



Identificación de la Intencionalidad de Movimientos del Miembro Superior mediante el uso de Sensores de Fuerza y Sensores Inerciales

Jaisson Hernando Forero Guerrero

Universidad Antonio Nariño
Facultad de Ingeniería Electrónica, Biomédica y Mecatrónica
Programa de Ingeniería Electronica
Bogotá, Colombia
2020

Identificación de la Intencionalidad de Movimientos del Miembro Superior mediante el uso de Sensores de Fuerza y Sensores Inerciales

Jaisson Hernando Forero Guerrero

Trabajo de grado presentado como requisito para optar al título de:
Ingeniero Electrónico

Director:
Ing. Leonardo Torres Londoño

Semillero de investigación: CASEIB
Universidad Antonio Nariño
Facultad de Ingeniería Electrónica, Biomédica y Mecatrónica
Programa de Ingeniería Electrónica
Bogotá, Colombia

2020

III

”La vida es como andar en bicicleta. Para mantener el equilibrio , tienes que seguir moviéndote” - ALbert Einstein

Dedico el presente trabajo a mi hijo Martín Alejandro, fuente inagotable que alimenta mis deseos para siempre alcanzar mis metas y objetivos, a Erika el amor de mi vida, mi esposa que se mantuvo persistente en su deseo de apoyarme para el desarrollo y culminación de este proyecto y por ultimo a mi madre y padre que inculcaron en mi los mas importantes valores y cualidades que han permitido que cumpla todos los objetivos que me he propuesto hasta el día de hoy.han estado presente en todos los momentos de mi vida suministrando su incondicional apoyo.

Agradecimientos

Agradezco de manera especial al Ingeniero Leonardo Torres Londoño; Aduedo todo el esfuerzo, tiempo y dedicación que enfoco en el inicio, ejecución y culminación del presente proyecto. Agradezco a mi hogar, familia y amigos por tener la disposición y apoyo en el transcurso del desarrollo del trabajo culminado.

Resumen

En el ámbito de la rehabilitación de personas con discapacidad por daño o falta de alguno de sus miembros, el uso de prótesis robóticas es de gran importancia. En la actualidad estas prótesis presentan un alto desarrollo tecnológico a nivel su hardware y software, sin embargo, su funcionamiento depende directamente de la interfaz hombre maquina (HMI), que es la encargada de registrar e identificar los datos generados ante la intencionalidad del movimiento del miembro; Un registro apropiado permitirá un óptimo control de la prótesis.

Se implementó un sistema que permite a la identificación de la intencionalidad del movimiento del miembro superior usando técnicas alternas como el uso sensores de fuerza y sensores de movimiento inercial. Como primer paso se fabricó un brazalete cuya función fue registrar los datos pertinentes a cambios de presión y posición del brazo.

Posteriormente se efectuó el protocolo de registro de información donde se establecieron parámetros para lograr la comunicación entre el brazalete microcontrolador, microcontrolador computador. Por medio de la herramienta de Matlab se implementaron algoritmos para captar las señales del brazalete, el cual fue usado en 10 sujetos de prueba cada uno de los voluntarios mantuvieron una posición del brazo fija y un agarre en el momento del registro. Se concreto el registro de ocho combinaciones: dos posiciones y cuatro tipos de agarre, estableciendo un procedimiento estándar para la ubicación del brazalete, y la toma de registros.

Todos los datos fueron procesados y almacenados en el computador, para luego ser utilizados en la construcción de conjuntos de entrenamiento y conjuntos de validación. Se realizaron múltiples pruebas de validación para dos

algoritmos de aprendizaje supervisado: Regresión logística y máquinas de soporte vectorial variando los parámetros de medida como: sujeto, número de ensayos, número de posiciones, inclusión y exclusión de los sensores de movimiento inercial. Los resultados de las pruebas exponen un alto índice de exactitud en la predicción de la posición y agarre ante la intencionalidad de movimiento del sujeto de prueba en modo Off Line. Por otro lado, se desarrolló una aplicación en Matlab que permite evaluar de manera gráfica el comportamiento en modo On Line. Sin embargo, la exactitud en la predicción disminuyó hasta en un cincuenta por ciento.

Abstract

In the field of rehabilitation of people with disabilities due to damage or lack of any of their members, the use of robotic prostheses is of great importance. At present, these prostheses present a high technological development at the level of their hardware and software, however, their operation depends directly on the human machine interface (HMI), which is in charge of recording and identifying the data generated before the intention of the movement of the member; Proper registration will allow optimal control of the prosthesis.

A system was implemented that allows the identification of the intentionality of the upper limb movement using alternative techniques such as the use of force sensors and inertial motion sensors. As a first step, a bracelet was manufactured whose function was to record data pertinent to changes in pressure and position of the arm.

Subsequently, the information recording protocol was carried out where parameters were established to achieve communication between the microcontroller bracelet, computer microcontroller. Through the Matlab tool, algorithms were implemented to capture the signals from the bracelet, which

was used in 10 test subjects, each one of the volunteers maintained a fixed arm position and a grip at the time of registration. The registration of eight combinations was specified: two positions and four types of grip, establishing a standard procedure for the location of the bracelet, and the taking of records.

All the data were processed and stored in the computer, to later be used in the construction of training sets and validation sets. Multiple validation tests were performed for two supervised learning algorithms: logistic regression and vector support machines, varying the measurement parameters such as: subject, number of trials, number of positions, inclusion and exclusion of inertial motion sensors. The results of the tests show a high index of accuracy in the prediction of the position and grip before the intention of movement of the test subject in Off Line mode. On the other hand, an application was developed in Matlab that allows to graphically evaluate the behavior in On Line mode. However, the accuracy of the prediction decreased by as much as fifty percent.

Keywords

Aprendizaje Automático, Aprendizaje supervisado, Inteligencia Artificial, Identificación intencionalidad del movimiento, Reconocimiento de Patrones.

Tabla de contenido

Agradecimientos	v
Resumen	vi
Índice de figuras	xiv
1. Introducción	1
1.1. Introducción	1
1.2. Estado del arte	4
1.3. Planteamiento del Problema	10
1.4. Objetivos	11
1.4.1. Objetivo General	11
1.4.2. Objetivos específicos	11
2. Marco Teórico	12
2.1. Técnicas Actuales de Identificación	12
2.1.1. Empleando electromiografía	12
2.1.2. Empleando electroencefalografía	13
2.2. Sensores	13
2.2.1. Sensores de Fuerza	13
2.2.2. Sensores de Movimiento Inercial	14
2.3. Aprendizaje de máquinas o aprendizaje automático	16
2.3.1. Aprendizaje Supervisado	17
2.3.2. Aprendizaje No Supervisado	17
2.4. Regresión Lineal	18
2.4.1. Una variable	18

Tabla de contenido

2.4.2. Multi variable	19
2.4.3. Regresión Logística	19
2.4.4. Máquinas de Soporte Vectorial	21
3. Descripción general de la Metodología	22
4. Fabricación del Brazaletes	25
4.1. Sensores	26
4.1.1. Materiales Generales	26
4.1.2. Sensor de Fuerza	27
4.1.3. Sensor de Movimiento Inercial (IMU)	28
4.1.4. Ensamble Brazaletes	29
4.2. Registro y transmisión de información	32
4.2.1. Microcontrolador	32
4.2.2. Descripción del Algoritmo desarrollado [Protocolo interno]	33
5. Elaboración de Base de Datos	40
5.1. Descripción general	40
5.1.1. Posiciones y agarres	43
5.1.2. Sujetos de Prueba	44
5.1.3. Resumen	47
5.2. Registro de información en el computador	48
5.2.1. Algoritmo desarrollado	48
5.2.2. Descripción del procedimiento práctico	51
5.2.3. Descripción de los archivos sin procesar	57
5.2.4. Descripción de archivos procesados	57
6. Algoritmos De Clasificación	64
6.1. Descripción Pruebas realizadas	64

Tabla de contenido

6.2. Regresión Logística One VS All	65
6.2.1. Método Clasificación One Vs All Regresión Logística, un ensayo, una posición, cuatro agarres, para cada sujeto.	65
6.2.2. Método Clasificación One Vs All Regresión Logística, cuatro ensayos, dos posición, cuatro agarres, para cada sujeto con IMU.	68
6.2.3. Método Clasificación One Vs All Regresión Logística, cuatro ensayos, dos posición, cuatro agarres, para cada sujeto sin IMU.	72
6.2.4. Comparación Método SVM Cuadrático con Método One Vs All Regresión Logística, cuatro ensayos, una posición, cuatro agarres, tres sujetos.	78
6.2.5. Comparación Método SVM Cuadrático con Método One Vs All Regresión Logística, cuatro ensayos, dos posiciones, cuatro agarres, tres sujetos con IMU. . . .	79
6.2.6. Comparación Método SVM Cuadrático con Método One Vs All Regresión Logística, cuatro ensayos, dos posiciones, cuatro agarres, tres sujetos sin IMU. . . .	81
6.2.7. Comparación Método SVM Cuadrático con Método One Vs All Regresión Logística, Tres ensayos de entrenamiento un ensayo de validación , dos posiciones, cuatro agarres, tres sujetos.	83
7. Aplicación para Validación Online	85
8. Conclusiones	90
9. Trabajos futuros	92
Anexos	
A. Anexos: General	95

Tabla de contenido

B. Anexo I: Código en lenguaje C – Programa del microcontrolador	96
C. Anexo II: Código – Aplicación en Matlab para el registro de datos	120
D. Anexo III: Código – Aplicación en Matlab para la validación On-Line	132
Bibliografía	146

Lista de Tablas

2-1. Conjunto de entrenamiento Regresión Lineal Una Variable	18
2-2. Conjunto de entrenamiento Regresión Lineal Multivariable	19
4-1. Materiales Fabricación Brazaletes	26
5-1. Clases a Registrar	44
5-2. Voluntarios Sanos	45
5-3. Relación Ensayo, registró y Clases	48
6-1. Relación Exactitud y Error Clasificador 1	66
6-2. Relación Exactitud y Error Clasificador 2	69
6-3. Relación Exactitud y Error Clasificador 3	73
6-4. Diferencia en la Exactitud del conjunto de validación con relación al uso de los sensores de Movimiento Inercial IMU	77
6-5. Comparación Método SVM Cuadrático con Método One Vs All Regresión Logística, cuatro ensayos, una posición, cuatro agarres, tres sujetos	78
6-6. Comparación Método SVM Cuadrático con Método One Vs All Regresión Logística, cuatro ensayos, dos posiciones , cua- tro agarres, tres sujetos	80
6-7. Comparación Método SVM Cuadrático con Método One Vs All Regresión Logística, cuatro ensayos, dos posiciones , cua- tro agarres, tres sujetos, Sin IMU	82

Lista de Tablas

6-8. Comparación Método SVM Cuadrático con Método One Vs All Regresión Logística, Tres ensayos de entrenamiento un ensayo de validación , dos posiciones, cuatro agarres, tres sujetos	83
7-1. Relación exactitud y Error Clasificador 1	88
7-2. Relación exactitud y Error Clasificador 1	89
7-3. Relación exactitud y Error Clasificador 1	89

Lista de Figuras

3-1. Diagrama Sistema de Identificación de Intencionalidad de Movimiento	23
4-1. Diagrama Brazaletes	25
4-2. FSR	27
4-3. IMU	28
4-4. Brazaletes Interno	29
4-5. Brazaletes Frontal	30
4-6. Canal Cableado Brazaletes	30
4-7. Brazaletes antebrazo	31
4-8. Brazaletes antebrazo con IMU, Brazaletes brazo con IMU . . .	31
4-9. Tarjeta de desarrollo Arduino Mega2560	33
4-10. Protocolo de registro de información y transmisión al computador	34
4-11. Diagrama de Flujo - Rutina Principal	37
4-12. Diagrama de flujo – Rutina de coordinación de procesos (Rutina de Interrupción por comparación del Temporizador No. 2)	38
4-13. Continuación Diagrama de flujo – Rutina de coordinación de procesos	39
5-1. Agarres	43
5-2. Posiciones	44
5-3. Formato Consentimiento	46
5-4. Protocolo Matlab	51

