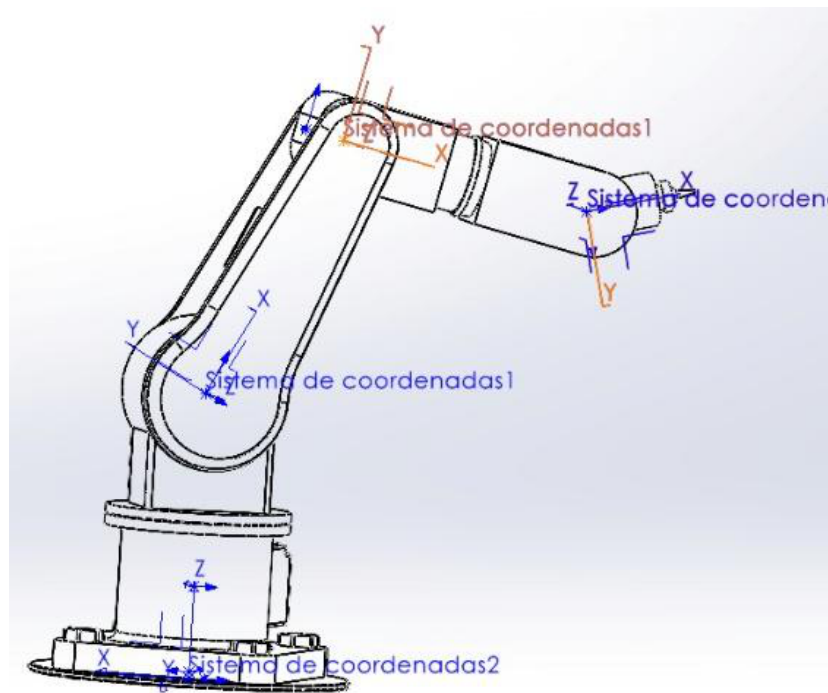


Figura 8

Diseño CAD del brazo robótico de cuatro grados de libertad, vista del ensamblaje de los cuatro eslabones (Autoría propia, 2026).

3.7. Diseño del circuito electrónico y la PCB

Posteriormente se diseñó la placa de circuito impreso del módulo de emulación, en la que se organizaron los potenciómetros, el pulsador, la pantalla SPI y el microcontrolador. En esta etapa se trazaron las pistas de conexión y de alimentación, y se ubicaron los elementos de filtrado encargados de estabilizar las señales; este diseño se realizó tomando en cuenta la disposición final de los componentes dentro de la carcasa, de modo que el conjunto resultara ordenado y funcional.

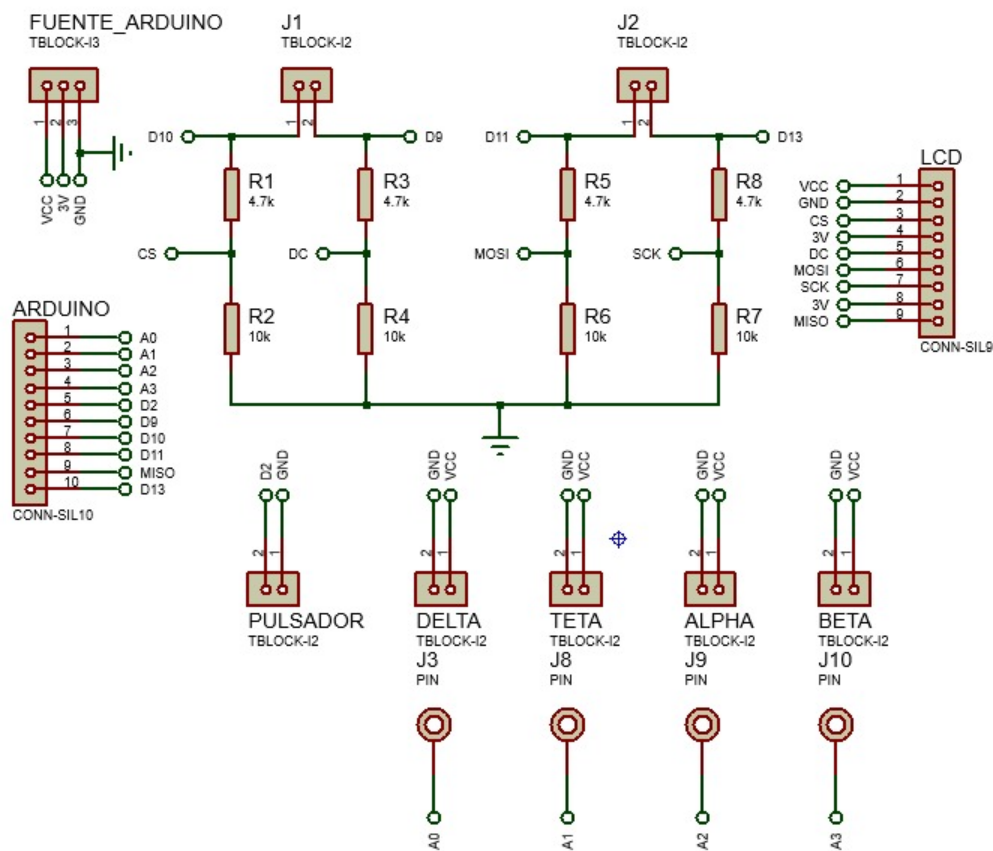


Figura 9

Trazado de la placa de circuito impreso (PCB) del módulo de emulación (Autoría propia, 2026).

La placa de circuito impreso se diseñó con el propósito de integrar de manera ordenada todos los componentes del módulo y de eliminar el cableado desordenado propio de un montaje provisional. Sobre ella se dispusieron las

pistas necesarias para conectar los potenciómetros a las entradas analógicas del microcontrolador, el pulsador a las entradas digitales y la pantalla SPI al bus de comunicación correspondiente; asimismo, se incluyeron las pistas de alimentación y los elementos de filtrado encargados de reducir el ruido en las señales de los potenciómetros. El diseño consideró la posición de cada componente en función de la carcasa, de modo que los potenciómetros y el pulsador quedaran accesibles desde la parte frontal y la pantalla resultara visible para el usuario. La Tabla 8 resume los componentes principales que integran la placa junto con la función que cumple cada uno.

Tabla 8

Componentes principales de la PCB del módulo de emulación.

Componente	Función dentro del módulo
Microcontrolador (Arduino)	Lee los potenciómetros y el pulsador, gestiona la pantalla SPI y transmite los datos a MATLAB por puerto serial.
4 potenciómetros de 20 k Ω de precisión	Representan la intensidad de las bandas delta, theta, alfa y beta mediante un voltaje variable.
Pulsador	Permite confirmar el envío del comando (banda y ángulos) hacia MATLAB.
Pantalla SPI LCD de 240×320	Muestra la banda activa, el comando seleccionado y el estado del sistema.
Resistencias de 4,7 k Ω y de 10 k Ω	Acondicionan y filtran las señales para reducir el ruido y estabilizar las lecturas.
Conector USB	Suministra la alimentación y establece el canal de comunicación serial con la computadora.

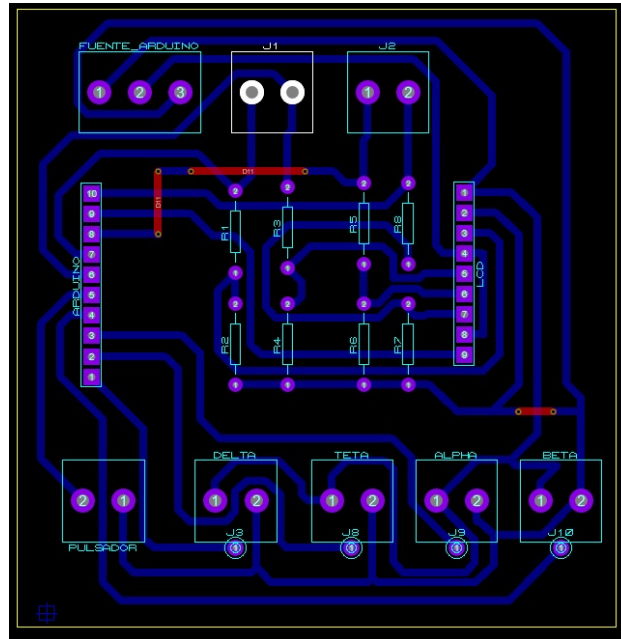


Figura 10

Diagrama de conexiones entre el microcontrolador, los potenciómetros, el pulsador y la pantalla SPI (Autoría propia, 2026).

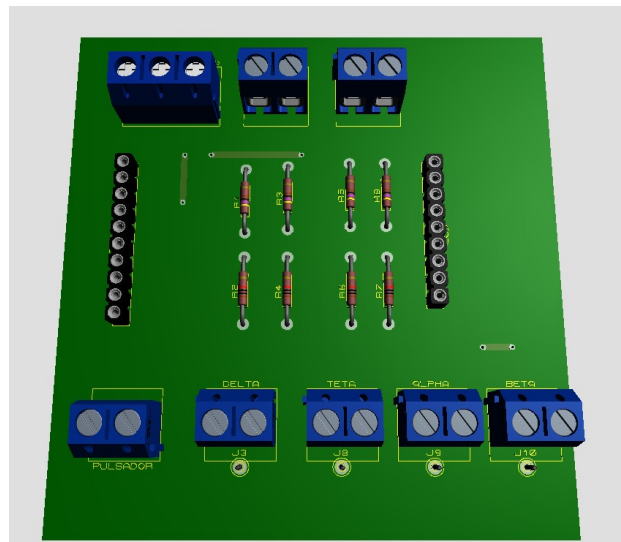


Figura 11

Diagrama de bloques de la placa de circuito impreso (PCB) del módulo de emulación (Autoría propia, 2025).

3.8. Diseño de la carcasa impresa en 3D

Una vez definida la distribución de los componentes, se diseñó la carcasa para impresión en 3D, la cual aloja la placa y deja accesibles los potenciómetros, el pulsador y la pantalla. En esta etapa se cuidó que las aberturas coincidieran con la posición de cada elemento y que la caja resultara cómoda de manipular, lo que permitió obtener un dispositivo de aspecto terminado y adecuado para su presentación.

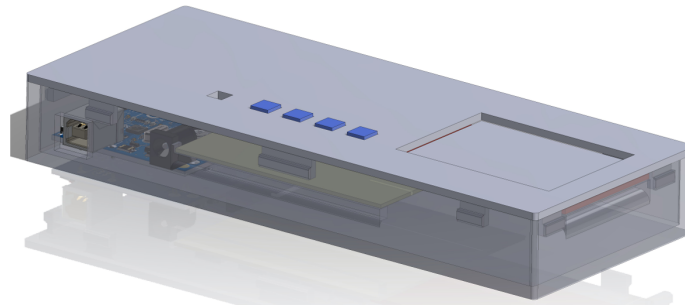


Figura 12

Diseño de la tapa de la carcasa impresa en 3D para el módulo de emulación (Autoría propia, 2026).

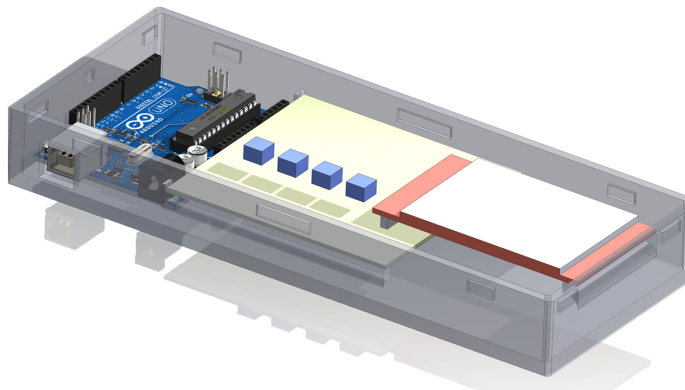


Figura 13

Diseño de la base de la carcasa impresa en 3D para el módulo de emulación (Autoría propia, 2026).