Lista de Figuras

5-5. Identificación Sensores FSR	52
5-6. Identificación Sensores IMU	52
5-7. Visualización Sensores FSR	53
5-8. Visualización Sensores IMU	53
5-9. Ubicación Brazaletes	54
5-10. FSR1 referencia ajuste brazaletes	55
5-11. Sensores IMU	56
5-12. Formulario	56
5-13. Primeras dos filas, Primer Archivo Raw Data	57
5-14. Primeras dos filas, Primer Archivo Base Datos	58
5-15. Datos FSR ensayo 1 Posición 1 agarres 1 2 3 4	59
5-16. Datos IMU ensayo 1 Posición 1 agarre 1 2 3 4	60
5-17. Datos FSR ensayo 1 Posición 2 agarres 1 2 3 4	61
5-18. Datos IMU ensayo 1 Posición 2 agarres 1 2 3 4	62
6-1. Matriz de confusión C1 S1yS2	65
6-2. Matriz de confusión C1 S3yS4	66
6-3. Matriz de confusión C1 S5yS6	67
6-4. Matriz de confusión C1 S7yS8	67
6-5. Matriz de confusión C1 S9yS10	68
6-6. Matriz de confusión C2 S1yS2	70
6-7. Matriz de confusión C2 S3yS4	70
6-8. Matriz de confusión C2 S5yS6	71
6-9. Matriz de confusión C2 S7yS8	71
6-10. Matriz de confusión C2 S9yS10	72
6-11. Matriz de confusión C3 S1yS2	74
6-12. Matriz de confusión C3 S3yS4	74
6-13. Matriz de confusión C3 S5yS6	75
6-14. Matriz de confusión C3 S7yS8	75
6-15. Matriz de confusión C3 S9yS10	76
6-16. Matriz de Confusión comparación Regresión Logística con SVM	79

Lista de Figuras

6-17. Matriz de Confusión comparación Regresión Logística con SVM	80
6-18. Matriz de Confusión comparación Regresión Logística con SVM	82
6-19. Matriz de Confusión comparación Regresión Logística con SVM	84
7-1. Ventana - Aplicación para validación On-line	86

1. Introducción

1.1. Introducción

La evolución tecnológica ha impulsado el desarrollo de nuevas técnicas y procesos, enfocados específicamente a poblaciones que presentan discapacidad relacionada a la afectación de una o varias de sus extremidades. Aunque el gobierno colombiano no tiene cifras exactas, el último reporte del DANE estima que alrededor de 2.624.898 personas presentan alguna discapacidad [1]. Desde el año 2002 la recopilación de estos datos está a cargo del observatorio nacional de discapacidad, que mediante el **RLCPD** (*registro de Localización y Caracterización de Personas con Discapacidad*) [2], alimenta y actualiza los registros de este sector poblacional a nivel nacional.

Sin embargo, no todas las personas discapacitadas se encuentran inscritas en el **RLCPD**; Actualmente los indicadores arrojan un aproximado de 1.342.222 personas con discapacidad, de los cuales el 34.05 por ciento se relaciona con problemas en la locomoción de manos, brazos y piernas [2][3], ya sea por factores de enfermedad, accidentes, o conflicto armado.

Actualmente el nivel tecnológico en el hardware y software facilita e impulsa la fabricación de prótesis robóticas con altos índices de calidad [4]. Sin embargo, el control de estas manos biónicas es deficiente, a pesar de que muchos profesionales se han dedicado a la investigación de esta área no se ha logrado implementar un óptimo control[4].

Todos los mecanismos de control parten de la necesidad de identificar apropiadamente la intención de movimiento del miembro superior, de ello depende que la acción ejecutada por los actuadores en la prótesis cumplan con lo deseado por el usuario. De acuerdo a lo reportado en la bibliografía existen diversas alternativas, entre ellas los resultados que arrojan las técnicas de electromiografía, ecografía, y miografía de la fuerza son similares[4], sin embargo en algunos experimentos donde se hizo uso de sensores de fuerza, se evidenció un mejor comportamiento en la interfaz hombre máquina[5] lo cual puede estar co-relacionado con el hecho de que los datos registrados por medio de sensores mioeléctricos tiene el inconveniente de no ser constantes en el tiempo a diferencia de los sensores de fuerza. Esto aunado a que la brecha de costo es bastante grande entre los métodos de electromiografía de superficie, electromiografía intramuscular y ecografía, respecto a la miografía de la fuerza, hacen bastante llamativo el uso en células de carga como método para la identificación de intención de movimiento[4][5]. Por otro lado algunos estudios han tomado en consideración la posición del brazo de los sujetos para mejorar el desempeño de los sistemas de identificación [6], siendo notable que la incorporación de sensores para esta labor no incrementa demasiado los costos de fabricación. Estos dos aspectos: uso de sensores de fuerza y reconocimiento de la posición del miembro superior del sujeto, constituyen la motivación principal de estudio del presente trabajo.

En este trabajo se implementó un sistema que identifica la intencionalidad del movimiento del miembro superior utilizando sensores de fuerza y sensores de movimiento inercial. En primera instancia se fabricó un brazalete con dos módulos: el primer módulo contiene ocho sensores de fuerza (FSR) en su cara interna y un sensor de movimiento inercial en su cara externa, el cual se ubica en el antebrazo del sujeto de prueba. El segundo módulo se ubica en el brazo del sujeto y solo contiene un sensor de movimiento inercial, su función es complementaria al sensor de movimiento inercial que se encuentra ubicado en el módulo No. 1, la comparación entre ambos permite identificar de manera automática la posición adoptada por el miembro superior del sujeto.

Se estableció un protocolo en el cual los voluntarios tuvieron que mantener dos posiciones del brazo en plano sagital combinados con cuatro diferentes agarres, los datos de presión y posición para cada combinación fueron registrados mediante la interfaz brazaletes-microcontrolador y microcontrolador-computador.

Los datos capturados fueron procesados y almacenados en una base de datos, a su vez esta fue utilizada para el entrenamiento y validación de algoritmos de clasificación supervisada mediante las técnicas de regresión logística y maquinas de soporte vectorial.

Los resultados obtenidos con el prototipo confirman la viabilidad de este tipo de configuración para realizar la identificación de movimientos de miembro superior, sin embargo las pruebas de validación on-line reflejan la necesidad de profundizar en el estudio de estos mecanismos de tal forma que se solventen algunos inconvenientes observados de cara a su implementación práctica.

1.2. Estado del arte

El nivel tecnológico actual permite un alto performance en el desarrollo de prótesis robóticas. No obstante, el control de estas manos biónicas no es eficiente.

El principal problema radica en la ineficiente interfaz hombre-maquina (**HMI**), específicamente cuando se quieren controlar más de dos grados de libertad. La capacidad de control se define como la activación del movimiento ante la intención de amputado[5]; En la actualidad la mayoría de las prótesis hacen uso de señales de electromiografía (**EMG**), mediante la utilización de sensores mioeléctricos ubicados sobre una extremidad [7] [6].

El reconocimiento de patrones, en el ámbito de la electromiografía (**EMG**), es de gran importancia para el desarrollo del control de dispositivos prostéticos, dado que permite realizar la clasificación de datos adquiridos. Fougner realizo el procesamiento de estos datos, para así excluir los datos erróneos y disminuir el error en la clasificación; no obstante incluyo en sus experimentos, datos captados por acelerómetros [7].

Con el fin de lograr una mayor exactitud en la clasificación, Khushaba realizó la adquisición de señales mioeléctricas, variando la posición del brazo tanto en hombres como mujeres sanos; de igual manera recopiló datos de hombres y mujeres con extremidades amputadas, en el momento en que ellos imaginaban la realización de una serie de movimientos, con el fin de lograr una contracción muscular; Por otra parte al momento de realizar la clasificación utilizó un enfoque alternativo considerando diferentes métodos para la extracción de características, como lo son las redes neuronales profundas mediante el aprendizaje autónomo por regresión logística [6].

Con ánimo de dar una posible solución a las personas que presentan amputación en miembros superiores, Benatti implementó una propuesta de prótesis, la cual integra un algoritmo de aprendizaje supervisado, cuyo objeto

es producir un control preciso y robusto de la prótesis. El avance en este algoritmo de aprendizaje supervisado mejoraría considerablemente el rendimiento de los métodos de regresión logística. Adicionalmente la prótesis cuenta con una *puerta de enlace* personal móvil, que se implementa en un teléfono inteligente, permitiendo el monitoreo y actualización del *firmware* de la prótesis [8].

Mattioli realizó estudios sobre prototipos de prótesis que basan su funcionamiento en algoritmos de aprendizaje supervisado, afirmando que estos presentan un buen desempeño en la clasificación de datos. Los experimentos se realizaron para el procesamiento de señales en tiempo real, donde se tendrá que tener en cuenta los posibles retardos ocasionados por la exclusión del procesamiento de datos fuera de línea.[9].

Mediante la ejecución de experimentos, Nglehart validó métodos que permiten lograr una mayor eficiencia y eficacia del sistema clasificador, entregando predicciones con un alto nivel de exactitud; De igual manera demostró un consumo moderado en los recursos informáticos, considerando una mínima capacidad de almacenamiento[10].

Por medio del uso de herramientas como *Matlab*, Zhang determinó que si se obtiene una mayor cantidad de datos se obtendrá una clasificación mucho más precisa [11]; Sin embargo esto aumentaría el tiempo de procesamiento necesario para la toma de una decisión, y si sumado a esto no se ejecuta el *post-procesamiento*, que es el período de la incorporación de todas las decisiones, el sistema de control tendría un retardo de operación demasiado largo [12][13].

En otro experimento, Chan realizó la captura de datos por medio de sensores mioeléctricos; no obstante, se puso en práctica el método de clasificación de modelos ocultos de Markov. Al realizar el prototipo se encontraron resultados de exactitud del 94.63 por ciento más eficiente en 2 puntos comparados

con otros métodos de clasificación, este enfoque describe un óptimo comportamiento para la implementación de un Control mioeléctrico Continuo, enfocado a prótesis [14].

Jiang Inspecciono y analizo las respuestas de señales mioeléctricas, pero en este caso, adicional al estudio de las extremidades o miembros[7] se incluyó en la adquisición de datos la medula espinal, mediante la micro estimulación. Utilizando los datos adquiridos se diseñó e implemento un modelo capaz de generar señales de electromiografía (**EMG**) de manera artificial.[15].

Eic J. desarrollo estudios en conjunto con personas que presentaban amputación parcial en extremidades superiores (*mano*); del fragmento de la muñeca registró datos de señales de electromiografía (**EMG**), justo en el momento en el que realizaba una serie de movimientos. Utilizo los datos para definir los criterios de operación de un clasificador el cual se integró a un control prostético.[16] .

Gabrilea W efectúo una serie de experimentos: registró datos de señales mioeléctricas de miembros superiores a personas sanas mientras realizaban una serie de movimientos; Posteriormente los datos fueron sometidos a un pre- procesamiento, donde se eliminó el offset y se rectificó y segmentó la señal para poder realizar la extracción de características. Con base en la caracterización se introdujo una herramienta de *red neuronal artificial*, en donde demostró una gran eficiencia dado a la velocidad de procesamiento y a la simplicidad en la ejecución de cálculos y tiempo utilizado, optimizando los recursos necesarios para su funcionamiento [17] .

L.Hargrove, mediante el uso de un dispositivo ortopédico integrado por electrodos, realizó el registró de señales mioeléctricas correspondientes a siete clases de movimientos ejecutados por seis sujetos sanos, durante la realización de contracciones isométricas. Posteriormente se realizó el procesamiento y clasificación de datos con el fin de entrenar un sistema de control

virtual; Posteriormente se permitió ingresar a los sujetos al entorno virtual, en donde todos los movimientos efectuados se transmitían en tiempo real y en condiciones físicas reales [18].

En la rehabilitación de una mujer que sufrió un accidente cerebro vascular ocasionado la perdida de la movilidad de un brazo, se diseñó un exoesqueleto el cual realiza los movimientos de la mano.[19] El exoesqueleto posee cinco dedos individuales, cada dedo es accionado por un actuador lineal que puede tirar y empujar. El funcionamiento se basa en la detección de señales mioeléctricas, generadas por pequeñas contracciones de los músculos que de acuerdo con patrones preestablecidos predicen el movimiento que se desea realizar y lo ejecuta [19] [20].

No obstante, las prótesis anteriormente consultadas están diseñadas para personas que cuentan con el miembro [19] o personas con amputaciones que tiene un fragmento del humero[8]. Lauretti diseñó una prótesis para personas con amputación *trans Humeral* [21].Esta prótesis deberá ser colocada en el área que comprende el tronco y muñón del paciente con el fin de evaluar los posibles ángulos del hombro, la flexión-extensión y abducción. A comparación de otras prótesis que solo se evalúan los las señales mioeléctricas de los músculos bíceps, tríceps y hombro, esta prótesis tiene en cuenta las señales de la medula espinal, posición, ángulo e inclinación [21]

Kuiken, desarrollo una prótesis variando la forma del registró de las señales mioeléctricas. En los sistemas convencionales las señales de electromiografía (**EMG**) son captadas directamente por electrodos sobre la piel. Esta prótesis utiliza una técnica más invasiva, la *re inervación* realiza el muestreo de las señales de electromiografía (**EMG**) directamente sobre el musculo en un implante por debajo de la piel. Según los resultados de los experimentos propuestos, el control en la clasificación de datos es mucho más eficiente, dado a que la señal mioeléctrica es adquirida con mayor potencia y velocidad [22].

Por otro lado, Vikram y Castellini incorporaron nuevos procesos con el fin de optimizar la interfaz hombre-maquina (**HMI**). Realizaron un experimento del cual fueron participes 10 sujetos, las pruebas realizadas a los pacientes se basaban en el movimiento de los dedos de las manos al momento de realizar un agarre, durante esta actividad se registraron señales mediante sensores mioeléctricos, células de carga y sistemas de ultrasonido simultáneamente[4].

Para electromiografía se encontró una alta eficiencia en sistemas de control de dos grados de libertad (On-Off)[5][4]; No obstante, al introducir un tercer grado de libertad su eficiencia disminuye; De igual manera se identificó que presenta alta variabilidad en los datos registrados a través del tiempo dado a factores como: Condiciones ambientales, condiciones fisiológicas y sudor del paciente y fatiga en sus materiales, ocasionando una constante calibración de los sensores. Estas circunstancias hacen que los pacientes pierdan el interés en el uso de prótesis cuya interfaz involucré el uso de sensores mioeléctricos [4].

En experimentos enfocados al ultrasonido se utilizó un equipo hospitalario portátil. La técnica consiste en captar las imágenes de ecografía en el momento en el que se realiza un tipo de agarre, se realizó la caracterización y posteriormente se implementó un algoritmo de clasificación supervisada. Los resultados expuestos de este método demostraron una efectividad similar a las técnicas de electromiografía; no obstante, demostró ser más estable en cuanto a sistemas que incluyan más de dos grados de libertad.[23].

Las células de carga o sensores de fuerza son elementos utilizados en la miografía de la fuerza, son sensores que varían su valor resistivo al detectar cambios de presión sobre su superficie. En pruebas realizadas fueron ubicados sobre las yemas de los dedos de la mano de los sujetos de prueba, se registraron datos en el momento en que se realizaba un agarre donde se evidencio variabilidad en los datos registrados; No obstante, el rango de variación era bajo y no se consideró como afectación en la implementación

de un algoritmo de clasificación.[4].

1.3. Planteamiento del Problema

En el mercado se pueden encontrar varias alternativas de prótesis, las cuales se diferencian en el objetivo para el cual fueron diseñadas, hay prótesis básicas creadas con el único objeto de aparentar la presencia del miembro faltante, o cumplen una función de apoyo-soporte en miembros inferiores o agarres fijos en miembros superiores, si bien estas presentan un bajo nivel tecnológico, también encontramos prótesis robóticas o avanzadas cuyo principal objetivo es la locomoción a partir de la intencionalidad de movimiento del amputado.

Por otro lado, de acuerdo a lo presentado en la sección anterior, aunque las prótesis avanzadas tienen un alto desempeño en cuanto a su hardware y software, su control aun es deficiente, el principal problema radica en una ineficaz interfaz hombre maquina (HMI). La mayoría de manos robóticas utilizan electrodos de superficie o intramusculares para identificar la intencionalidad del movimiento, también suelen ser usados sistemas de ultrasonido y de posicionamiento.

En el caso del uso de electrodos de superficie, estos presentan un buen desempeño en prótesis de dos grados de libertad o al ejecutar dos acciones: activar y desactivar, pero al momento de utilizarlos en prótesis de más de dos grados de libertad baja su desempeño; También presentan una alta varianza en las señales captadas, dado a cambios de humedad en la piel por sudor y circunstancias fisiológicas, pueden verse afectados por señales externas, sus materiales presentan fatiga y sus señales son variables en el tiempo.

De igual manera, los electrodos intramusculares si bien, pueden tener una leve mejora respecto a los de superficie son de carácter invasivo. Además, los sistemas de ultrasonido y posicionamiento presentan un alto margen de error y costo, e identifica solo la posición del miembro lejos de identificar la intencionalidad de movimiento respectivamente.

1.4. Objetivos

1.4.1. Objetivo General

Implementar un sistema que permita identificar la intencionalidad de movimientos del miembro superior, por medio del uso de sensores de fuerza y sensores de movimiento inercial.

1.4.2. Objetivos específicos

- Fabricar un brazalete que contenga los sensores de fuerza y los sensores de movimiento inercial, este se acoplara en la parte alta del antebrazo.

- Elaborar una base de datos a partir de las señales extraídas de los sensores de fuerza y de los sensores de movimiento inercial, con diferentes agarres y posiciones.

- Implementar un algoritmo clasificador que permita identificar la intencionalidad de movimiento a partir de la información adquirida.

- Desarrollar un aplicación de usuario mediante la cual se puedan ejecutar pruebas On line del algoritmo clasificador.

2. Marco Teórico

2.1. Técnicas Actuales de Identificación

2.1.1. Empleando electromiografía

Al indagar en el estudio de sistemas que permiten la identificación de la intencionalidad del movimiento, se encuentra en gran porcentaje técnicas clásicas como la electromiografía, que basa su funcionamiento en el registro de señales mioeléctricas. Mediante electrodos se registra la actividad eléctrica del músculo al ejecutar una serie de agarres o movimientos[4].

Dentro de los métodos de electromiografía se puede intuir una clasificación dependiendo del uso, tipo y ubicación de los electrodos[4][23]. La electromiografía de superficie emplea electrodos que son ubicados en la superficie de la piel del miembro del cual se desea obtener registros. No obstante, al usar electrodos de superficie estos presentan una gran variabilidad en los datos capturados dado a factores como la anatomía y fisiología del paciente, temperatura y sudor, sin mencionar que son susceptibles a cambios en sus parámetros de calibración ante la línea del tiempo[23].

Por otro lado, se encuentran los electrodos intramusculares que son ubicados bajo la piel del paciente, esta técnica reduce potencialmente los errores generados en la electromiografía de superficie, aumenta la exactitud y exactitud de los datos capturados, pero es una técnica con un alto nivel de invasión dado la naturaleza de los electrodos[23].

2.1.2. Empleando electroencefalografía

La Electroencefalografía es utilizada en diversos procesos en el diagnóstico de ciertas patologías relacionadas con la actividad cerebral. No obstante, también ha sido utilizado en el entrenamiento de algoritmos que permiten la identificación en la intencionalidad del movimiento de miembros[4].

Mediante un encefalograma se fija una malla de electrodos en el cuero cabelludo del paciente o sujeto de prueba y se registran los datos de las ondas cerebrales en el momento que el sujeto realiza un movimiento específico. Generalmente estos registros son captados simultáneamente con otros patrones como señales de electromiografías corrientes, incluso se han hecho investigaciones con el procesamiento de imágenes captadas por una cámara en el momento que el sujeto realiza la acción[4].

Las señales de electroencefalografía necesitan ser calibradas teniendo en cuenta un parámetro como; el género, la edad y patologías presentes entre otros. Los algoritmos entrenados por estos sistemas han obtenido un resultado satisfactorio, pero el costo en el desarrollo de los mismos es demasiado alto debido a la instrumentación utilizada[4].

2.2. Sensores

2.2.1. Sensores de Fuerza

Los sensores de fuerza hacen parte de la familia de sensores resistivos, su funcionalidad consiste básicamente en el cambio del valor resistivo entregado por sus dos terminales en el momento que se ejerce una presión o fuerza sobre su área activa[23].

Por consiguiente es necesario un flujo de corriente eléctrica a través del sen-

2. Marco Teórico

sor, dependiendo de la fuerza aplicada este generara una oposición de la misma que se relaciona directamente de parámetros en la construcción del sensor como la resistividad y dimensión del material[23].

En algunos casos la señal de salida de estos sensores debe pasar por una etapa de amplificación[23].

2.2.2. Sensores de Movimiento Inercial

Los sensores IMU o por sus siglas unidad de medición inercial permiten conocer la orientación y posición de un objeto en el espacio, mediante el registro y combinación de la información suministrada por acelerómetros giroscopios y en algunos casos magnetómetros, cuando se refiere a una IMU de 6DOF es que tiene acelerómetro y giroscopio, la IMU de 9 DOF incluye el magnetómetro, las siglas (DOF Degrees Of Freedom) un grado de libertad por cada eje X, Y, Z.

El acelerómetro es capaz de medir la aceleración en los tres ejes y usa como referencia el valor de la aceleración de la gravedad de la tierra 9.8 m/s². Al ubicar la IMU sobre su eje y este marcara el valor 9.8 si por el contrario lo giramos al eje x este es el que marcara 9.8; de esta manera y con relaciones trigonométricas podemos estimar un Ángulo de inclinación.

El giroscopio basa su funcionamiento en la medición de la velocidad angular , para el calculo del ángulo se utiliza la siguiente relación matemática.

$$AnguloY = AnguloY Anterior + GiroscopioY.\Delta t \quad (2-1)$$

La formula varia dependiendo de los ejes a procesar, para medir el ángulo en X se cambian la variables de Y por X.

2. Marco Teórico

Calculo de filtro complementario

El funcionamiento de las IMU se ve afectado por factores como el ruido del cual son susceptibles los dispositivos electrónicos, de igual manera el acelerómetro es capaz de reconocer cualquier otra aceleración externa a parte de la de la tierra. El giroscopio arroja medidas mas precisas con un pequeño error. No obstante, este error es de carácter acumulativo y con el tiempo se genera un drift.

Por ende para la corrección de estos factores se hace necesario la implementación de un filtro.

El filtro Complementario esta compuesto por dos filtros:

- El filtro High-pass Filter es aplicado para el giroscopio y su función es dejar pasar valores por encima de cierto parámetro.
- El Low-pass Filter es aplicado al acelerómetro y permite el paso de valores por debajo del parámetro establecido.

La ejecución de este filtro se realizo por medio de la siguiente expresión matemática:

$$Angulo = 0,98 * (Angulo + AnguloGyro.\Delta t) + 0,02.AnguloAccel \quad (2-2)$$

2.3. Aprendizaje de máquinas o aprendizaje automático

El termino Aprendizaje de máquinas no es familiar o conocido por la mayoría de las personas, no obstante, cotidiana y frecuentemente hacemos el uso de algoritmos de aprendizaje automático en aplicaciones muy comunes, por ejemplo al utilizar motores de búsqueda como Google o Bing, al recibir un correo electrónico clasificado como spam, en el momento que interactuamos con robots de chats, o cuando gestores de almacenamiento en la nube clasifican las imágenes o cuando las redes sociales exponen contenido dependiendo del comportamiento o preferencias del usuario.

Arthur Samuel, definió el aprendizaje automático como el campo de estudio que da a las computadoras la capacidad de aprender sin enseñarles explícitamente[24]. Samuel se dio a conocer en el año de 1950 gracias al desarrollo de un software cuyo objetivo era jugar damas. Si bien Arthur no era un buen jugador, la técnica que utilizo para implementar el algoritmo fue programar miles de partidas contra sí mismo, determinando que movimientos en el tablero podrían conducir a victorias o derrotas. El programa aprendió con que jugadas podría tener la victoria siendo mucho mejor jugador que el mismo Samuel[24].

Por otro lado, Tom Mitchell expone que de una manera mucho más técnica lo siguiente: un programa de computadora aprende de la experiencia E con respecto a alguna tarea T y alguna medida del rendimiento P , si su rendimiento en T , medido por P , mejora con la experiencia E . Si se explica la definición con el juego de damas sería así: La experiencia E sería la experiencia del programa, La tarea T sería jugar damas medida del rendimiento P será la probabilidad de ganar la próxima partida contra algún oponente nuevo[24].

Dentro de los tipos de algoritmos de Machine Learning o Aprendizaje automático, resaltan dos principales: Algoritmos de aprendizaje supervisado y algoritmos de aprendizaje no supervisado[24].