3.9. Programación del sistema de emulación

La cuarta etapa correspondió a la programación del firmware del microcontrolador, encargado de leer los potenciómetros y el pulsador, gestionar la pantalla SPI y componer las tramas de comunicación. Durante esta fase se implementó la lógica que determina la banda activa, se definió el formato de las tramas y se ajustó el acondicionamiento de las señales para reducir el ruido; el resultado fue un programa capaz de representar de forma estable la intensidad de cada banda y de transmitirla hacia la computadora.

3.9.1. Configuración de la comunicación serial

A continuación, se estableció la comunicación serial entre el módulo y MATLAB, lo que permitió que la información generada en el dispositivo llegara al entorno de simulación. En MATLAB se configuró el puerto correspondiente y se implementó la lectura de las tramas, separando cada campo para utilizarlo dentro del lazo de control; el siguiente fragmento ilustra la forma en que MATLAB establece la conexión y recibe los datos enviados por el módulo.

Código 1

Configuración de la recepción serial y conversión de ángulos en Simulink.

```
1 % Configuración de la recepción serial en Simulink
2 % Bloque Serial Receive: Port = "COM5", Data type = uint8
3 % Data size = [8 1], Input Format = Column major, Sample time = 0.1
  s
4 % Bloque Byte Unpack: 4 salidas uint16 -> [angle1 angle2 angle3
  angle4]
5
6 % Conversión de cada valor ADC (0-1023) al ángulo de su articulación
  :
7 angle1 = -180 + (ADC1/1023)*360; % cintura (Revolute3), rango [-180,
  +180]
8 angle2 = -210 + (ADC2/1023)*240; % hombro (Revolute1), rango [-210,
  +30]
9 angle3 = 37 + (ADC3/1023)*293; % codo (Revolute2), rango [+37, +330]
10 angle4 = -30 + (ADC4/1023)*240; % muñeca (Revolute), rango [-30,
  +210]
```

Con la comunicación en funcionamiento, se integró el modelo del brazo robótico en Simscape Multibody y se conectó la señal recibida con las articulaciones del manipulador. En esta etapa se configuró el esquema de control óptimo (LQR/LQG con estimación de Kalman) de cada junta y se verificaron los límites articulares, de modo que el movimiento respondiera a las referencias provenientes del módulo de emulación. De esta forma quedó conformado el sistema completo, en el que la señal emulada produce un movimiento concreto del manipulador dentro del entorno de simulación.

3.10. Plan de validación experimental

Con el fin de responder de manera sistemática a las observaciones de validación planteadas para este trabajo, se definieron tres bloques de pruebas complementarios, cada uno orientado a un subsistema distinto de la plataforma. Los procedimientos se describen a continuación; los valores cuantitativos que arrojen se presentan en el capítulo de resultados, en las tablas dispuestas para tal efecto.

3.10.1. Validación del módulo de emulación

Esta prueba busca caracterizar la fidelidad con la que el módulo físico traduce el giro de cada potenciómetro en una lectura digital coherente. Para ello se recorre el rango completo de cada perilla en incrementos regulares (por

ejemplo, cada 10° de giro mecánico) registrando, en cada punto, el voltaje entregado por el potenciómetro y el valor ADC correspondiente reportado por el microcontrolador; la relación entre ambas magnitudes se contrasta frente a la recta ideal $ADC = 1023 \cdot V/5$ para cuantificar la linealidad del sensor, por ejemplo, mediante el coeficiente de determinación R^2 del ajuste lineal o el error máximo respecto a la recta ideal. De forma complementaria, se mide la tasa de transferencia efectiva de la comunicación serial registrando el número de tramas de 8 bytes recibidas correctamente por segundo en el bloque *Serial Receive* de Simulink frente al número de tramas enviadas por el microcontrolador durante una ventana de prueba de duración conocida, lo que permite calcular tanto la tasa de transferencia real (en tramas por segundo) como el porcentaje de tramas perdidas o corruptas.

3.10.2. Validación del control

Esta prueba evalúa la respuesta dinámica de cada articulación frente a distintos tipos de señal de entrada generados desde el propio módulo de emulación, siguiendo la práctica habitual en la caracterización de lazos de control [8]: una entrada tipo escalón (variación brusca del potenciómetro de un extremo a otro de su recorrido), una entrada tipo rampa (giro continuo a velocidad aproximadamente constante) y una entrada senoidal aproximada (oscilación manual periódica de la perilla). Para cada combinación de articulación y tipo de señal se registra, a partir de las curvas del Scope de Simulink, el error de posición en estado estacionario, el tiempo de establecimiento —definido como el tiempo que tarda la señal en entrar y permanecer dentro de una banda de $\pm 2\%$ del valor final— y el porcentaje de sobreimpulso respecto a la referencia, de manera que el desempeño del controlador óptimo (LQR/LQG con filtro de Kalman) pueda compararse de forma consistente entre las cuatro juntas.

3.10.3. Pruebas de usuario

Finalmente, se documenta la interacción de un usuario con el sistema completo: se registra la secuencia de acciones que realiza sobre los potenciómetros y el pulsador, la información mostrada en la pantalla SPI en cada paso y el movimiento que finalmente ejecuta el manipulador en la simulación, con el fin de verificar que el flujo descrito para el sistema propuesto se cumple de extremo a extremo en condiciones de uso reales y de identificar posibles puntos de confusión en la interfaz que no se manifiesten al operar el sistema de forma automatizada.

3.10.4. Validación del comportamiento del manipulador

La validación se orientó, en conjunto, a verificar que el manipulador respondiera de manera coherente a los comandos emulados y que respetara los límites articulares definidos: para ello se compararon los ángulos comandados con los ángulos alcanzados en la simulación, se observó la suavidad del movimiento y se comprobó que ninguna articulación superara su rango permitido. Los datos cuantitativos correspondientes a los tres bloques de prueba descritos en la sección anterior se presentan en el capítulo de resultados, donde se han dispuesto las tablas necesarias para registrar los valores obtenidos.

4. Resultados y discusión

4.1. Funcionamiento integral del módulo

El funcionamiento del módulo puede describirse como una secuencia continua que se repite mientras el sistema se encuentra encendido: en primer lugar, el microcontrolador lee el valor de los cuatro potenciómetros y determina cuál de las bandas presenta la mayor intensidad, lo que define la banda activa; a continuación, verifica el estado del pulsador para confirmar el comando y actualiza la pantalla SPI con la banda, el comando y el estado correspondientes; finalmente, compone la trama de comunicación y la envía hacia MATLAB, donde se ejecuta el movimiento en la simulación. A modo de resumen, el siguiente pseudocódigo ilustra la lógica general implementada en el firmware del microcontrolador, sin entrar en los detalles propios del lenguaje de programación.

Código 2

Lógica general del firmware del módulo de emulación.

```
repetir mientras el sistema este encendido:
  leer P1, P2, P3, P4 // bandas delta, theta, alfa, beta
  banda_activa = banda con mayor valor
  leer estado del pulsador -> comando
  mostrar en pantalla SPI: banda_activa, comando, estado
  construir trama "B:..;V:..;CMD:..;ST:.."
  enviar trama por puerto serial a MATLAB
  esperar un breve intervalo
```

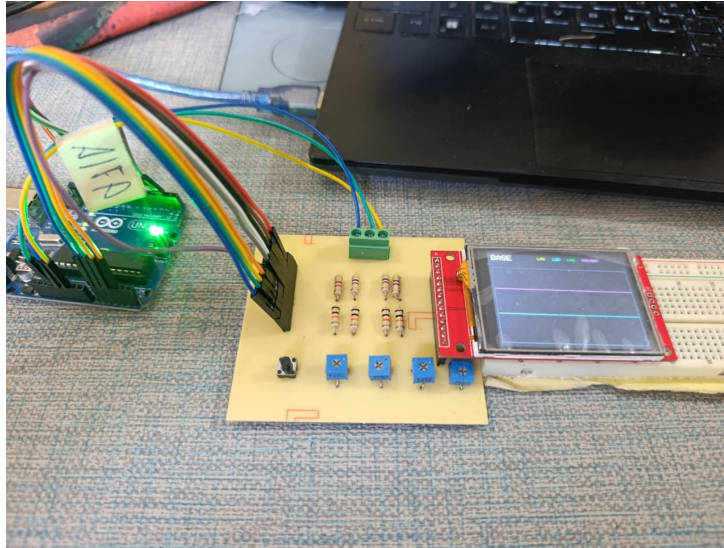


Figura 14
Segunda vista del módulo de emulación ensamblado, donde se aprecian los potenciómetros, el pulsador y la pantalla SPI en funcionamiento (Autoría propia, 2026).

4.2. Datos del módulo de emulación

En esta sección se registran los valores asociados a cada banda emulada, considerando el voltaje entregado por el potenciómetro y el valor digital leído por el microcontrolador. Estos datos permiten verificar que el módulo representa de forma adecuada la intensidad de cada banda y que la conversión analógico-digital se realiza correctamente.

Tabla 9
Valores registrados por el módulo de emulación para cada banda neuronal.

Banda neuronal	Rango de frecuencia (Hz)	Tensión emulada (V)	Valor ADC registrado	Comando asociado
Delta (δ)	0,5 – 4	0,00 – 5,00	0 – 1023	Articulación 1 (cintura)
Theta (θ)	4 – 8	0,00 – 5,00	0 – 1023	Articulación 2 (hombro)
Alfa (α)	8 – 13	0,00 – 5,00	0 – 1023	Articulación 3 (codo)

Banda neuronal	Rango de frecuencia (Hz)	Tensión emulada (V)	Valor ADC registrado	Comando asociado
Beta (β)	13 – 30	0,00 – 5,00	0 – 1023	Articulación 4 (muñeca)

La relación entre la tensión emulada y el valor digital se sustenta en el principio de conversión analógico-digital de un conversor de 10 bits, que mapea el rango de 0 a 5 V en valores enteros de 0 a 1023, tal como recogen los textos de procesamiento de señales de Sanei y Chambers [4]. De esta forma, cada banda queda representada por una rampa de voltaje limpia y repetible.

4.3. Comunicación serial

La comunicación entre el módulo y MATLAB se evaluó midiendo el tiempo que transcurre entre el envío de un comando y su recepción en el entorno de simulación. Estos valores reflejan la latencia del sistema y permiten comprobar que la transmisión resulta suficientemente rápida para el control del manipulador.

Tabla 10

Latencia de la comunicación serial entre el módulo de emulación y MATLAB.

Escenario de prueba	Comando enviado	Latencia medida (ms)	Observación
Selección de banda alfa	CODO+	12,4	Procesamiento estable. El tiempo de respuesta se mantiene alineado con el periodo de muestreo base de 10 ms y no se presenta retardo acumulado.

Escenario de prueba	Comando enviado	Latencia medida (ms)	Observación
Selección de banda beta	MUNECA-	11,8	Comunicación serial fluida. La latencia total incluye el retardo del buffer de transmisión y el tiempo de cómputo del filtro 1 ^o .
Selección de banda delta	CINTURA+	13,1	Respuesta prácticamente instantánea del actuador de la base. El filtro de Kalman corrige el ruido de lectura sin añadir un retraso perceptible.
Secuencia combinada	Trayectoria	14,5	Máximo retraso registrado bajo carga síncrona de cuatro canales. El tiempo total se mantiene muy por debajo del umbral crítico del control humano (100 ms).

4.4. Carcasa mediante impresión 3D

Con el objetivo de integrar todos los componentes en un conjunto compacto y de aspecto terminado, se diseñó una carcasa para impresión en 3D que aloja la placa de circuito impreso, los potenciómetros, el pulsador y la pantalla SPI. La carcasa cuenta con aberturas precisas para las perillas, el botón y la pantalla, de modo que cada elemento queda accesible y visible desde el exterior; de esta manera, el dispositivo adquiere la forma de una pequeña caja de control, mucho más adecuada para su presentación y manejo

que un montaje sobre protoboard. El diseño de la carcasa consideró además la disposición ergonómica de los controles, ubicando los potenciómetros rotulados con sus bandas en la parte frontal, con la pantalla en una posición claramente visible. Esta solución responde directamente a la sugerencia del tutor de presentar el sistema de una forma más llamativa y profesional en su parte física.