2.3.1. Aprendizaje Supervisado

Los algoritmos de aprendizaje supervisado basan su funcionamiento en conjuntos de muestras de datos, las cuales son mediciones de variables de entrada relacionadas directamente con una etiqueta que define el valor de salida[25]. Es decir, dado un conjunto de datos se conoce cual es la salida correcta, estableciendo una relación directa entre la entrada y la salida[24].

Por otra parte, Los problemas de aprendizaje supervisado se clasifican en problemas de regresión y clasificación. Cuando se habla de regresión, se trata de predecir resultados para una salida continua, en cuando a la clasificación se trata de predecir resultados de una salida discreta [24].

Por ejemplo, se tiene una muestra de datos donde se relaciona el valor de casas respecto a su área en metros cuadrados; El precio en función del tamaño es una salida continua, por lo tanto, es un problema de regresión. Para el caso de clasificación se puede definir si un tumor es maligno o benigno dependiendo de variables de entrada [24] [25].

2.3.2. Aprendizaje No Supervisado

A diferencia del aprendizaje supervisado, estos algoritmos no presentan etiquetas que relacionen la salida del sistema, por lo tanto, se tienen muestras de datos, pero no se conoce ninguna relación con la salida. Es apto para la solución de problemas en los cuales se tiene poca o ninguna idea de cómo será el resultado[24].

2.4. Regresión Lineal

2.4.1. Una variable

La regresión lineal es un algoritmo de aprendizaje supervisado. Por lo tanto, el conjunto de datos o conjunto de entrenamiento brindara la respuesta correcta para cada uno de los ejemplos, de igual manera el termino de regresión hace referencia a la predicción del resultado el cual sera un valor continuo[24].

Para una mejor comprensión se expone un ejemplo:

Se requiere implementar un algoritmo cuyo objeto sera predecir el valor en el que se puede vender una casa. El primer paso es disponer de un conjunto de entrenamiento, que tendrá datos registrados relacionado el valor de varias casas respecto a su tamaño.

	Área en metros ² (x)	Precio(\$) millones de pesos (y)	
1	368	960.000.000	m = 44
2	232	520.000.000	
3	200	501.000.000	
4	67	295.000.000	
5	160	550.000.000	
44	

Tabla 2-1.: Conjunto de entrenamiento Regresión Lineal Una Variable

2.4.2. Multi variable

En la regresión lineal multivariable hay presencia de mas de una variable de entrada (x). para el ejemplo de estudio de predicción del valor de la casa además del área se agregaran características como el numero de habitaciones y numero de baños[24].

	Área en metros ²	Numero de habitaciones	Numero de baños	Precio(\$) millones de pesos	
	X1	X2	X3	Y	
1	368	5	7	960000000	m = 44
2	232	4	4	520000000	
3	200	4	3	501000000	
4	67	3	2	295000000	
5	160	4	2	550000000	
44	
n=3					

Tabla 2-2.: Conjunto de entrenamiento Regresión Lineal Multivariable

2.4.3. Regresión Logística

En el algoritmo de clasificación supervisada se registran las variables de entrada y se clasifican en clases discretas. Se observa la implementación de estos algoritmos en la clasificación del correo como spam o no spam, en los tumores malignos o benignos o en transacciones fraudulentas o legítimas. En común, todos los ejemplos expuestos tienen la característica de que las variables son clasificadas en dos clases: positivas o negativas: cero o uno es decir son problemas de clasificación binaria[24] .

la hipótesis se define como:

$$h = g(\theta^T X) \tag{2-3}$$

Donde g es la función sigmoidea o función logística y se representa como:

$$g(z) = \frac{1}{1 + e^{-z}} \quad (2-4)$$

Simplificando ambas relaciones matemáticas tenemos:

$$h(\theta) = \frac{1}{1 + e^{-\theta^T X}} \quad (2-5)$$

Función costo Regresión Logística

la función de costo para la regresión logística binaria se define como el costo de la función para clasificar la variable de entrada como clase positiva o clase Negativa[24].

$$Cost(h\theta(x), y) = -\log(h\theta(x)) \quad (2-6)$$

$$Cost(h\theta(x), y) = -\log(1 - h\theta(x)) \quad (2-7)$$

Descenso de gradiente Regresión Logística

Al igual que en la regresión lineal el descenso de gradiente permite encontrar el valor óptimo de los parámetros θ minimizando la función de costo[24].

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m (h\theta(x^i) - y^i) x_j^i \quad (2-8)$$

2.4.4. Máquinas de Soporte Vectorial

El algoritmo de regresión logística representa una muy buena función de aprendizaje en conjuntos separables linealmente. No obstante, al momento de aplicar un algoritmo de aprendizaje supervisada en sistemas complejos no lineales la regresión logística no representa una solución aceptable. El algoritmo maquina de soporte vectorial (SVM) funciona bastante bien al aplicarla en funciones complejas no lineales[24].

Función costo Máquinas de soporte vectorial (SVM)

La función de costo para maquinas de soporte vectorial a diferencia de la función de costo del algoritmo de regresión logística no minimizara el error hallando los parámetros óptimos de θ , realizará directamente una predicción de clase.

$$\min_{\theta} \sum_{i=1}^m C[y^i \text{cost}_1(\theta^T x^i) + (1 - y^i) \text{cost}_0(\theta^T x^i)] + \frac{1}{2} \sum_{j=1}^n \theta_j^2 \quad (2-9)$$

Hipótesis Máquinas de Soporte Vectorial (SVM)

La hipótesis en Maquinas de soporte vectorial sera la predicción respecto al entrenamiento del sistemas para $Y=0$; $Y=1$. Se representa:

$$\left. \begin{array}{l} h_{\theta}^x = 1, \theta_x^T \Rightarrow 0; \\ h_{\theta}^x = 0, \theta_x^T \Rightarrow < 0; \end{array} \right\} \text{hipótesis}$$

3. Descripción general de la Metodología

En este documento se presenta el desarrollo de un sistema para la identificación de intencionalidad del movimiento del miembro superior. En la figura 3.1 se observa un diagrama general de sistema desarrollado. los componentes principales son: sensores de fuerza, sensores de movimiento inercial, un microcontrolador y técnicas de aprendizaje automático.

Inicialmente se fabricó un dispositivo electrónico para registrar los datos de presión superficial ejercida por los músculos del antebrazo, el cual captura de forma simultánea la posición del miembro superior. Los sensores fueron dispuestos en un brazalete para permitir una fácil adaptación a los diferentes sujetos de prueba. Para coordinar el registro de las señales y su posterior envío al computador, se empleó un microcontrolador de la familia ATmega, específicamente un ATmega2560.

3. Descripción general de la Metodología

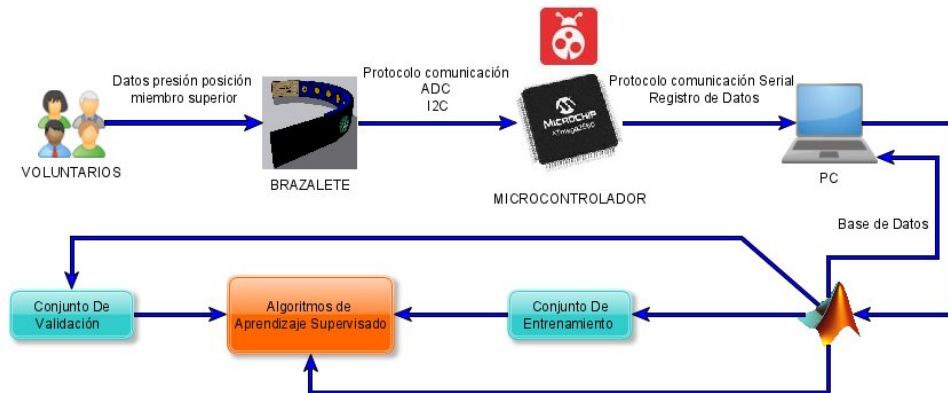


Figura 3-1.: Diagrama Sistema de Identificación de Intencionalidad de Movimiento

Empleando la herramienta Matlab se elaboraron diversos algoritmos para recopilar y analizar la información suministrada por el dispositivo electrónico. Se elaboró un protocolo estándar de medida y con base en él se registraron los datos provenientes de diez (10) voluntarios. Durante la fase de registro cada sujeto de prueba (voluntario) debía mantener una posición del brazo fija y un agarre de la mano, en total se evaluaron 8 combinaciones: 4 agarres y 2 posiciones diferentes por cada sujeto. Todas las medidas fueron almacenadas en el computador siguiendo una denominación estándar, con el objetivo de facilitar su posterior uso y análisis.

En la etapa siguiente se validaron dos (2) algoritmos de clasificación supervisada, Regresión Logística y máquinas de Soporte Vectorial. Se realizaron diversos ensayos para evaluar el desempeño del sistema en función de los diversos parámetros de medida, tales como sujeto, número de ensayos, número de posiciones etc. El objetivo principal de los algoritmos fue identificar de forma automática el agarre de la mano efectuado por el sujeto de prueba. Los resultados obtenidos reflejan un desempeño adecuado del sistema en general, principalmente en las pruebas off-line, sin embargo las validaciones

3. Descripción general de la Metodología

on-line indican que es necesario aumentar el número de ensayos por cada sujeto para disminuir el error en la predicción de los algoritmos.

4. Fabricación del Brazalete

Se fabrico un brazalete que se compone por dos módulos, el módulo uno se ubica en el antebrazo y contiene en su cara interna ocho sensores de fuerza (FSR) en su cara externa tiene un sensor de movimiento inercial (IMU); El módulo dos se ubica en el brazo y contiene en su cara externa un sensor de movimiento inercial (IMU).

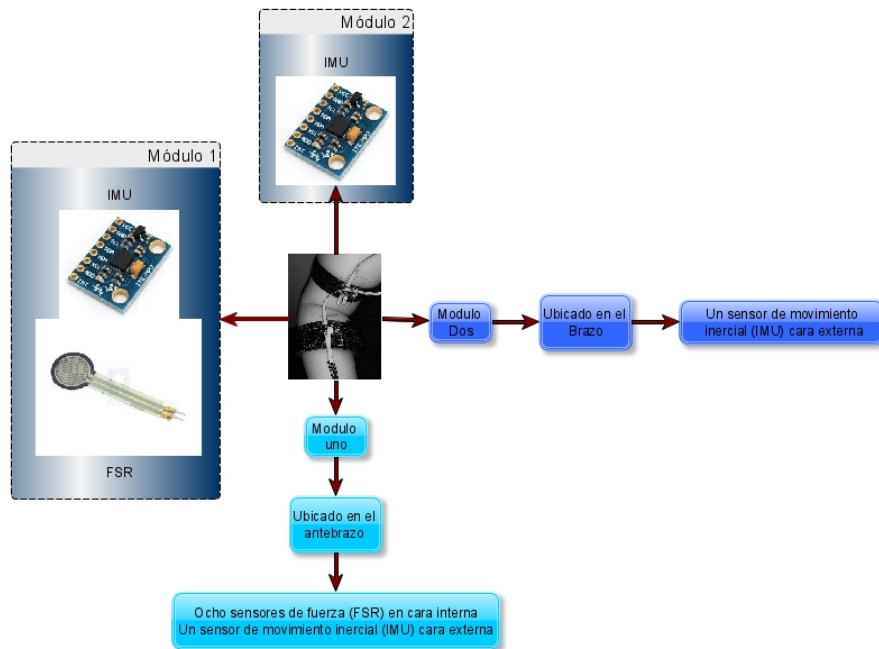


Figura 4-1.: Diagrama Brazalete

4.1. Sensores

4.1.1. Materiales Generales

Para la fabricación del brazalete se escogieron materiales de fácil adquisición y de un valor económico.

Elemento	Cantidad
Sensor FSR 402	8
Sensor Mpu 6050	2
Resistencias de exactitud 100.000 ohmios	8
Cable Cinta	3 (metros)
Funda termoencogible	3 (metros)
Tela cinta o Reata	1 (metro)
Cinta Velcro	1 (metro)
Foami	1 lamina
Lamina PVC	2

Tabla 4-1.: Materiales Fabricación Brazalette

4.1.2. Sensor de Fuerza

Se utilizó el sensor de fuerza FSR 402, Para el proceso de selección se tuvo en cuenta los siguientes parámetros.



Figura 4-2.: FSR

- El tamaño del sensor de 18.28 mm de diámetro y 12.7 mm de diámetro del área activa, es conveniente para la ubicación espacial dentro del brazalete, permitiendo distribuir un número óptimo de sensores sobre la superficie del antebrazo.
- Se realizó una prueba de diseño, basado en la caracterización plasma-da en la hoja de datos del sensor, los resultados fueron satisfactorios.
- Su operación no se ve afectada por cambios de temperatura dentro del rango de -30 a 70 grados Celsius.

- Los datos registrados no se ven afectados considerablemente a través del tiempo.
- Fue de fácil adquisición dentro del comercio de componentes electrónicos en la ciudad de Bogotá.
- Su Valor Comercial no es elevado respecto a otros sensores de fuerza.

4.1.3. Sensor de Movimiento Inercial (IMU)

Se utilizó el sensor de movimiento inercial MPU 6050, Para el proceso de selección se tuvo en cuenta los siguientes parámetros.



Figura 4-3.: IMU

- Su tamaño y presentación es adecuada para el desarrollo del brazalete.

- Sus prestaciones se adecuan a las necesidades del proyecto, dado que solo requiere registrar la posición o ángulos del brazo y antebrazo en modo estático no dinámico.
- Su operación no se ve afectada por cambios de temperatura y por el uso prolongado a través del tiempo.
- Fue de fácil adquisición dentro del comercio de componentes electrónicos en la ciudad de Bogotá.
- Su Valor Comercial no es elevado respecto a otros sensores de fuerza.

4.1.4. Ensamble Brazalete

Para el ensamble del Brazalete se utilizó componentes textiles como tela cinta o reata, cinta velcro, foami.



Figura 4-4.: Brazalete Interno

los ocho sensores de fuerza fueron ubicados sobre la cara del foami distanciados 15 milímetros uno del otro y con una base sólida echa con láminas de PVC.

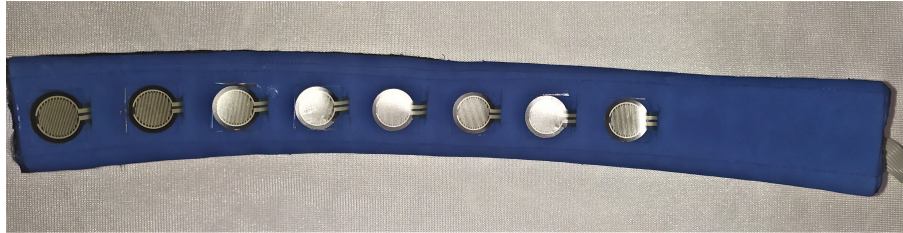
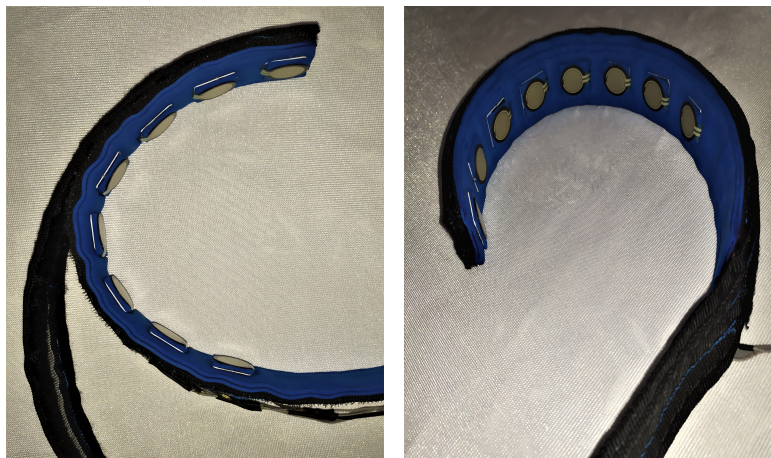


Figura 4-5.: Brazalete Frontal

El Cableado de los sensores y las resistencias estarán contenidos en un canal formado por tela velcro, que además sellara el brazalete.



(a) Canal Brazalete

(b) Brazalete

Figura 4-6.: Canal Cableado Brazalete



Figura 4-7.: Brazalete antebrazo

Para Los sensores de movimiento Inercial se elaboró una base en acrílico y tela velcro, esto con el fin de adherirlo al brazalet de los sensores de fuerza y a un segundo brazalet ubicado en el brazo.

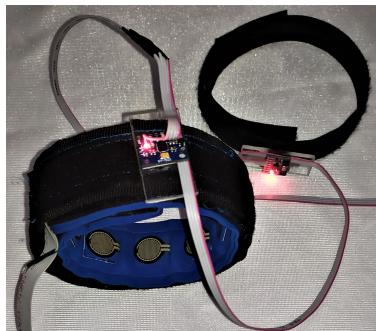


Figura 4-8.: Brazalet antebrazo con IMU, Brazalet brazo con IMU

4.2. Registró y transmisión de información

Las señales provenientes de los sensores de fuerza, y la información suministrada por los sensores de movimiento inercial, fueron recopiladas y posteriormente transmitidas a un computador para su almacenamiento, procesamiento y visualización. El dispositivo seleccionado para llevar a cabo esta labor fue el microcontrolador ATmega 2560, a continuación se explican los motivos por los cuales se utilizó este microcontrolador, y los detalles del algoritmo implementado (protocolo interno).

4.2.1. Microcontrolador

Los requerimientos mínimos establecidos, con los cuales debía contar el dispositivo fueron:

- ADC para el registró de las señales provenientes de los sensores de fuerza
- módulo de comunicación I2C para comunicarse con los sensores de movimiento inercial.
- módulo de comunicación USB o USART para la transferencia de información hacia y desde el computador.
- Temporizadores para coordinar los diferentes procesos y garantizar los tiempos de muestreo.

Se eligió el microcontrolador ATmega 2560 ya que éste cuenta con 4 módulos USART, 1 módulo I2C, ADC de 10 bits y 16 canales, 5 temporizadores entre otros periféricos; comercialmente es de bajo costo y se encuentra integrado en el sistema de desarrollo Arduino Mega2560, lo cual facilita su uso permitiendo un acceso cómodo a los pines del microcontrolador y elimina la necesidad de elementos adicionales para su polarización; la programación se

puede realizar en lenguaje C y en un entorno de desarrollo libre (Atmel Studio). Todo lo anterior, aunado a la disposición del mismo, y la familiaridad con el entorno y lenguaje de programación, fueron los elementos principales para la elección de este microcontrolador. No se desarrolló un proceso de selección más minucioso debido a que las prestaciones de este microcontrolador eran suficientes para lo requerido en el proyecto, y al disponer con anterioridad del mismo, se facilitaba el desarrollo de todas las actividades, sin perjuicio de los objetivos planteados en el proyecto.

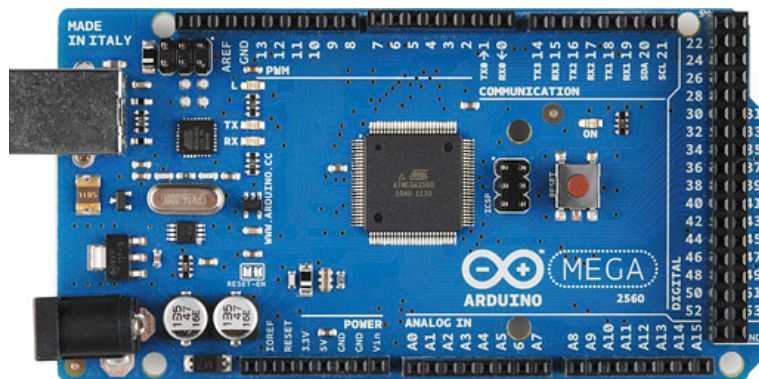


Figura 4-9.: Tarjeta de desarrollo Arduino Mega2560

4.2.2. Descripción del Algoritmo desarrollado [Protocolo interno]

El protocolo de registro de información y transmisión al computador define la manera mediante la cual se enlazan el brazaletes, el microcontrolador y el ordenador.

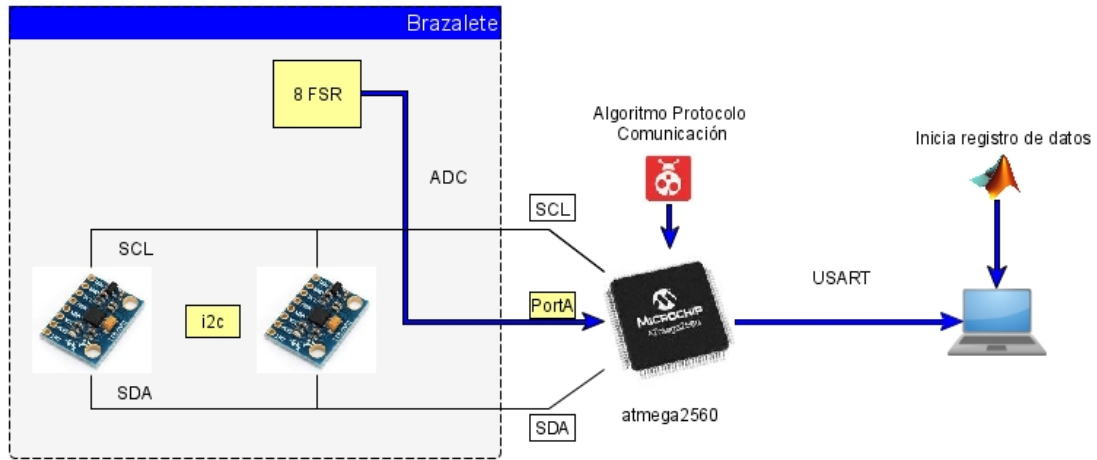


Figura 4-10.: Protocolo de registro de información y transmisión al computador

A continuación se describe la secuencia de pasos que ejecuta el microcontrolador durante el proceso de registro de información.

Se adjunta como Anexo A algoritmo Pregistro AT.

El algoritmo *Pregistro AT* establece los siguientes pasos:

- Al momento de energizar el microcontrolador, o después de un proceso de reinicio, lo primero que éste realiza es configurar apropiadamente todos los periféricos internos: puertos digitales, interrupciones externas, temporizadores, ADC, I2C, USART. Posteriormente procede a configurar los sensores de movimiento inercial, y si todo funciona adecuadamente ingresa en un estado de reposo e informa al usuario esta situación a través de 8 LEDs que se encuentran conectados al puerto digital A. Si algo falla durante la configuración, esta situación también es informada al usuario a través de los LEDs.

- Para iniciar el registro de datos, es necesario que el usuario presione el botón ‘Inicio’ en la aplicación de Matlab *Pregistro MT*, en ese momento el computador procede a enviar el número decimal ‘0’ al microcontrolador empleando comunicación serial (y el módulo USART del microcontrolador).
- Cuando el microcontrolador recibe el carácter (recepción a través de USART empleando interrupciones), se activa un temporizador el cual se encarga de coordinar todos los procesos que deben ser efectuados para el registro y transmisión de información, garantizando a su vez el tiempo de muestreo de las diferentes señales. Exactamente cada 40ms el temporizador activa la comunicación con los sensores inerciales (empleando el protocolo I2C) e inicia el registro de las señales analógicas provenientes de los sensores de fuerza (a través del ADC), en ese mismo momento procede a enviar al computador la información recopilada durante el último ciclo de medida, esto para garantizar la conformación adecuada de los paquetes de información y evitar la colisión o pérdida de datos.
- La información recopilada de los diferentes sensores es recopilada en un solo buffer para facilitar su manejo y conservarla durante el periodo de transmisión de información. El paquete de información está constituido por:
 1. Los 4 primeros bytes contienen información del estado del proceso. Los dos primeros indican si todo transcurre de manera normal (se colocan en ‘0’) o por el contrario se identificó algún inconveniente en el proceso de medida (se colocan en ‘1’). Los siguientes dos bytes informan el estado de registro de información, estos mantienen en ‘0’ cuando la información registrada está siendo utilizada únicamente para fines de visualización en la aplicación de Matlab, o por el contrario se colocan en ‘1’ para indicar que los valores serán almacenados en el computador para su posterior

análisis.

2. Los siguientes 14 bytes contienen la información recopilada del sensor de movimiento inercial No. 1. Ellos están compuestos por los valores de aceleración y velocidad angular detectados en cada uno de los ejes: X, Y y Z, y la Temperatura actual del sensor.
3. Los siguientes 14 bytes contienen la información recopilada del sensor de movimiento inercial No. 2.
4. Los siguientes 16 bytes contienen la información registrada por el ADC, y corresponden a las medidas (en 10bits) de los 8 canales a los cuales se encuentran conectados los sensores de fuerza.
5. Los siguientes dos bytes contienen un valor que se incrementa de forma consecutiva a medida que se van registrando nuevas muestras, esta valor permite identificar cuando se ha producido perdidas de información.
6. Los últimos dos bytes indican el final de la transmisión, siempre permanecen en 255 y pueden ser utilizados para la detección de errores de comunicación.