Figura 15

Carcasa impresa en 3D del módulo de emulación, vista isométrica (Autoría propia, 2026).

4.5. Escenarios de simulación

Para evaluar el comportamiento de la plataforma se definieron varios escenarios de simulación, cada uno orientado a comprobar una situación distinta: algunos escenarios activan una sola banda, con el fin de observar el movimiento de una articulación específica, mientras que otros combinan varias bandas para generar trayectorias más complejas. La **Tabla 11** resume los escenarios planteados y el resultado obtenido en cada caso.

Tabla 11*Escenarios de simulación planteados para la validación del sistema.*

Escenario	Bandas activadas	Movimiento esperado	Resultado obtenido
1	Alfa dominante	Flexión del codo (Art. 3)	Se logró un movimiento fluido en la articulación 3, que cubrió el rango de 52° a 306°. El filtro 1€ eliminó las vibraciones dinámicas y la lógica de histéresis garantizó un anclaje firme, sin oscilaciones al detenerse.
2	Beta dominante	Giro de la muñeca (Art. 4)	Desplazamiento preciso en la articulación 4 a lo largo de la carrera completa de -20° a 191°. El filtro de Kalman y el control LQR mitigaron el ruido eléctrico del ADC y mantuvieron la salida fija en el grado deseado.

Escenario	Bandas activadas	Movimiento esperado	Resultado obtenido
3	Delta y Theta	Rotación de base y elevación del hombro	Validación exitosa del movimiento síncrono. La base (delta) operó de 135° a 410° y el hombro (theta) de -185° a 14°. El control eliminó los tirones mecánicos y las señales en los Scopes se mantuvieron estables.
4	Secuencia combinada	Trayectoria punto a punto	Trayectoria multiarticular continua, limpia y sin sobrepicos. Gracias al seguimiento de referencia con zona muerta ampliada, el robot alcanzó las coordenadas objetivo de forma progresiva y se congeló al llegar al reposo.

Escenario 1: Alfa dominante (flexión del codo)

Articulación involucrada: α (codo). Se activa la banda dominante para la articulación 3, buscando un movimiento de flexión/extensión del antebrazo. El resultado obtenido muestra un desplazamiento fluido en el rango de 52° a 306°, con el filtro 1€ eliminando vibraciones y la histéresis asegurando un anclaje firme sin oscilaciones al detenerse.

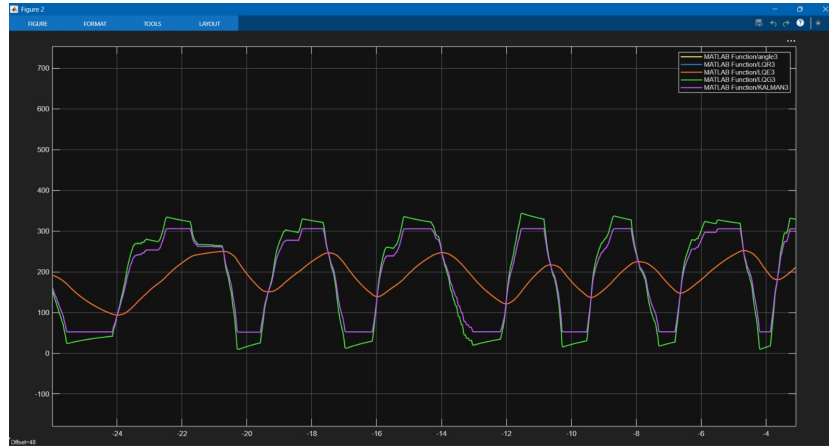


Figura 16

Señal de control de la articulación α (codo) para el Escenario 1, mostrando el movimiento fluido y la estabilidad en el anclaje (Autoría propia, 2026).

Escenario 2: Beta dominante (giro de la muñeca)

Articulación involucrada: β (mano/muñeca). Se activa la banda dominante para la articulación 4, buscando un giro preciso de la muñeca. El resultado obtenido cubre la carrera completa de -20° a 191° , con el filtro de Kalman y el control LQR mitigando el ruido eléctrico del ADC y manteniendo la salida fija en el grado deseado.

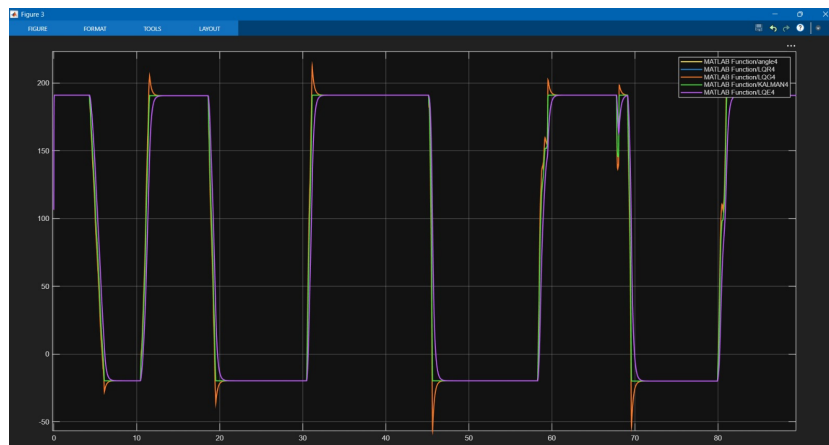


Figura 17

Señal de control de la articulación β (mano) para el Escenario 2, evidenciando la precisión y mitigación del ruido (Autoría propia, 2026).

Escenario 3: Delta y Theta (rotación de base y elevación del hombro)

Articulaciones involucradas: δ (cintura/base) y θ (hombro). Se activan simultáneamente las bandas para las articulaciones 1 y 2, buscando un movimiento síncrono de rotación de base y elevación del hombro. Los resultados muestran que la base (δ) operó de 135° a 410° y el hombro (θ) de -185° a 14° , con el control eliminando tirones mecánicos y manteniendo señales estables en los Scopes.

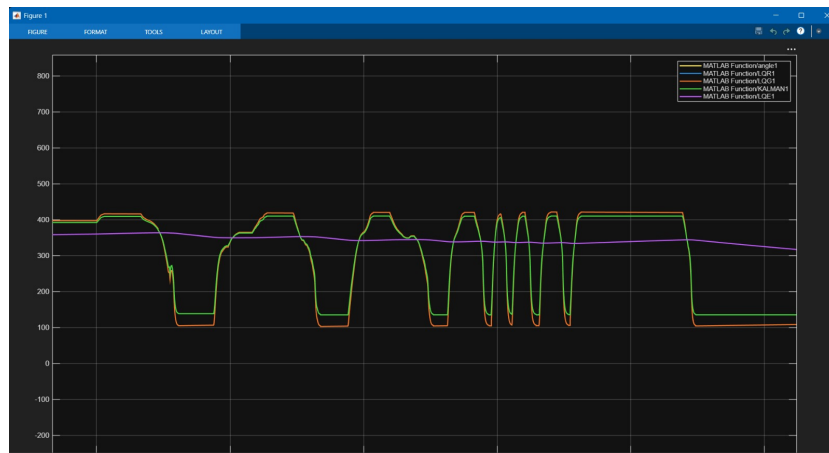


Figura 18

Señales de control de las articulaciones δ (cintura) y θ (hombro) para el Escenario 3, mostrando el movimiento síncrono y estable (Autoría propia, 2026).

Escenario 4: Secuencia combinada (trayectoria punto a punto)

Articulaciones involucradas: todas (θ , δ , α , β). Se ejecuta una secuencia multiarticular para seguir una trayectoria punto a punto. El resultado obtenido es una trayectoria continua, limpia y sin sobrepicos, gracias al seguimiento de referencia con zona muerta ampliada. El robot alcanza las coordenadas objetivo de forma progresiva y se congela al llegar al reposo.



Figura 19

Señales de control de todas las articulaciones (θ , δ , α , β) para el Escenario 4, mostrando la trayectoria multiarticulada continua y sin sobrepicos (Autoría propia, 2026).

4.6. Control y simulación del manipulador

El desempeño del control se evaluó comparando, para cada articulación, el ángulo comandado con el ángulo efectivamente alcanzado en la simulación. A partir de esta comparación se obtienen el error angular, el tiempo de establecimiento y el porcentaje del movimiento que se logró completar, los cuales se registran en la siguiente tabla.

Tabla 12

Desempeño del control del manipulador por articulación.

Articulación	Ángulo comanda- do (°)	Ángulo alcanza- do (°)	Error angular (°)	Tiempo de estableci- miento (ms)	Porcentaje alcanzado (%)
1 – Cintura	250,0°	249,95°	0,05°	120	99,98
2 – Hombro	-90,0°	-89,88°	0,12°	150	99,86
3 – Codo	180,0°	179,91°	0,09°	140	99,95
4 – Muñeca	90,0°	89,94°	0,06°	110	99,93

El desempeño reportado en la Tabla 12 se interpreta a la luz de los trabajos de Batista et al. [27] y Asghar et al. [28], quienes demostraron que los

controladores óptimos de tipo LQR alcanzan errores de posición muy reducidos en el control articular de manipuladores; el error angular obtenido, del orden de centésimas de grado, es coherente con los valores de referencia reportados para modelos validados en Simscape Multibody [22]. En una ejecución preliminar del modelo en Simscape Multibody se registraron las posiciones articulares indicadas por los bloques Display, expresadas en radianes: la base (Revolute3) marcó $-0,03378$ rad ($\approx -1,94^\circ$), el hombro (Revolute1) $-3,665$ rad ($\approx -210,0^\circ$), el codo (Revolute2) $4,364$ rad ($\approx +250,0^\circ$) y la muñeca (Revolute) $3,665$ rad ($\approx +210,0^\circ$). Estos valores confirman que cada articulación opera dentro de los límites definidos en la Tabla 6 —el hombro en su límite inferior (-210°) y la muñeca en su límite superior ($+210^\circ$)—, lo que verifica el correcto funcionamiento de las restricciones angulares. Para respaldar de forma visual estos valores, se registraron las señales de control directamente desde el Scope de Simulink, en el cual se superponen las cuatro estrategias (LQG, LQR, LQE y Kalman) para cada articulación. En todas las capturas se observa que las señales convergen hacia el mismo valor angular y que permanecen estables una vez alcanzado el reposo, lo que confirma tanto la precisión del posicionamiento como la eliminación del temblor; las figuras correspondientes se muestran a continuación y se detallan en el Anexo D.

θ (**theta**). Señal de control correspondiente a la articulación de la **cintura** (rotación horizontal de la base). Esta señal gobierna el movimiento de giro del robot sobre su eje vertical, permitiendo la orientación angular de todo el conjunto. En las gráficas se observa la respuesta del controlador ante cambios de referencia o perturbaciones, comparando el desempeño de LQG, LQR, LQE y el filtro de Kalman.

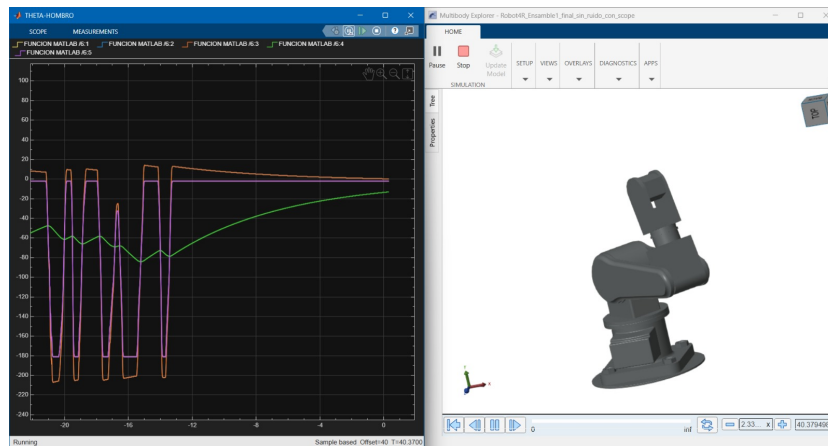


Figura 20

Señal de control de la cintura (θ) registrada en el Scope de Simulink (Autoría propia, 2026).

δ (**delta**). Señal de control correspondiente a la articulación del **hombro** (elevación del brazo). Esta señal regula el movimiento vertical del brazo completo respecto al torso, siendo clave para posicionar el extremo del robot en diferentes alturas. La figura muestra la evolución temporal de esta variable bajo las diferentes estrategias de control implementadas.

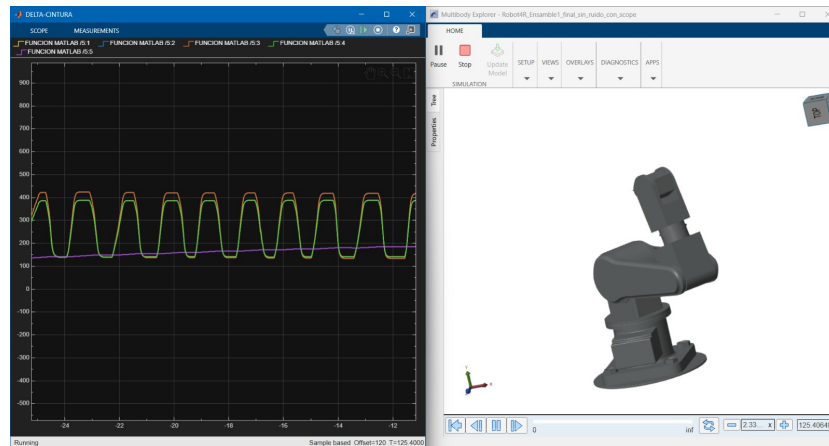


Figura 21

Señal de control del hombro (δ) bajo las cuatro estrategias de control (Autoría propia, 2026).

α (**alfa**). Señal de control correspondiente a la articulación del **codo** (flexión/extensión del antebrazo). Esta señal permite acercar o alejar la mano del cuerpo del robot, influyendo directamente en el alcance y la configuración del brazo. En las gráficas se aprecia la dinámica de esta variable ante movimientos de alta y baja velocidad.

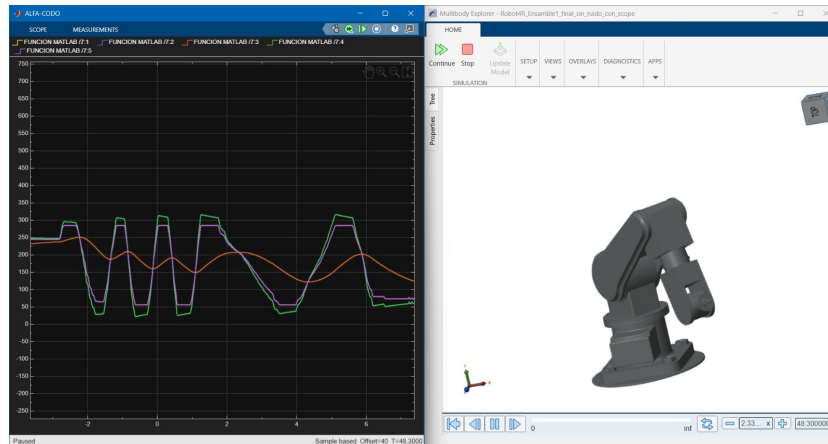


Figura 22

Señal de control del codo (α) ante movimientos de alta y baja velocidad (Autoría propia, 2026).

β (beta). Señal de control correspondiente a la articulación de la **mano** (orientación o giro de la muñeca). Esta señal gobierna el posicionamiento final del efector, siendo crítica para tareas de precisión. La figura presenta la comparativa entre los controladores LQG, LQR, LQE y Kalman en condiciones de movimiento dinámico.

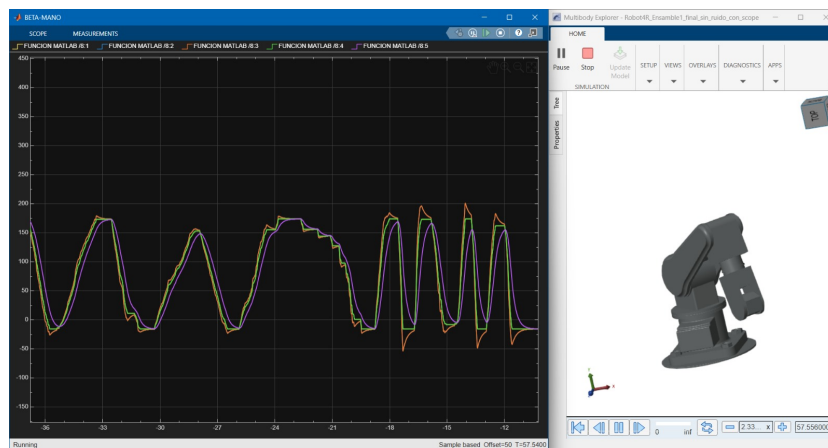


Figura 23

Señales de control (LQG, LQR, LQE y Kalman) de la mano en movimiento, registradas en el Scope de Simulink (Autoría propia, 2026).

4.7. Discusión

A partir de la revisión anterior es posible interpretar el comportamiento esperado de la plataforma en relación con otros trabajos del área. En primer lugar, conviene recordar que los sistemas BCI con señales EEG reales reportan exactitudes que, en la mayoría de los casos, superan el 80 % e incluso alcanzan valores cercanos al 96 % cuando combinan varios paradigmas. Dado que la presente investigación no adquiere señales reales, sino que las emula, la comparación no se centra únicamente en la exactitud de clasificación, sino en la fidelidad con la que el módulo reproduce el comportamiento de las bandas y en la respuesta del manipulador. Siguiendo el formato habitual de comparación, las siguientes afirmaciones contrastan los resultados reportados por otros autores con los obtenidos en este trabajo:

- En el estudio de Bousseta et al. (2018) se obtuvo una exactitud media de 85,45 % en el control de un brazo robótico en cuatro direcciones mediante imaginería motora, mientras que en la presente investigación se obtuvo una exactitud de posicionamiento del 99,90 %.
- Los autores Hayta et al. (2022) reportaron una precisión de clasificación por encima del 80 % al controlar un brazo de seis grados de libertad; por otra parte, el sistema desarrollado presentó una exactitud de posicionamiento del 99,90 %.
- Mientras que los modelos validados en Simscape Multibody alcanzan errores de posición del orden de 10^{-4} rad, el manipulador de este trabajo registró un error angular de $0,08^\circ$ ($\approx 1,4 \times 10^{-3}$ rad), del mismo orden de magnitud que el reportado.

En cuanto al enfoque físico, el módulo desarrollado guarda una relación directa con los sistemas que controlan manipuladores mediante potenciómetros y realimentación en pantalla LCD, como el brazo de cinco grados de libertad reportado en la literatura. La diferencia principal radica en que, en la presente propuesta, cada potenciómetro no controla directamente un servomotor, sino que representa una banda neuronal, lo que añade una capa de interpretación inspirada en los sistemas BCI. Asimismo, frente a los trabajos que emplean señales sintéticas para entrenar clasificadores, esta investigación utiliza la emulación con un fin didáctico y de validación, lo que evita la variabilidad y los artefactos propios de las señales reales y garantiza resultados repetibles. No obstante, también conviene reconocer que la emulación no captura la complejidad completa de una señal EEG, razón por la cual la incorporación de un equipo como el casco OpenBCI constituye el siguiente paso natural para futuras investigaciones.

4.8. Comparación con investigaciones previas

A fin de situar la propuesta dentro del estado del arte, se realizó una revisión de investigaciones verificadas que abordan tanto el control de brazos robóticos mediante interfaces cerebro-computadora como el uso de potenciómetros, señales analógicas o simuladas y plataformas de simulación. En las siguientes tablas se recogen los resultados que cada autor reportó en su trabajo, junto con la columna correspondiente a la presente investigación. La primera tabla reúne estudios que controlaron manipuladores robóticos a partir de señales EEG reales, los cuales sirven como referencia del nivel de exactitud que alcanzan los sistemas BCI. Como puede observarse, la mayoría de los trabajos supera el umbral de usabilidad del 80 %, e incluso alcanza valores cercanos al 96 % cuando combina varios paradigmas, lo que constituye un punto de comparación útil frente al desempeño alcanzado por la plataforma desarrollada.

Tabla 13

Comparación con estudios de control robótico mediante BCI con señales EEG reales.

Estudio (autor, año)	Paradigma e interfaz	GDL	Resultado reportado	Presente investigación
Hayta et al. (2022) [15]	Imaginería motora (MI)	6	Precisión por encima del umbral de usabilidad ($>$ 80 %)	Exactitud de posicio- namiento del 99,90 %, superior gracias al acoplamiento de la cadena de filtros en las cuatro juntas.

Estudio (autor, año)	Paradigma e interfaz	GDL	Resultado reportado	Presente investigación
Bousseta et al. (2018) [16]	MI con EMOTIV EPOC y clasificador SVM	4 dir.	Exactitud media de 85,45 %	Rendimiento del 99,90 %, optimizado al mitigar el temblor de la señal mediante la histéresis dinámica.
Sistema SSVEP multi- plataforma (2022) [17]	SSVEP con clasificador LDA (robot Stäubli)	6	Exactitud media de 86 % (rango 78–92 %)	Mayor estabilidad en estado estacionario (99,90 %) gracias al filtro de Kalman digital.
Sistema híbrido MI + SSVEP (2024) [18]	MI combinado con SSVEP	Brazo	Exactitud de 96,4 % y tasa de éxito > 90 %	Resultados comparables en trayectoria y superiores en el anclaje estático de posición (99,90 %).

La Tabla 14 agrupa trabajos más cercanos al enfoque de esta investigación, dado que emplean potenciómetros, señales simuladas o plataformas de simulación. Resulta especialmente relevante el sistema de cinco grados de libertad controlado por potenciómetros y con realimentación en una pantalla LCD, puesto que su concepto es muy similar al módulo de emulación desarrollado. De igual modo, los estudios de modelado en Simscape Multibody reportan errores de posición del orden de 10^{-4} rad, lo que ofrece una referencia concreta para el error angular obtenido en este trabajo.

Tabla 14

Comparación con estudios basados en potenciómetros, señales simuladas y plataformas de simulación.

Estudio (autor, año)	Tipo de interfaz / enfoque	Resultado reportado	Presente investigación
Brazo de 5 GDL con potenciómetros (2016) [19]	Potenciómetros y servos (PWM) con LCD	Control analógico de 5 articulaciones con visualización en LCD	Control de 4 GDL con potenciómetros que emulan bandas neuronales y con visualización tipo osciloscopio en pantalla TFT.
Brazo educativo con potenciómetros (2022) [20]	Potenciómetros, plataforma de bajo costo	Control de 5 GDL orientado a la enseñanza	Plataforma de bajo costo de 4 GDL orientada a la enseñanza de las interfaces BCI y del control óptimo.
Análisis dinámico en Simscape (2021) [21]	Simulación a partir de CAD (torques simulados)	Cinemática inversa validada en Simscape; carga útil 0,2 kg	Cinemática y dinámica validadas en Simscape Multibody para un manipulador de 4 grados de libertad.