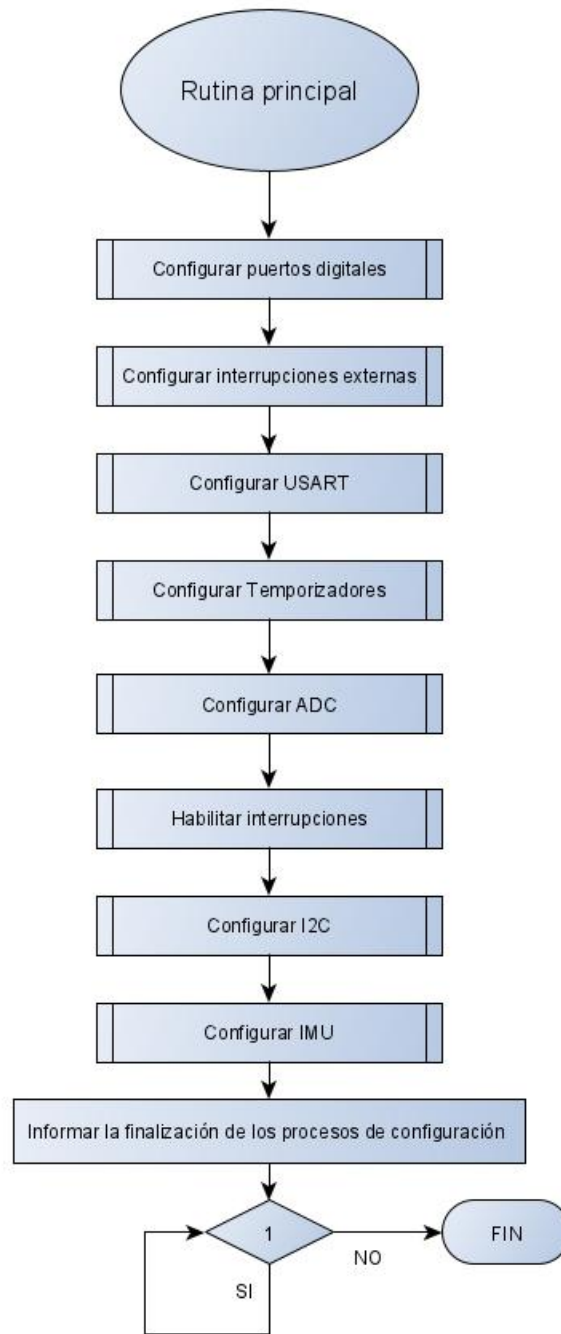


Figura 4-11.: Diagrama de Flujo - Rutina Principal

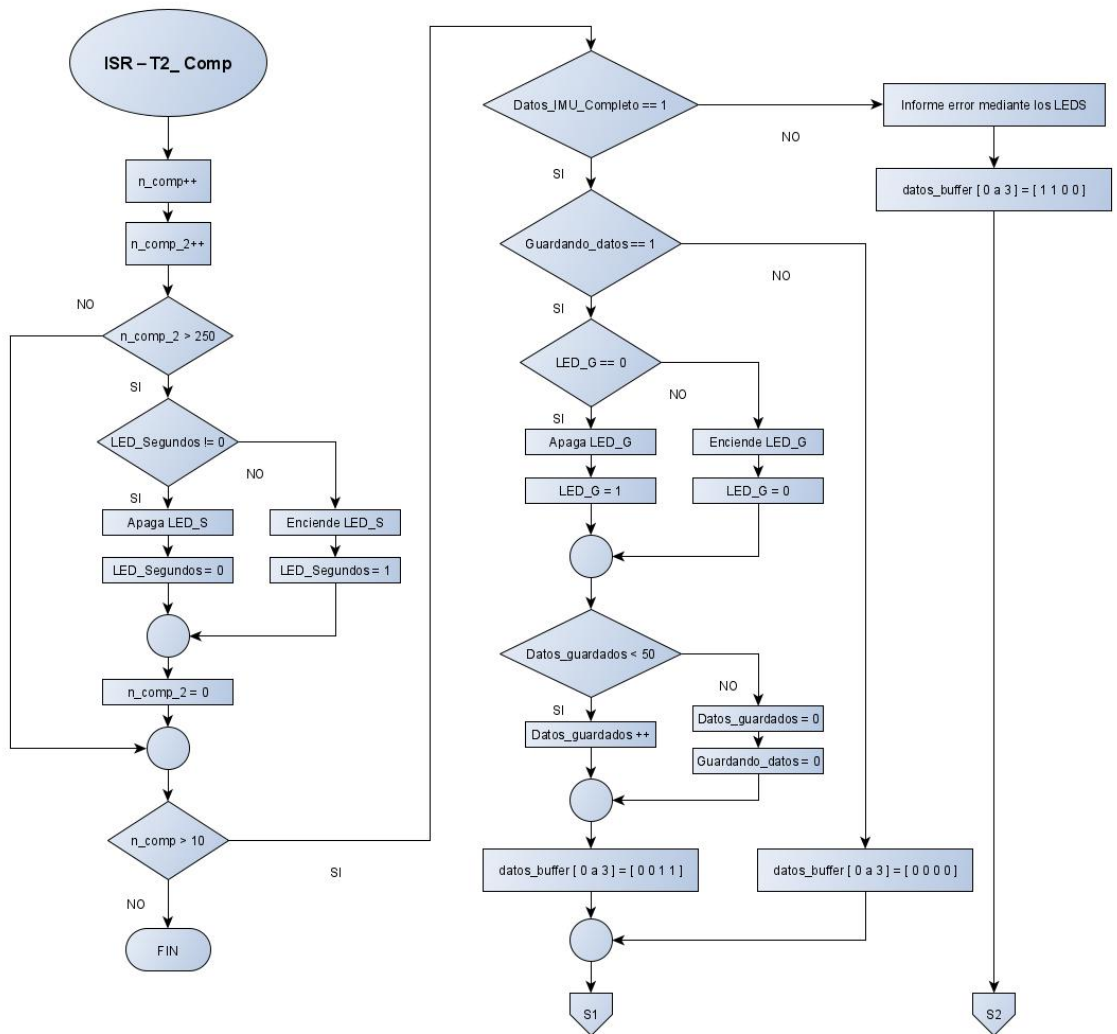


Figura 4-12.: Diagrama de flujo – Rutina de coordinación de procesos (Rutina de Interrupción por comparación del Temporizador No. 2)

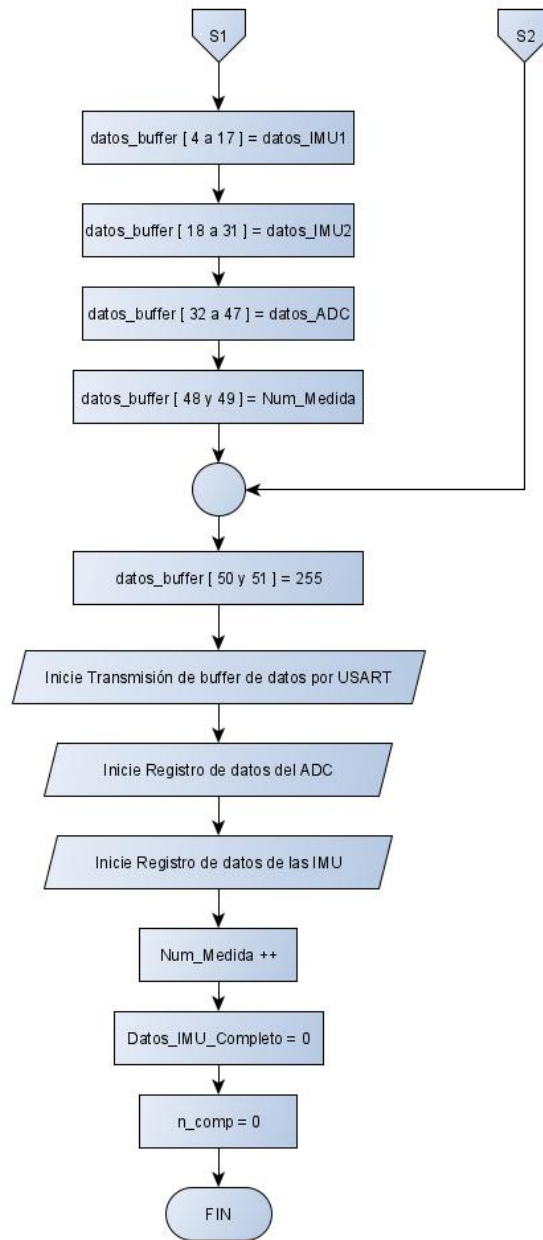


Figura 4-13.: Continuación Diagrama de flujo – Rutina de coordinación de procesos

5. Elaboración de Base de Datos

5.1. Descripción general

Las técnicas de aprendizaje automático requieren contar con un conjunto amplio de ejemplos de entrenamiento y validación, ellos son utilizados para calcular los parámetros óptimos que garanticen un buen desempeño del algoritmo. El procedimiento relacionado con la adquisición de estos ejemplos es referido comúnmente como “Elaboración de la base de datos”.

En términos generales podríamos describir una base de datos en función de dos parámetros principales: el número de muestras (ejemplos) y el número de características (variables) de cada muestra. Entre mayor sea el número de muestras, mejor será la representación estadística de la población y así mismo el desempeño del algoritmo (aunque el costo computacional de procesamiento se incrementará), y entre más grande sea el número de características más complejo será el modelo (por ejemplo un algoritmo de clasificación podrá identificar mas clases diferentes). Como norma siempre se busca que el número de ejemplos sea mayor que el número de características.

Para determinar el número de ejemplos y características a recopilar en la base de datos del presente proyecto, se tomaron en cuenta los siguientes aspectos:

1. El número de características viene dado por la cantidad de información que puede ser recopilada de cada sujeto de prueba, en este caso las

señales entregadas por los sensores de fuerza y las medidas entregadas por los sensores de movimiento inercial. Previamente se estableció el uso de 8 sensores de fuerza y dos sensores de movimiento inercial, por tal motivo el número de características será:

- a) 8 voltajes entre 0 y 5V, los cuales corresponderán a las señales registradas en cada uno de los terminales de los sensores de fuerza
- b) 4 medidas de ángulos, los cuales se extraerán de la información suministrada por los sensores de movimiento inercial, estos ángulos corresponderán a la posición espacial de dichos sensores, específicamente su rotación en los ejes X & Y

2. Para determinar el número de ejemplos se tomaron en cuenta varios factores:

- a) Primero se definió el número de clases diferentes que debía ser capaz de identificar el algoritmo. En esta etapa se planteó limitar el estudio a 4 agarres y 2 posiciones diferentes, lo cual dio como resultado 8 clases diferentes.
- b) Se consideró conveniente realizar varios ensayos en cada sujeto de prueba, esto con el ánimo de validar el efecto que se produce al retirar el brazalete del sujeto y posteriormente instalarlo de nuevo, lo cual debería ser una tarea recurrente durante el uso de una prótesis. Desde luego esto introduce en el experimento una nueva variable la cual es implícita a los datos registrados y debería ser ajustada por el algoritmo de clasificación.
- c) Uno de los estudios que fue referencia para el presente trabajo corresponde al desarrollado por Erina Cho, en su trabajo reco-

pilaron las diferentes medidas cada 100ms en periodos de 3s (30 muestras por cada registró). Tomando estos valores como referencia, y buscando permitir una fácil adaptación del sistema a futuro, por ejemplo para combinar estos registró con información de electromiografía (desde luego esto requeriría varios ajustes), se consideró adecuado utilizar un tiempo de muestreo de 40ms y registró s de 2s (50 muestras por cada registró), ya que el sistema implementado así lo permite[23].

- d) Para determinar el número de sujetos de prueba se tomó como referencia el estudio realizado por Vikram y Castellini. el cual fue desarrollado con una población de 10 sujetos. También se consideró apropiado este valor dado que las condiciones de salud pública limitaban el acceso a sujetos de prueba para los diferentes ensayos, por lo cual se debía recurrir únicamente a familiares cercanos al autor del presente trabajo [4].