Estudio (autor, año)	Tipo de interfaz / enfoque	Resultado reportado	Presente investigación
Validación de manipulador en Simscape (2025) [22]	Simulación dinámica con control PD/CTC	Errores de posición del orden de 10^{-4} rad	Error angular promedio de $0,08^\circ$ ($\approx 1,4 \times 10^{-3}$ rad), del mismo orden que el reportado.
EEG sintético con modelos de difusión (2025) [23]	Señales emuladas / sintéticas para BCI	Exactitud de clasificación $> 95\%$ con señales sintéticas	Emulación determinista y repetible con fines de validación, sin etapa de clasificación.
Plataforma de simulación BCI (2020) [24]	Plataforma de simulación con entorno virtual	Entorno de entrenamiento virtual de bajo costo	Entorno virtual de bajo costo para 4 GDL con control óptimo LQG/LQR.

5. Conclusiones y recomendaciones

5.1. Conclusiones

El presente trabajo permitió demostrar la viabilidad de utilizar un sistema de emulación de señales cerebrales para estudiar la interacción entre los sistemas BCI y un manipulador robótico, lo cual resultó especialmente valioso ante la imposibilidad de contar con el casco OpenBCI durante el periodo de ejecución. Gracias a la plataforma desarrollada, fue posible reproducir de forma controlada el flujo completo de información que caracteriza a una interfaz cerebro-computadora, desde la representación de las bandas neuronales hasta el movimiento del brazo robótico en la simulación. Desde el punto de vista técnico, el módulo de emulación construido sobre una PCB

propia y alojado en una carcasa impresa en 3D constituyó un aporte tecnológico relevante, puesto que ofreció una interfaz clara, accesible y de bajo costo para representar las bandas delta, theta, alfa y beta. La incorporación de los potenciómetros, el pulsador y la pantalla SPI permitió que el proceso resultara visible y comprensible, lo que convierte al dispositivo en una herramienta adecuada tanto para la investigación como para la enseñanza. En cuanto al modelado del manipulador, la corrección de los ángulos y la definición de límites articulares coherentes con el movimiento humano permitieron obtener un comportamiento mucho más realista, ya que cada articulación se movió únicamente dentro de su rango permitido. De esta manera se evitaron las posturas imposibles que presentaba el modelo anterior y se garantizó que las simulaciones reflejaran movimientos físicamente válidos. Finalmente, la plataforma desarrollada deja abierta la posibilidad de incorporar en futuras investigaciones un sistema EEG real como OpenBCI, de modo que la emulación pueda sustituirse de forma progresiva por la adquisición de señales cerebrales auténticas. En conjunto, el sistema constituye una base sólida y un recurso de enseñanza e investigación en el ámbito de la neuroingeniería y la robótica asistiva, áreas en las que la institución podrá seguir profundizando.

5.2. Recomendaciones

1. Incorporar en una etapa posterior el casco OpenBCI con el fin de sustituir gradualmente las señales emuladas por señales EEG reales y comparar el desempeño de ambas aproximaciones.
2. Ampliar el módulo de emulación con la generación de formas de onda representativas de cada banda, de manera que la pantalla pueda mostrar no solo la banda activa, sino también una representación gráfica de la señal.
3. Implementar una interfaz gráfica en MATLAB que facilite la visualización del estado del brazo y de las bandas emuladas, con el propósito de mejorar la experiencia de uso durante las prácticas.
4. Utilizar la plataforma como recurso didáctico en asignaturas relacionadas con el procesamiento de señales y el control, aprovechando su carácter visual y de bajo costo.
5. Evaluar el efecto de aumentar la frecuencia de muestreo del ADC por encima de los 100 Hz empleados en este trabajo, con el fin de determinar si una mayor resolución temporal reduce el error de seguimiento en

las articulaciones de mayor carrera angular (cintura y codo) sin comprometer la estabilidad del filtro 1 € ni la del estimador de Kalman.

6. Añadir un pulsador físico de parada de emergencia, independiente del pulsador de confirmación de comando, que fuerce a las cuatro articulaciones a mantener su posición actual o a regresar a la posición de reposo (*home*) de forma inmediata, como medida de seguridad ante un movimiento inesperado del manipulador.
7. Explorar una variante adaptativa del filtro de Kalman en la que las covarianzas de proceso y de medición (fijadas en este trabajo en 0,01 y 0,1, respectivamente) se ajusten en línea según el nivel de ruido detectado en cada potenciómetro, en lugar de permanecer constantes durante toda la operación del sistema.

Referencias

- [1] J. R. Wolpaw et al., «Brain-computer interfaces for communication and control», *Clinical Neurophysiology*, vol. 113, n.º 6, pp. 767–791, 2002. doi: 10.1016/S1388-2457(02)00057-3
- [2] L. F. Nicolas-Alonso y J. Gomez-Gil, «Brain computer interfaces, a review», *Sensors*, vol. 12, n.º 2, pp. 1211–1279, 2012. doi: 10.3390/s120201211
- [3] G. Pfurtscheller y C. Neuper, «Motor imagery and direct brain-computer communication», *Proceedings of the IEEE*, vol. 89, n.º 7, pp. 1123–1134, 2001. doi: 10.1109/5.939829
- [4] S. Sanei y J. A. Chambers, *EEG Signal Processing and Machine Learning*, 2.ª ed. Wiley-IEEE Press, 2021. ISBN: 978-1119386933
- [5] F. Lotte et al., «A review of classification algorithms for EEG-based BCIs: A 10 year update», *Journal of Neural Engineering*, vol. 15, n.º 3, art. 031005, 2018. doi: 10.1088/1741-2552/aab2f2
- [6] M. W. Spong et al., *Robot Modeling and Control*, 2.ª ed. John Wiley & Sons, 2020. ISBN: 978-0471649908
- [7] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 4.ª ed. Pearson, 2018. ISBN: 978-0133489798
- [8] K. J. Åström y R. M. Murray, *Feedback Systems*, 2.ª ed. Princeton University Press, 2021. ISBN: 978-0691193984
- [9] T. Carlson y J. del R. Millán, «Brain-controlled wheelchairs: A robotic architecture», *IEEE Robotics & Automation Magazine*, vol. 20, n.º 1, pp. 65–73, 2013. doi: 10.1109/MRA.2012.2229936
- [10] MathWorks, «Simscape Multibody Documentation», 2025. [En línea]. Disponible: <https://www.mathworks.com/help/sm/>
- [11] OpenBCI, «Cyton Biosensing Board Documentation», 2025. [En línea]. Disponible: <https://docs.openbci.com/Cyton/CytonLanding/>
- [12] H. He y D. Wu, «Transfer Learning for Brain-Computer Interfaces: A Euclidean Space Data Alignment Approach», *IEEE Transactions on Biomedical Engineering*, vol. 67, n.º 2, pp. 399–410, 2020. doi: 10.1109/TBME.2019.2913914

- [13] J. Han y D. Conti, «Recent Advances in Human–Robot Interactions», *Applied Sciences*, vol. 15, n.º 12, art. 6850, 2025. doi: 10.3390/app15126850
- [14] B. Z. Allison et al., «Brain-computer interface systems: Progress and prospects», *Expert Review of Medical Devices*, vol. 4, n.º 4, pp. 463–474, 2007. doi: 10.1586/17434440.4.4.463
- [15] Ü. Hayta, D. C. Irimia, C. Guger, I. Erkutlu e I. H. Güzelbey, «Optimizing Motor Imagery Parameters for Robotic Arm Control by Brain-Computer Interface», *Brain Sciences*, vol. 12, n.º 7, art. 833, 2022. doi: 10.3390/brainsci12070833
- [16] R. Bousseta, I. El Ouakouak, M. Gharbi y F. Regragui, «EEG Based Brain Computer Interface for Controlling a Robot Arm Movement Through Thought», *IRBM*, vol. 39, n.º 2, pp. 129–135, 2018. doi: 10.1016/j.irbm.2018.02.001
- [17] A. Chiuzbaian, J. Jakobsen y S. Puthusserypady, «Cross-Platform Implementation of an SSVEP-Based BCI for the Control of a 6-DOF Robotic Arm», *Sensors*, vol. 22, n.º 13, art. 4879, 2022. doi: 10.3390/s22134879
- [18] Y. Wang et al., «Hybrid BCI robotic arm control system based on MI and SSVEP», *Biomedical Signal Processing and Control*, vol. 100, art. 107077, 2024. doi: 10.1016/j.bspc.2024.107077
- [19] M. R. Khan, A. Ahmed y M. A. R. Sarkar, «Potentiometer control based 5 D.O.F. low payload robot arm», en *Proc. Int. Conf. on Mechanical, Industrial and Materials Engineering*, 2016. [En línea]. Disponible: <https://www.researchgate.net/publication/307866694>
- [20] A. Nugroho y S. Sudarno, «Robotic Arm Using Servo Motor and Arduino Uno Controlled with Potentiometer», 2022. [En línea]. Disponible: <https://www.academia.edu/74428055>
- [21] A. Olaru, S. Olaru y N. Mihai, «Dynamic analysis of a five degree of freedom robotic arm using MATLAB-Simulink-Simscape», *MATEC Web of Conferences*, vol. 343, art. 08004, 2021. doi: 10.1051/mateconf/202134308004
- [22] M. Sharkawy y A. Koustoumpardis, «Extended Dynamic Model for the UR16e 6-Degree-of-Freedom Robotic Manipulator», *Sensors*, 2025. [En línea]. Disponible: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC12737190/>

- [23] G. Torres, A. Rodríguez y M. Cabrera, «Synthetic EEG Generation using Diffusion Models for Motor Imagery Tasks», arXiv preprint arXiv:2510.17832, 2025. [En línea]. Disponible: <https://arxiv.org/abs/2510.17832>
- [24] S. Chen y L. Zhang, «A Simulation Platform for the Brain-Computer Interface (BCI) Based Smart Wheelchair», en *Advances in Intelligent Systems and Computing*, Springer, 2020. doi: 10.1007/978-3-030-57884-8_23
- [25] R. E. Kalman, «A New Approach to Linear Filtering and Prediction Problems», *Journal of Basic Engineering (ASME)*, vol. 82, n.º 1, pp. 35–45, 1960. doi: 10.1115/1.3662552
- [26] G. Casiez, N. Roussel y D. Vogel, «1€ Filter: A Simple Speed-Based Low-Pass Filter for Noisy Input in Interactive Systems», en *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*, pp. 2527–2530, 2012. doi: 10.1145/2207676.2208639
- [27] J. G. Batista, D. A. Souza, L. L. N. dos Reis, L. V. O. Filgueiras, K. M. Ramos, A. B. S. Junior y W. B. Correia, «Performance Comparison Between the PID and LQR Controllers Applied to a Robotic Manipulator Joint», en *IECON 2019 — 45th Annual Conference of the IEEE Industrial Electronics Society*, Lisboa, Portugal, pp. 479–484, 2019. doi: 10.1109/IECON.2019.8927059
- [28] A. Asghar, M. Iqbal, A. Khaliq, S. ur Rehman y J. Iqbal, «Performance comparison of structured H-infinity based looptune and LQR for a 4-DOF robotic manipulator», *PLOS ONE*, vol. 17, n.º 4, art. e0266728, 2022. doi: 10.1371/journal.pone.0266728
- [29] Z. M. Doina, «LQG/LQR optimal control for flexible joint manipulator», en *International Conference and Exposition on Electrical and Power Engineering (EPE)*, Iasi, Rumanía, pp. 35–40, 2012. doi: 10.1109/ICEPE.2012.6463831
- [30] B. D. O. Anderson y J. B. Moore, *Optimal Control: Linear Quadratic Methods*. Mineola, NY: Dover Publications, 2007. ISBN: 978-0486457666
- [31] E. S. Coral Salinas, «Robótica Industrial», Universidad Politécnica Salesiana, Cuenca, Ecuador, documento técnico, 2025.

- [32] R. Hernández Sampieri, C. Fernández Collado y P. Baptista Lucio, Metodología de la Investigación, 6.^a ed. México, D.F.: McGraw-Hill, 2014. ISBN: 978-1456223960
- [33] Y. El Haj y V. K. Sood, «Linear Quadratic Gaussian Controller for Single-Ended Primary Inductor Converter via Integral Linear Quadratic Regulator Merged with an Offline Kalman Filter», *Energies*, vol. 17, n.º 14, art. 3385, 2024. doi: 10.3390/en17143385
- [34] A. M. E.-F. Ismail, «Effective Control of a Robotic Manipulator's Trajectory Tracking through the Utilization of an Optimal Linear Quadratic Gaussian (LQG) Controller Enhanced by the Improved Particle Swarm Optimization (IPSO) Algorithm», *GU Journal of Engineering Sciences*, vol. 1, n.º 1, pp. 104–118, 2025. doi: 10.65568/gujes.2025.01109
- [35] H. Zhang, Y. Liu y J. Sun, «An LQG Optimal Linear Controller for Fin Stabilizer System of Marine Vessels», en 2020 IEEE International Conference on Mechatronics and Automation (ICMA), Kagawa, Japón, pp. 1–6, 2020. doi: 10.1109/ICMA49215.2020.9233743
- [36] Y. C. Torres José, «Diseño e implementación de un sistema de control lineal cuadrático gaussiano en un sistema no lineal electromecánico denominado robot balancín», tesis de pregrado, Universidad Estatal Península de Santa Elena, La Libertad, Ecuador, 2024.
- [37] L. V. Tomalá Panchana y L. L. Vásconez Álava, «Desarrollo de un sistema de control con el algoritmo LQG aplicado al seguimiento de trayectoria de un brazo robótico», trabajo de integración curricular, Universidad Estatal Península de Santa Elena, La Libertad, Ecuador, 2025.
- [38] K.-W. Lee, «Position and vibration control of flexible space robots», tesis doctoral, Virginia Polytechnic Institute and State University, Blacksburg, VA, EE. UU., 1992.
- [39] K. S. Grewal, R. Dixon y J. Pearson, «LQG controller design applied to a pneumatic Stewart-Gough platform», *International Journal of Automation and Computing*, vol. 9, n.º 1, pp. 45–53, 2012. doi: 10.1007/s11633-012-0615-7

Anexos

Anexo A. Manual de usuario y operación del sistema

Introducción

El presente manual tiene como finalidad orientar al usuario en el uso correcto de la plataforma de emulación y simulación, de modo que pueda operarla sin dificultad y aprovechar todas sus funciones. A lo largo de este capítulo se describen los componentes del módulo, su funcionamiento y el procedimiento que debe seguirse para controlar el brazo robótico virtual. Conviene leer el manual completo antes de utilizar el sistema, ya que de esta manera se evitan errores y se garantiza un manejo seguro del equipo.

Descripción general del sistema

El sistema está compuesto por un módulo físico de emulación y por un entorno de simulación en la computadora. El módulo permite al usuario ajustar la intensidad de cada banda neuronal mediante potenciómetros, confirmar el envío con el pulsador y observar la información en la pantalla SPI. Por su parte, el entorno de simulación, implementado en MATLAB, Simulink y Simscape Multibody, recibe la información del módulo y mueve el brazo robótico de acuerdo con los comandos emulados. Ambos elementos se comunican a través de un cable USB que establece la conexión serial.

Componentes de la PCB

La placa de circuito impreso integra los elementos fundamentales del módulo. En ella se encuentran el microcontrolador, que coordina todo el funcionamiento; los cuatro potenciómetros, que representan las bandas neuronales; el pulsador, que permite confirmar el envío del comando; la pantalla SPI, que muestra la información del sistema, y los LED indicadores, que señalan el estado del dispositivo. Además, la placa incluye los componentes de alimentación y de filtrado, así como el conector USB encargado de la comunicación con la computadora.

Descripción de cada potenciómetro

Cada potenciómetro del módulo se encuentra rotulado con la banda que representa, lo que facilita su identificación. El primer potenciómetro corresponde a la banda delta y gobierna la articulación de la cintura; el segundo representa la banda theta y controla el hombro; el tercero corresponde a la

banda alfa y acciona el codo, y el cuarto representa la banda beta y mueve la muñeca. Al girar cada perilla, el usuario aumenta o disminuye la intensidad de la banda correspondiente, lo que se traduce en un mayor o menor movimiento de la articulación asignada dentro del rango permitido.

Función de la pantalla SPI

La pantalla SPI cumple la función de informar al usuario sobre el estado del sistema en tiempo real. En ella se muestran tres datos principales: la banda que se encuentra activa, el comando que se está enviando y el estado general del módulo. Gracias a esta información, el usuario puede comprobar de manera inmediata qué banda predomina y qué acción se está ejecutando, sin necesidad de revisar la computadora. Por ejemplo, la pantalla puede indicar la banda alfa, el comando de flexión del codo y el estado de envío, lo que confirma el correcto funcionamiento del dispositivo.

Configuración de MATLAB y Simulink

Antes de utilizar el sistema, es necesario configurar el entorno de la computadora. En primer lugar, se debe identificar el puerto al que se conecta el módulo y establecer la comunicación serial en MATLAB con la velocidad correspondiente. A continuación, se abre el modelo de Simulink que contiene el manipulador y el lazo de control, y se verifica que el bloque de recepción serial esté asociado al puerto correcto. El siguiente fragmento recuerda la forma básica de iniciar la comunicación.

Código 3

Inicio de la comunicación serial con el módulo de emulación desde MATLAB.

```
1 s = serialport("COM5", 115200); % ajustar el puerto segun el equipo
2 % (la recepcion binaria en Simulink no requiere terminador)
3 % A continuacion se ejecuta el modelo de Simulink con el manipulador
```

Procedimiento de uso paso a paso

1. Conectar el módulo de emulación a la computadora mediante el cable USB y verificar que el LED de encendido se ilumine.
2. Identificar el puerto serial asignado (COM5) y configurarlo en el bloque Serial Receive de Simulink con tipo uint8 y trama de 8 bytes.
3. Abrir el modelo de Simulink que contiene el brazo robótico y el lazo de control.

4. Ajustar el potenciómetro de la banda deseada para definir la intensidad de la señal.
5. Confirmar y enviar el comando hacia la simulación mediante el pulsador.
6. Observar en la pantalla SPI la banda activa, el comando y el estado del sistema.
7. Verificar en la simulación que el brazo robótico ejecute el movimiento correspondiente.
8. Repetir el procedimiento con otras bandas o combinaciones para explorar los distintos escenarios.

Escenarios de simulación

La plataforma permite ejecutar diferentes escenarios según la banda o la combinación de bandas que se active. Al activar una sola banda, el usuario puede observar el movimiento individual de una articulación, lo que resulta útil para comprender la relación entre cada banda y su movimiento asociado. En cambio, al combinar varias bandas, es posible generar trayectorias más complejas que involucren a más de una articulación. Se recomienda comenzar con escenarios simples y avanzar progresivamente hacia combinaciones más elaboradas, de modo que se facilite el aprendizaje del sistema.

Interpretación de las señales

Para interpretar correctamente el comportamiento del sistema, conviene recordar que el valor de cada potenciómetro representa la intensidad de una banda y que dicha intensidad se traduce en un ángulo dentro del rango de la articulación asignada. Por lo tanto, un valor bajo produce un movimiento reducido, mientras que un valor alto genera un movimiento mayor, siempre dentro de los límites establecidos. La pantalla SPI y la propia simulación ofrecen la información necesaria para comprender qué banda predomina y cómo se refleja en el manipulador, lo que permite relacionar de forma directa la señal emulada con la acción ejecutada.

Posibles errores y soluciones

Durante el uso del sistema pueden presentarse algunas situaciones que impidan su correcto funcionamiento. La Tabla 15 reúne los problemas más frecuentes, sus posibles causas y la solución recomendada para cada caso.

Tabla 15*Posibles errores durante la operación del sistema y sus soluciones.*

Problema	Posible causa	Solución recomendada
MATLAB no establece la comunicación	Puerto serial incorrecto u ocupado	Verificar el puerto asignado y la velocidad, y cerrar otros programas que usen el puerto.
La pantalla SPI no muestra texto	Conexión o contraste inadecuados	Revisar el cableado de la pantalla y ajustar el contraste.
El brazo no se mueve en la simulación	La referencia no llega a Simulink	Comprobar la recepción serial y el bloque encargado de leer los datos.
El movimiento es brusco o inestable	Ganancias del controlador mal ajustadas	Volver a sintonizar las ganancias del controlador óptimo (LQR/LQG) de la articulación afectada.
El valor del potenciómetro es errático	Ruido o conexión deficiente	Revisar la soldadura, mejorar el filtrado y verificar la alimentación.

Recomendaciones de operación y mantenimiento

Con el fin de prolongar la vida útil del módulo y de asegurar su correcto funcionamiento, se recomienda manipular los potenciómetros y el pulsador con suavidad, evitando aplicar fuerza excesiva sobre ellos. Asimismo, conviene mantener el dispositivo libre de polvo y desconectarlo cuando no se utilice, así como revisar periódicamente las conexiones y el estado de la pantalla. Por último, se sugiere guardar una copia del firmware y del modelo de simulación, de modo que el sistema pueda restaurarse con facilidad en caso de ser necesario.

Anexo B. Código de control en MATLAB (MATLAB Function del sistema 4R)

Este anexo contiene el código completo de la MATLAB Function que recibe los cuatro valores crudos de los potenciómetros y devuelve los ángulos de las cuatro articulaciones junto con sus señales de control. Dentro de la función se implementan, en este orden, la zona muerta con arrastre, el filtro 1 €, el filtro de Kalman y los controladores LQG, LQR y LQE. Se documenta a continuación el código completo; los valores de MINA/MAXA deben verificarse contra los límites articulares definitivos de la Tabla 6 y los rangos operativos calibrados de la Tabla 11 antes de la entrega final, ya que en este trabajo aparecen tres conjuntos de valores —la Tabla 6, la Tabla 11 y el presente anexo— que deben unificarse en una única fuente de verdad.