- e) Tomando en cuenta el número de clases a identificar (8), la cantidad de sujetos (10), el tiempo de muestreo (40ms) y el tiempo de registró de cada conjunto de muestra (2s), se estimó que 30 registró s sería un valor adecuado para cada combinación agarreposición (es decir 1500 muestras por cada clase, lo que equivale a 12000 muestras en total por cada ensayo-sujeto), estas permitirían construir una base de datos adecuada en un tiempo razonable, adecuada en el sentido de que se podría llegar a realizar estudios individuales con cada sujeto de prueba, y razonable tomando en cuenta las limitantes de tiempo disponible con cada sujeto (en las condiciones de salud pública actual) y el tiempo proyectado para llevar a cabo esta labor en el presente proyecto.

5.1.1. Posiciones y agarres

A continuación se presenta de manera gráfica los diferentes agarres y posiciones que fueron abordados en el presente estudio. Agarres: Neutral, Pinza, Fuerte, Palma Abierta. Posiciones: Ángulo recto, Totalmente horizontal

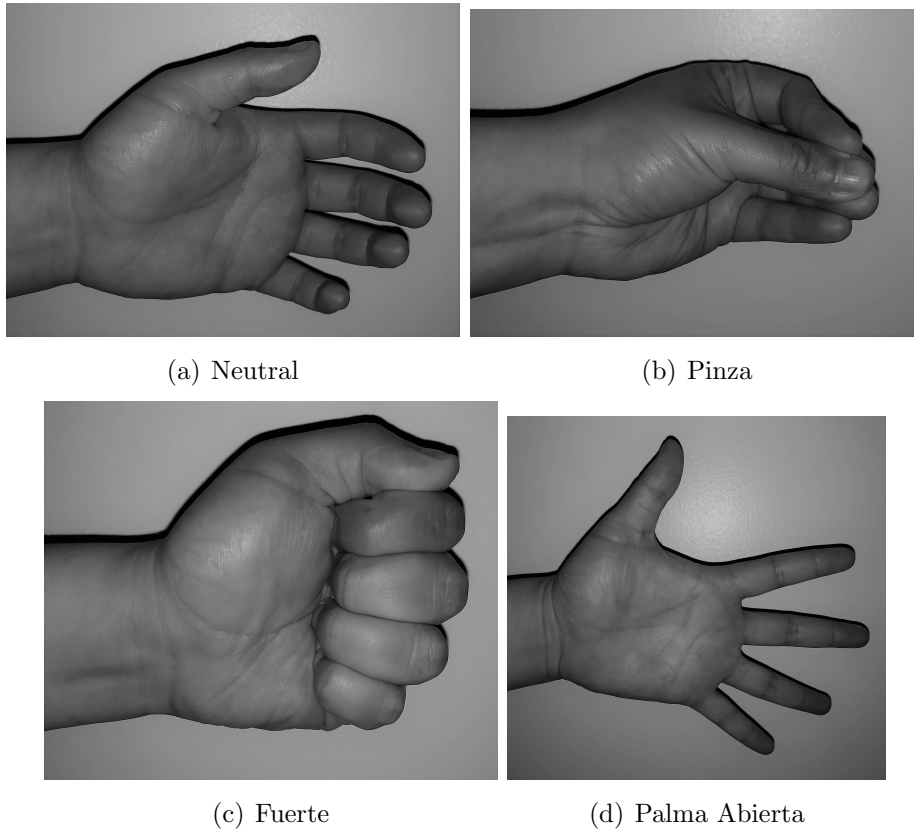
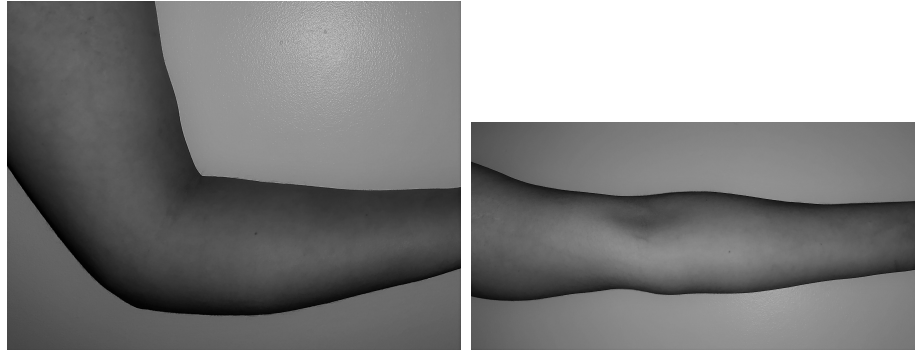


Figura 5-1.: Agarres



(a) Ángulo Recto

(b) Posición Totalmente horizontal

Figura 5-2.: Posiciones

Clase	Posicion	Agarre
1	Ángulo Recto	Neutral
2		Pinza
3		Fuerte
4		Palama Abierta
5	Totalmente Horizontal	Neutral
6		Pinza
7		Fuerte
8		Palama Abierta

Tabla 5-1.: Clases a Registrar

5.1.2. Sujetos de Prueba

Todos los sujetos de prueba corresponden a voluntarios sanos, con edades comprendidas entre los 18 y 64 años. La restricción que produce realizar la validación únicamente con sujetos sanos, es un factor que debería tenerse en cuenta en futuras extensiones del presente proyecto. A continuación, se

relaciona la edad y género de los sujetos de prueba que participaron en el proyecto.

Voluntarios	
género	Edad
Femenino	29
Masculino	26
Masculino	64
Masculino	32
Masculino	18
Masculino	59
Masculino	30
Femenino	60
Femenino	58
Femenino	25

Tabla 5-2.: Voluntarios Sanos

A todos los sujetos prueba se les informo el motivo del estudio, procedimiento y tiempo estimado se diligencia formato de Consentimiento.

Se adjunta como Anexo B los formatos de consentimiento correspondientes a los diez sujetos de prueba.

Fecha: 21-06-2020

Lugar: Bogotá

Entidad: Universidad Antonio Nariño

Investigador: Jaisson Hernando Forero Guerrero

PROPOSITO DEL ESTUDIO

Usted ha sido invitado a participar del proyecto denominado: *Identificación De La Intencionalidad Del Movimiento Del Miembro Superior Mediante El Uso De Sensores De Fuerza Y Sensores De Movimiento Inercial*.

El propósito de este estudio es el registro se señales captadas del miembro superior derecho, mediante el uso de un brazalete que reconocerá los cambios de presión sobre la superficie del antebrazo en el momento que realice cuatro agarres. De igual manera identificará dos posiciones preestablecidas por el investigador.

Durante el desarrollo de la actividad el investigador informará y dará las respectivas indicaciones para la adecuada ejecución de este. El estudio consta de 4 ensayos, cada ensayo tiene una duración aproximadamente de entre 20 y 30 minutos, entre ensayo y ensayo es necesario el retiro del brazalete y un descanso de el menos 10 minutos (tiempo estimado aproximadamente 3 horas).


Se informa que las técnicas y métodos utilizados en los experimentos no son de carácter invasivo y no atentan en contra de la salud de ningún voluntario.

CONSENTIMIENTO

Yo, H. Gonzalo Forero G identificado con la cedula de ciudadanía número 1018463800, manifiesto que he sido informado de la naturaleza y propósito del proyecto *Identificación De La Intencionalidad Del Movimiento Del Miembro Superior Mediante El Uso De Sensores De Fuerza Y Sensores De Movimiento Inercial*, por lo tanto, a través de este documento, autorizo mi participación.

Finalmente, manifiesto que participo voluntariamente y entiendo que es un procedimiento no invasivo.

Firma de Investigador,


 Nombre: Jaisson Hernando Forero Guerrero
 CC: 1.018.413349

Firma participante,

46 
 Nombre:
 CC: 1018463800

Figura 5-3.: Formato Consentimiento

5.1.3. Resumen

Se fijaron parámetros estándar de medida para cada uno de los diez sujetos de prueba:

- A cada sujeto de prueba se le practico cuatro ensayos.
- En cada ensayo se evaluó treinta registros por cada una de las ocho clases.
- Cada registro capturo un total de cincuenta muestras.

Se relaciona el numero de muestras registradas:

- **12.000** muestras por voluntario por cada ensayo.
- **48.000** muestras por voluntario por los cuatro ensayos.
- **60.000** muestras por cada clase.
- **480.000** muestras por toda la base de datos.

Ensayo	registró s	Clases
1	1	1
		2
	
		8
	2	1
		2
	
		8
	1
		2
	
		8
	30	1
		2
	
		8

Tabla 5-3.: Relación Ensayo, registró y Clases

5.2. Registró de información en el computador

Durante el procedimiento de registró , la información es enviada desde el microcontrolador al computador para su almacenamiento, procesamiento y visualización, este proceso es llevado a cabo mediante varios algoritmos *Pre-gistró AT* y una aplicación de usuario desarrollada en Matlab *Pregistró MT*.

5.2.1. Algoritmo desarrollado

A continuación se describe la secuencia de pasos que ejecuta la aplicación de Matlab durante el proceso de registró de información.

- Para iniciar el registro de datos, es necesario que el usuario presione el botón ‘Inicio’ en la aplicación de Matlab *Registró MT*, en ese momento el computador procede a enviar el número decimal ‘0’ al microcontrolador empleando comunicación serial.
- Cuando el microcontrolador recibe este valor, se activa un temporizador el cual se encarga de coordinar todo el proceso y garantiza el tiempo de muestreo de las diferentes señales. Exactamente cada 40ms el temporizador activa la comunicación con los sensores inerciales (empleando el protocolo I2C) e inicia el registro de las señales analógicas provenientes de los sensores de fuerza (a través del ADC), en ese mismo momento procede a enviar al computador la información recopilada durante el ciclo de medida anterior, esto para garantizar la conformación adecuada de los paquetes de información y evitar la colisión o pérdida de datos. Los detalles de la conformación del paquete de datos y demás aspectos del algoritmo implementado en el microcontrolador, pueden ser consultados en el capítulo 4.
- La aplicación de Matlab *Registró MT* despliega dos ventanas emergentes: La primera ventana visualiza los datos On line de los ocho FSR y las dos IMU. Para obtener estos datos es necesario procesar la información suministrada por el microcontrolador, aquí se concatenan y/o convierten los valores recibidos (tipo ‘unsigned char’) de acuerdo a las necesidades del proceso (en unos casos pasan a ser ‘unsigned int’ y en otros casos ‘int’ o ‘double’). También se procede a emplear el filtro complementario para calcular los ángulos que determinan la posición espacial de las IMU, esto a partir de los datos de aceleración y velocidad angular registrados. En la segunda ventana se concibe un formulario el cual es diligenciado con los datos del sujeto de prueba. De igual manera en esta ventana se encuentra el botón *Grabar Datos a*

través del cual se le indica al microcontrolador y a la propia aplicación de Matlab, que debe proceder a almacenar los datos registrados.

- Cuando se oprime el botón *Grabar datos* el computador envía el número decimal '10' al microcontrolador para que este conozca que a partir de ese momento el computador empezará a grabar los datos registrados durante un periodo de 2 segundos. En ese momento el microcontrolador cambia dos de los bytes de la cabecera del paquete de tal forma que en la base de datos pueda identificarse esta situación (algo como: *en proceso de grabación*) y de ser el caso realizar un proceso de validación.
- Cada vez que se presiona el botón *Grabar Datos* se generan varios archivos, en un archivo tipo CSV quedan almacenados los datos provenientes del microcontrolador (sin procesar), en otro archivo CSV quedan almacenados los datos procesados (este incluye las medidas del ADC, los ángulos calculados a partir de las IMU, y la información de agarre y posición suministradas por el operario de la aplicación durante el desarrollo de la prueba), y un 3er archivo, en este caso tipo TXT donde quedan almacenados en formato texto los parámetros de la medida y del sujeto de prueba (Nombre, Edad, género, No. de la Medida, Posición, Agarre etc). Todos los archivos son almacenados empleando una notación estándar en su nombre, esto con el objetivo de permitir una fácil manipulación mas adelante.