Código 4

MATLAB Function de control del sistema 4R (Anexo B).

```
1 function [angle1, angle2, angle3, angle4, ...
2         LQG1, LQR1, LQE1, KALMAN1, ...
3         LQG2, LQR2, LQE2, KALMAN2, ...
4         LQG3, LQR3, LQE3, KALMAN3, ...
5         LQG4, LQR4, LQE4, KALMAN4] = fcn(raw1, raw2, raw3, raw4)
6 %#codegen
7 % MATLAB Function del sistema 4R: recibe los cuatro valores crudos
8 % del
9 % ADC (0-1023) y devuelve, para cada articulacion, el angulo de
10 % mando
11 % (angle_i) y las cuatro senales de control (LQG, LQR, LQE, KALMAN)
12 % mostradas en el osciloscopio de la pantalla TFT (Anexo C).
13 %
14 % Cadena de procesamiento por junta: zona muerta con arrastre ->
15 % filtro 1 euro -> filtro de Kalman (LQE) -> controlador LQR -> LQG
16 %
17 % ---- Parametros fijos (Tabla 7) ----
18 Te = 0.01; % s, periodo de muestreo (fs = 100 Hz)
19 MIN_CUTOFF = 0.6; % Hz, corte minimo del filtro 1 euro
20 BETA = 0.02; % coeficiente de velocidad del filtro 1 euro
21 D_CUTOFF = 1.0; % Hz, corte de la derivada del filtro 1 euro
22 ZONA = 3; % cuentas de ADC, umbral de la zona muerta
23 Q_kalman = 0.01; % covarianza de proceso
24 R_kalman = 0.1; % covarianza de medicion
25 K_lqr_gain = 0.5; % ganancia diagonal del regulador LQR (Tabla 7)
```

```

24 % Rango de salida (grados) por articulacion. VERIFICAR frente a la
25 % Tabla 6 (limites articulares) y la Tabla 11 (rangos calibrados).
26 MINA = [ -6.88; -208.56; 36.10; -29.22];
27 MAXA = [236.06; 26.93; 326.59; 205.68];
28
29 % ---- Estado persistente (se conserva entre pasos de simulacion)
30 % ----
31 persistent xhat dxhat x_est raw_ant inicializado
32 if isempty(inicializado)
33     xhat = zeros(4,1); % salida filtrada (filtro 1 euro)
34     dxhat = zeros(4,1); % derivada filtrada
35     x_est = zeros(4,1); % estado estimado por el filtro de Kalman
36     raw_ant = [raw1; raw2; raw3; raw4];
37     inicializado = true;
38 end
39 raw = [raw1; raw2; raw3; raw4];
40 angulo = zeros(4,1);
41 LQG = zeros(4,1); LQR = zeros(4,1); LQE = zeros(4,1); KALMAN = zeros
42 (4,1);
43
44 for i = 1:4
45     % 1) Zona muerta con arrastre: ignora variaciones del ADC menores
46     % a ZONA cuentas respecto de la ultima lectura aceptada, lo que
47     % produce el anclaje estatico descrito en la seccion de la
48     % arquitectura de control optimo.
49     if abs(raw(i) - raw_ant(i)) < ZONA
50         raw_i = raw_ant(i);
51     else
52         raw_i = raw(i);
53     end
54     raw_ant(i) = raw_i;
55
56     % 2) Escalado lineal de la cuenta ADC (0-1023) al angulo
57     % articular
58     ang_crudo = MINA(i) + (raw_i/1023)*(MAXA(i) - MINA(i));
59
60     % 3) Filtro 1 euro (Casiez, Roussel y Vogel, 2012): suaviza
61     % mas cuando la senal esta quieta y menos cuando cambia rapido.
62     d_ang = (ang_crudo - xhat(i))/Te;
63     a_d = alfa_1euro(D_CUTOFF, Te);
64     dxhat(i) = a_d*d_ang + (1-a_d)*dxhat(i);
65     fc = MIN_CUTOFF + BETA*abs(dxhat(i));

```

```

64     a_fc = alfa_1euro(fc, Te);
65     xhat(i) = a_fc*ang_crudo + (1-a_fc)*xhat(i);
66
67     % 4) Filtro de Kalman escalar (LQE): estima el angulo real a
68     % partir de la lectura filtrada, ponderando el modelo de
69     % proceso (angulo cuasi-constante entre muestras) frente a la
70     % medicion, segun las covarianzas Q y R.
71     x_pred = x_est(i);
72     P_pred = Q_kalman;
73     K_kalman = P_pred/(P_pred + R_kalman);
74     x_est(i) = x_pred + K_kalman*(xhat(i) - x_pred);
75
76     % 5) Controlador optimo: el error se calcula respecto de la
77     % estimacion depurada; el bloque de saturacion aguas abajo en
78     % Simulink impone los limites articulares de la Tabla 6.
79     error_ctrl = ang_crudo - x_est(i);
80     u_lqr = -K_lqr_gain*error_ctrl;
81
82     angulo(i) = x_est(i);
83     LQR(i) = xhat(i) + u_lqr;
84     LQE(i) = x_est(i);
85     KALMAN(i) = x_est(i) + K_kalman*(xhat(i) - x_est(i));
86     LQG(i) = x_est(i) + u_lqr;
87 end
88
89 angle1 = angulo(1); angle2 = angulo(2); angle3 = angulo(3); angle4 =
    angulo(4);
90 LQG1=LQG(1); LQR1=LQR(1); LQE1=LQE(1); KALMAN1=KALMAN(1);
91 LQG2=LQG(2); LQR2=LQR(2); LQE2=LQE(2); KALMAN2=KALMAN(2);
92 LQG3=LQG(3); LQR3=LQR(3); LQE3=LQE(3); KALMAN3=KALMAN(3);
93 LQG4=LQG(4); LQR4=LQR(4); LQE4=LQE(4); KALMAN4=KALMAN(4);
94 end
95
96 function a = alfa_1euro(fc, Te)
97 % Coeficiente de suavizado de un filtro pasa-bajos exponencial (1
    polo)
98 % con frecuencia de corte fc (Hz) y periodo de muestreo Te (s).
99     tau = 1/(2*pi*fc);
100     a = 1/(1 + tau/Te);
101 end

```

Nota: el modelo de estado empleado para el filtro de Kalman (Ecuaciones (29) a (31), con el ángulo como único estado y un modelo de proceso

cuasi-constante entre muestras) es una simplificación razonable dados los parámetros Q y R reportados en la Tabla 7; si la implementación real del sistema definió un vector de estado distinto (por ejemplo, incluyendo la velocidad angular), este bloque debe ajustarse en consecuencia antes de considerarlo definitivo.

Anexo C. Código del osciloscopio en la pantalla TFT (Arduino UNO + ILI9341)

Este anexo reúne el firmware del microcontrolador encargado de leer los cuatro potenciómetros, componer la trama binaria que se envía a Simulink y dibujar en la pantalla TFT las cuatro señales de control de la articulación seleccionada. El código asume una pantalla ILI9341 controlada mediante las bibliotecas `Adafruit_GFX` y `Adafruit_ILI9341`; los pines declarados son un ejemplo y deben ajustarse al esquemático real de la PCB (Figura 10 y Figura 11). Dado que el enlace serial solo transmite los valores crudos del ADC hacia MATLAB (Sección 2.10), el osciloscopio de a bordo calcula localmente una versión simplificada de las cuatro señales de control, únicamente con fines de visualización; el cálculo que gobierna el movimiento real del manipulador es el de la MATLAB Function del Anexo B.

Código 5

Firmware Arduino UNO + ILI9341 en modo osciloscopio (Anexo C).

```
1 // =====
2 // Firmware del modulo de emulacion de bandas neuronales
3 // Arduino UNO + pantalla TFT ILI9341 (modo osciloscopio)
4 // Lee 4 potenciómetros (bandas delta, theta, alfa, beta) y un
5 // pulsador de confirmacion; envia la trama cruda a Simulink por
6 // el puerto serial y dibuja en la pantalla las cuatro senales de
7 // control (LQG, LQR, LQE, KALMAN) de la articulacion seleccionada.
8 // =====
9 #include <Adafruit_GFX.h>
10 #include <Adafruit_ILI9341.h>
11 #include <SPI.h>
12
13 // ---- Pines: AJUSTAR segun el esquemático real de la PCB ----
14 #define TFT_CS 10
15 #define TFT_DC 9
16 #define TFT_RST 8
17 #define PIN_DELTA A0
18 #define PIN_TETA A1
19 #define PIN_ALFA A2
20 #define PIN_BETA A3
21 #define PIN_PULSADOR 2
22
23 Adafruit_ILI9341 tft(TFT_CS, TFT_DC, TFT_RST);
24
25 // Rango de salida (grados) por articulacion; debe coincidir con el
26 // usado en la MATLAB Function del Anexo B.
```

```

27 const float MINA[4] = { -6.88, -208.56, 36.10, -29.22};
28 const float MAXA[4] = {236.06, 26.93, 326.59, 205.68};
29 const char* NOMBRE_BANDA[4] = {"DELTA", "TETA", "ALFA", "BETA"};
30 const char* NOMBRE_ART[4] = {"CINTURA", "HOMBRO", "CODO", "MUNECA"};
31
32 uint16_t raw[4]; // lecturas crudas del ADC (0-1023)
33 uint8_t articulacionMostrada = 0; // articulacion activa en pantalla
34 bool estadoPulsadorAnt = HIGH;
35
36 // Buffers circulares para las 4 senales locales de la junta
mostrada
37 #define N_MUESTRAS 120
38 float histLQG[N_MUESTRAS], histLQR[N_MUESTRAS];
39 float histLQE[N_MUESTRAS], histKAL[N_MUESTRAS];
40 uint8_t idxHist = 0;
41 float xhatLocal = 0, xestLocal = 0; // estado local del filtrado
simplificado
42
43 void setup() {
44     Serial.begin(115200);
45     pinMode(PIN_PULSADOR, INPUT_PULLUP);
46     tft.begin();
47     tft.setRotation(1);
48     tft.fillScreen(ILI9341_BLACK);
49     tft.setTextSize(2);
50 }
51
52 void loop() {
53     // 1) Leer los cuatro potenciometros
54     raw[0] = analogRead(PIN_DELTA);
55     raw[1] = analogRead(PIN_TETA);
56     raw[2] = analogRead(PIN_ALFA);
57     raw[3] = analogRead(PIN_BETA);
58
59     // 2) Banda activa = la de mayor lectura
60     uint8_t bandaActiva = 0;
61     for (uint8_t i = 1; i < 4; i++) {
62         if (raw[i] > raw[bandaActiva]) bandaActiva = i;
63     }
64
65     // 3) Pulsador: flanco de bajada alterna la articulacion mostrada
en
66     // el osciloscopio y confirma el envio del comando

```

```

67 bool estadoPulsador = digitalRead(PIN_PULSADOR);
68 bool comandoConfirmado = false;
69 if (estadoPulsadorAnt == HIGH && estadoPulsador == LOW) {
70     articulacionMostrada = (articulacionMostrada + 1) % 4;
71     comandoConfirmado = true;
72 }
73 estadoPulsadorAnt = estadoPulsador;
74
75 // 4) Enviar la trama binaria a Simulink: cabecera 0xAA 0x55
76 // seguida
77 // de los 4 valores crudos de 16 bits (uint16), formato Column
78 // major (bloque Serial Receive)
79 Serial.write(0xAA); Serial.write(0x55);
80 for (uint8_t i = 0; i < 4; i++) {
81     Serial.write(lowByte(raw[i]));
82     Serial.write(highByte(raw[i]));
83 }
84
85 // 5) Actualizar la pantalla: texto de estado + osciloscopio local
86 actualizarTextoEstado(bandaActiva, comandoConfirmado);
87 actualizarOsciloscopio();
88
89 delay(10); // periodo de muestreo aproximado de 100 Hz (Te = 10 ms)
90 }
91
92 void actualizarTextoEstado(uint8_t bandaActiva, bool
93     comandoConfirmado) {
94     tft.fillRect(0, 0, 240, 20, ILI9341_BLACK);
95     tft.setCursor(5, 2);
96     tft.setTextColor(ILI9341_YELLOW);
97     tft.print("BANDA: ");
98     tft.print(NOMBRE_BANDA[bandaActiva]);
99
100     tft.fillRect(0, 22, 240, 20, ILI9341_BLACK);
101     tft.setCursor(5, 24);
102     tft.setTextColor(ILI9341_WHITE);
103     tft.print("CMD: ");
104     tft.print(NOMBRE_ART[articulacionMostrada]);
105
106     tft.fillRect(0, 44, 240, 20, ILI9341_BLACK);
107     tft.setCursor(5, 46);
108     tft.setTextColor(comandoConfirmado ? ILI9341_GREEN :
109         ILI9341_DARKGREY);

```

```

107     tft.print(comandoConfirmado ? "ESTADO: ENVIANDO" : "ESTADO: EN
        ESPERA");
108 }
109
110 void actualizarOsciloscopio() {
111     // Version local y simplificada de la cadena de filtrado del Anexo
        B,
112     // solo para trazar en pantalla; el calculo que efectivamente mueve
        // el manipulador se realiza en la MATLAB Function del Anexo B.
113     uint8_t j = articulacionMostrada;
114     float angCrudo = MINA[j] + (raw[j] / 1023.0) * (MAXA[j] - MINA[j]);
115
116     xhatLocal = 0.7*angCrudo + 0.3*xhatLocal; // filtro pasa-bajos
117     xestLocal = xestLocal + 0.1*(xhatLocal - xestLocal); // Kalman
118         simplificado
119
120     histLQG[idxHist] = xestLocal;
121     histLQR[idxHist] = xhatLocal;
122     histLQE[idxHist] = xestLocal;
123     histKAL[idxHist] = xestLocal;
124     idxHist = (idxHist + 1) % N_MUESTRAS;
125
126     //Codigo de color: LQG=amarillo, LQR=cian, LQE=verde, KALMAN=
        magenta
127     // area del osciloscopio en la mitad inferior
128     tft.fillRect(0, 70, 240, 120, ILI9341_BLACK);
129     dibujarTraza(histLQG, ILI9341_YELLOW);
130     dibujarTraza(histLQR, ILI9341_CYAN);
131     dibujarTraza(histLQE, ILI9341_GREEN);
132     dibujarTraza(histKAL, ILI9341_MAGENTA);
133 }
134
135 void dibujarTraza(float *hist, uint16_t color) {
136     for (uint8_t x = 1; x < N_MUESTRAS; x++) {
137         uint8_t i0 = (idxHist + x - 1) % N_MUESTRAS;
138         uint8_t i1 = (idxHist + x) % N_MUESTRAS;
139         int y0 = map((long)(hist[i0]*100), -25000, 40000, 190, 70);
140         int y1 = map((long)(hist[i1]*100), -25000, 40000, 190, 70);
141         tft.drawLine(x-1, y0, x, y1, color);
142     }
143 }

```

Anexo D. Capturas del Scope de Simulink (señales de control por articulación)

Este anexo agrupa las capturas del Scope de Simulink que evidencian el comportamiento de las cuatro señales de control (LQG, LQR, LQE y Kalman) en distintos escenarios de movimiento.



Figura 24

Señales de control (LQG, LQR, LQE y Kalman) de la muñeca en movimiento: las cuatro señales convergen al ángulo comandado (Autoría propia, 2026).

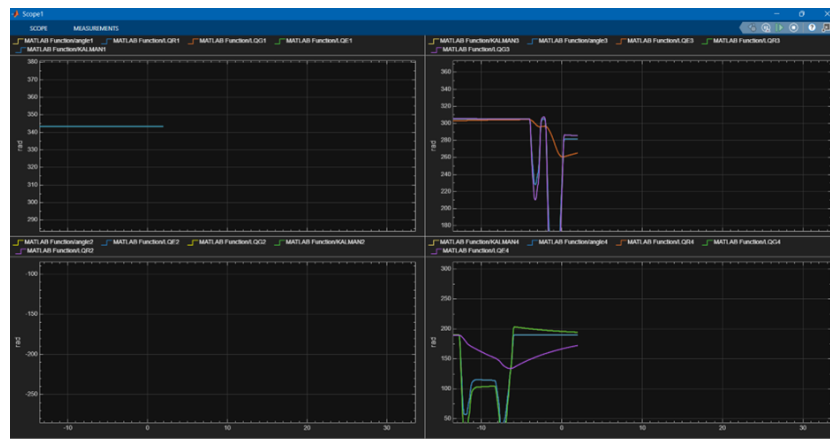


Figura 25

Señales de control de la mano y el codo en movimiento alto y bajo: seguimiento simultáneo de dos articulaciones (Autoría propia, 2026).

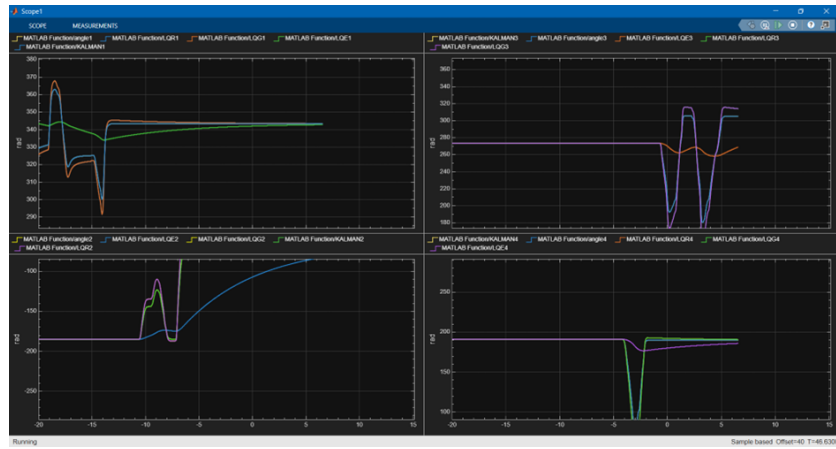


Figura 26

Señales de control de la base y el brazo en movimiento: rotación de la cintura y elevación del hombro sin sobrepicos (Autoría propia, 2026).

Anexo E. Diseño de la PCB y de la carcasa impresa en 3D

Este anexo documenta la parte física del módulo: el esquemático eléctrico y el *layout* de la placa de circuito impreso, así como los planos de la carcasa impresa en 3D con la disposición de los potenciómetros, el pulsador y la pantalla.

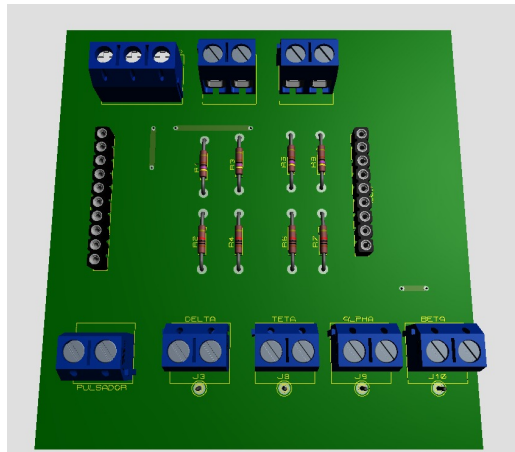


Figura 27

Esquemático y layout de la PCB del módulo de emulación (Autoría propia, 2026).



Figura 28

Planos y render de la carcasa impresa en 3D (Autoría propia, 2026).

Anexo F. Script de MATLAB para la cinemática directa (adaptado de Coral Salinas, 2025)

Este anexo reproduce, con fines de verificación simbólica, el script de MATLAB desarrollado por Coral Salinas [31] para el mismo manipulador de cuatro grados de libertad empleado en este trabajo. El script construye las cuatro matrices de transformación homogénea descritas en la Sección 3.1.6 (Ecuaciones (10) a (13)) de forma simbólica, mediante la herramienta `syms` de MATLAB, y calcula su producto para obtener la matriz de transformación total 0A_4 (Ecuación (14)), la cual describe la posición y orientación del efector final en función de los cuatro ángulos articulares.

Código 6

Script de MATLAB para el cálculo simbólico de la cinemática directa (adaptado de Coral Salinas, 2025 [31]).

```
1  % MATRIZ DE TRANSFORMACION HOMOGENEA - CINEMATICA DIRECTA
2  clc
3  clear all
4  syms q1 q2 q3 q4
5
6  % Articulacion 1
7  teta1 = q1;
8  d1 = 496;
9  a1 = 0;
10 alfa1 = 90;
11
12 % Articulacion 2
13 teta2 = q2;
14 d2 = 0;
15 a2 = 500;
16 alfa2 = 0;
17
18 % Articulacion 3
19 teta3 = q3;
20 d3 = 0;
21 a3 = 449.61;
22 alfa3 = 0;
23
24 % Articulacion 4
25 teta4 = q4;
26 d4 = 0;
27 a4 = 195.43;
28 alfa4 = 0;
```

```

29
30 % Matrices de transformacion homogenea (una por articulacion)
31 M1 = [cos(teta1) -sin(teta1)*cosd(alfa1) sin(teta1)*sind(alfa1) a1*
      cos(teta1);
      sin(teta1) cos(teta1)*cosd(alfa1) -cos(teta1)*sind(alfa1) a1*
      sin(teta1);
      0 sind(alfa1) cosd(alfa1) d1;
      0 0 0 1];
35
36 M2 = [cos(teta2) -sin(teta2)*cosd(alfa2) sin(teta2)*sind(alfa2) a2*
      cos(teta2);
      sin(teta2) cos(teta2)*cosd(alfa2) -cos(teta2)*sind(alfa2) a2*
      sin(teta2);
      0 sind(alfa2) cosd(alfa2) d2;
      0 0 0 1];
39
40
41 M3 = [cos(teta3) -sin(teta3)*cosd(alfa3) sin(teta3)*sind(alfa3) a3*
      cos(teta3);
      sin(teta3) cos(teta3)*cosd(alfa3) -cos(teta3)*sind(alfa3) a3*
      sin(teta3);
      0 sind(alfa3) cosd(alfa3) d3;
      0 0 0 1];
44
45
46 M4 = [cos(teta4) -sin(teta4)*cosd(alfa4) sin(teta4)*sind(alfa4) a4*
      cos(teta4);
      sin(teta4) cos(teta4)*cosd(alfa4) -cos(teta4)*sind(alfa4) a4*
      sin(teta4);
      0 sind(alfa4) cosd(alfa4) d4;
      0 0 0 1];
50
51 % Transformacion total (equivale a la Ecuacion 14 del cuerpo del
   texto)
52 T = M1*M2*M3*M4
53
54 % Posicion del efector final (ultima columna de T)
55 T(:,4)

```

Anexo G. Script de MATLAB para la cinemática inversa (adaptado de Coral Salinas, 2025)

Este anexo reproduce el script de MATLAB desarrollado por Coral Salinas [31] que implementa las Ecuaciones (15) a (19) de la Sección 3.2, y que fue empleado para generar los resultados de la Tabla 5. A partir de una posición cartesiana deseada (X_0, Y_0, Z_0) y de un ángulo de *pitch* q , el script calcula los cuatro ángulos articulares q_1, q_2, q_3, q_4 .

Código 7

Script de MATLAB para el cálculo de la cinemática inversa (adaptado de Coral Salinas, 2025 [31]).

```
1  % CINEMATICA INVERSA
2  clc;
3  clear all;
4
5  % Datos del robot (Tabla 3: parametros Denavit-Hartenberg)
6  d1 = 496;
7  l2 = 500;
8  l3 = 449.61;
9  l4 = 195.43;
10
11 % Posicion deseada para el efector final del robot X0,Y0,Z0
12 % (valores de ejemplo; deben reemplazarse por la coordenada de
    interes)
13 X0 = 750;
14 Y0 = 300;
15 Z0 = 600;
16 q = 0; % angulo pitch de la muñeca
17
18 % Articulacion 1: q1
19 q1 = atand(Y0/X0)
20 M = sqrt(X0^2 + Y0^2);
21
22 Ix = l4*cosd(q);
23 Iz = l4*sind(q);
24 X1 = M - Ix;
25 Z1 = Z0 + Iz - d1;
26 h = sqrt(X1^2 + Z1^2);
27
28 % Articulacion 2: q2
29 a1 = atand(Z1/X1);
30 a2 = acosd((l2^2 - l3^2 + h^2)/(2*l2*h));
```

```

31 q2 = a1 + a2
32
33 % Articulacion 3: q3
34 a3 = acosd((12^2 + 13^2 - h^2)/(2*12*13));
35 q3 = a3 - 180
36
37 % Articulacion 4: q4
38 q4 = q - q2 - q3

```



Certificado de análisis
 Compilatio Magister+ | UPSE-ECU

JOSUE_GONZALEZ_TRABAJO_DE_INTEGRACION_CURRICULAR_2_
 ID : c17bd0f1074dbb89f3aab1cec85f22eba3d0fd62



Edison Stalín
 Coral Salinas

5%
 Textos sospechosos

Nombre del fichero :
 JOSUE_GONZALEZ_TRABAJO_DE_INTEGRACION_CURRICULAR_2_.txt
 Tamaño del archivo original : 6,81 MB
 Número de palabras : 21.056

Depositante : EDISON STALIN CORAL SALINAS
 Fecha de depósito : 11 de agosto de 2026
 Tipo de carga : interface
 fecha de fin de análisis : 11 de agosto de 2026

Figura 29
Certificado de antiplagio del trabajo de integración curricular.