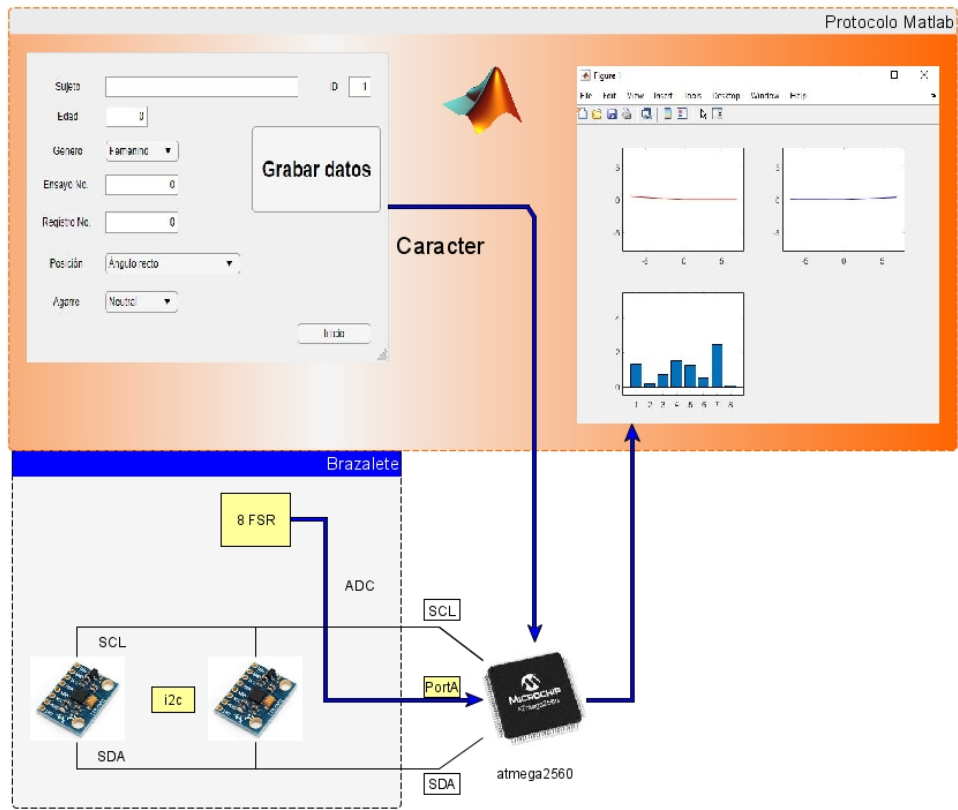


Figura 5-4.: Protocolo Matlab

5.2.2. Descripción del procedimiento práctico

Se adjunta como Anexo C el Algoritmo Pregistró MT

Se estableció un procedimiento estándar para la ejecución de los ensayos y el registró de datos.

Identificación de sensores IMU y FSR



Figura 5-5.: Identificación Sensores FSR

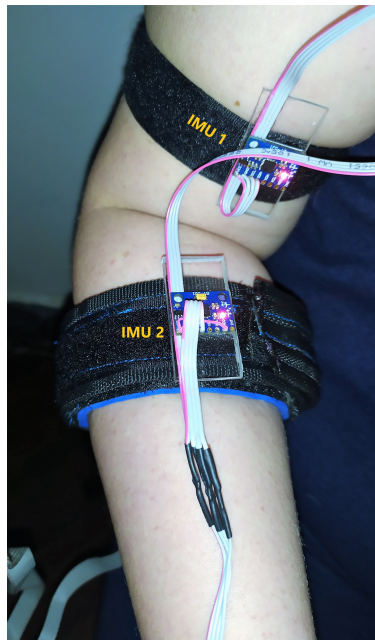


Figura 5-6.: Identificación Sensores IMU

En primer instancia es necesario identificar los sensores de fuerza (FSR) y los Sensores de movimiento inercial (IMU)

Identificación y visualización de sensores FSR e IMU en la aplicación Pregistró MT

De igual manera se identifican los sensores IMU y FSR al ejecutar la aplicación. *Pregistró MT*

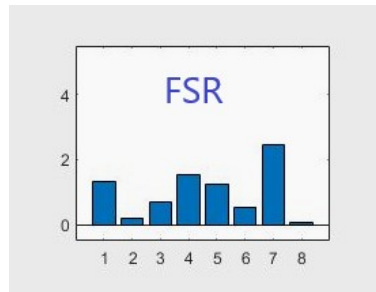


Figura 5-7.: Visualización Sensores FSR

Al ejecutar la aplicación en las barras azules de visualiza la fuerza ejercida sobre cada uno de los ocho sensores FSR, Donde el FSR1 sera la barra etiquetada como 1.

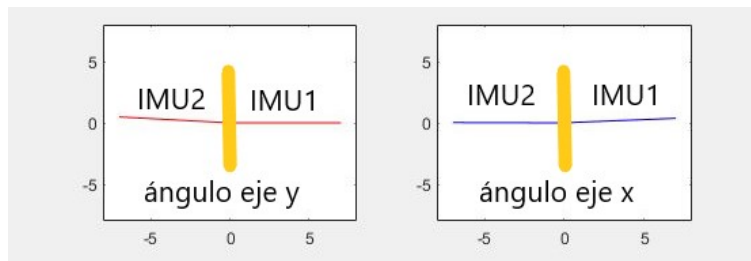


Figura 5-8.: Visualización Sensores IMU

Por otra parte, se visualiza los ángulos de rotación en los ejes X y Y entregados por los sensores IMU. la línea Horizontal de color rojo representa los ángulos de rotación de las IMU1 e IMU2 sobre el eje Y; De igual manera la línea horizontal azul representa los ángulos de rotación de las IMU1 e IMU2 sobre el eje X.

Ubicación del brazalete en el brazo

Dado que el brazalete esta conformado por dos módulos, se ubica primero el que contiene los FSR.

Se solicito a los sujetos de prueba que se ubicaran de pie con el brazo totalmente relajado y el empeine de la mano mirando de frente. El Sensor FSR1 fue utilizado como referencia en la ubicación del brazalete y en el ajuste sobre el antebrazo.



(a) Ubicación FSR1 (b) Posición del brazalete en el Antebrazo

Figura 5-9.: Ubicación Brazalete

El FSR1 se ubico bajo la articulación que une el brazo con el antebrazo, los demás sensores FSR se ubicaron de acuerdo a la separación dada por el brazalete. De igual manera se utilizo la medición de fuerza en el sensor FSR1 para el ajuste del brazalete sobre el antebrazo.

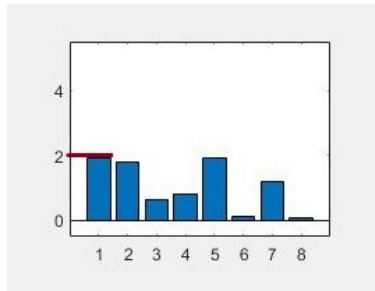


Figura 5-10.: FSR1 referencia ajuste brazalete

Se busco que una vez puesto el brazalete en el brazo totalmente relajado y el empeine de la mano mirando de frente, la medida del FSR1 estuviera lo mas cercano al valor dos, de acuerdo a lo observado en la aplicación *Pregistró MT*.

En esta etapa el módulo uno del brazalete esta ubicado en el antebrazo en una posición y ajuste estándar.

Paso Siguiente, posicionar los sensores IMU1 y IMU2.

Se coloco el módulo dos del brazalete a la altura del bíceps, con el sujeto sentado se le dio indicaciones de mantener la posición fija en ángulo recto, plano sagital, agarre neutro.

En esta posición y agarre se ubicarán los sensores IMU1 y IMU2, garantizando que los ejes de rotación sobre el eje X y que son representados en la

aplicación *Pregistró MT* por las líneas azules sean aproximadamente de cero.

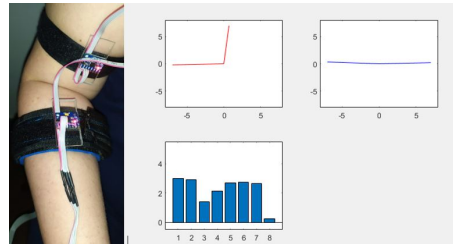


Figura 5-11.: Sensores IMU

Ejecución de la aplicación *Pregistró MT*

Cuando se ejecuta la aplicación *Pregistró MT* se abren dos ventanas, la ventana donde se visualizan los datos de los sensores IMU y FSR y una ventana que contiene un formulario y el botón *Grabar datos*.

Figura 5-12.: Formulario

Los datos del formulario indican el ID del sujeto de prueba, la edad, género, numero de ensayo, numero de registró , posición y agarre; Estos datos se utilizaron para la construcción del nombre del archivo en el que se guardaron las muestras registradas.

Después de diligenciar el Formulario se procede a oprimir el botón *Grabar datos*, en este momento se guardo en el computador un archivo de extensión .CSV que contine cincuenta muestras.

5.2.3. Descripción de los archivos sin procesar

Al terminar los cuatro ensayos, fueron generados 960 archivos por sujeto, 9600 archivos en toda la base de datos y dado a que cada archivo contiene 50 muestras, en total se obtuvieron 480.000 muestras. No obstante, esos datos están sin procesar se implementa el algoritmo de Matlab *Procesamiento* el cual computa el contenido de los archivos del Raw Data y crea nuevos archivos para conformar la base de datos.

Cada uno de los archivos del Raw Data contiene una matriz de 52 columnas.

Se adjunta como Anexo D el Raw Data

0,0,0,0,95,0,32,236,51,0,244,224,251,236,255,96,255,157,37,16,30,4,103,152,244,0,254,0,255,98,0,16,1,71,1,146,0,97,0,196,1,111,1,144,0,222,0,14,48,32,255,255
0,0,0,0,94,140,31,188,51,176,245,0,251,212,255,86,255,72,36,252,30,164,103,248,244,32,255,1,255,55,0,90,1,75,1,145,0,96,0,196,1,108,1,142,0,224,0,17,48,33,255,255

Figura 5-13.: Primeras dos filas, Primer Archivo Raw Data

5.2.4. Descripción de archivos procesados

Se realizo un procesamiento al Raw Data para obtener la base de datos, para ello se implemento el algoritmo en Matlab *Procesamiento*.

Se adjunta como Anexo E el algoritmo Procesamiento

La base de datos procesada esta compuesta por 9600 archivos un total de 480000 muestras por todos los sujetos de prueba.

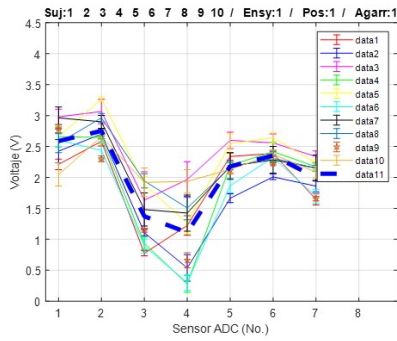
2,54	-76,99	-3,82	1,06	1,60	1,96	0,47	0,96	1,79	1,96	1,09	0,07	1,00	2,00	2,00
1,50	-76,47	-3,25	1,12	1,62	1,96	0,47	0,96	1,78	1,95	1,09	0,08	1,00	2,00	2,00

Figura 5-14.: Primeras dos filas, Primer Archivo Base Datos

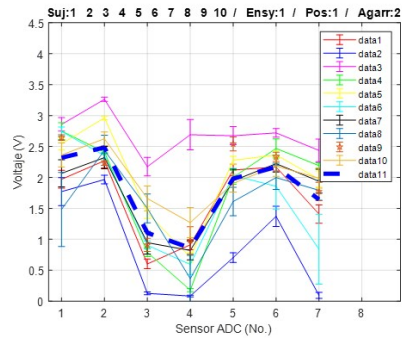
Se adjunta como Anexo F la base de datos

Análisis gráfico de datos de los sensores FSR de los diez sujetos de prueba en el ensayo numero 1, Posición 1 Agarres 1 2 3 4

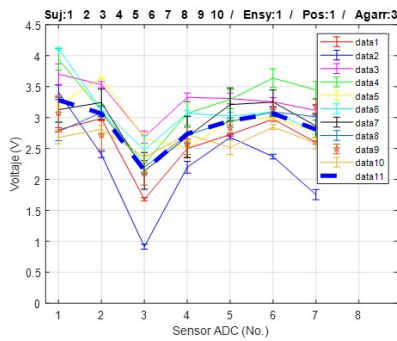
Se realizo un análisis con base en las muestras registradas de los diez sujetos de prueba, para el ensayo numero 1, posición 1 agarres 1 2 3 4. por lo tanto se analizaron 6000 muestras.



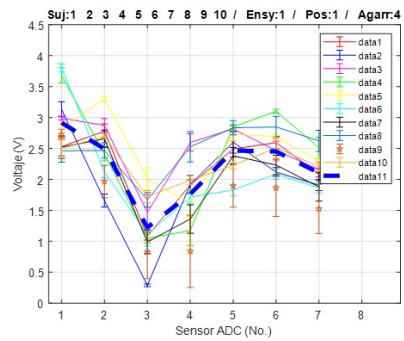
(a) Agarre 1



(b) Agarre 2



(c) Agarre 3

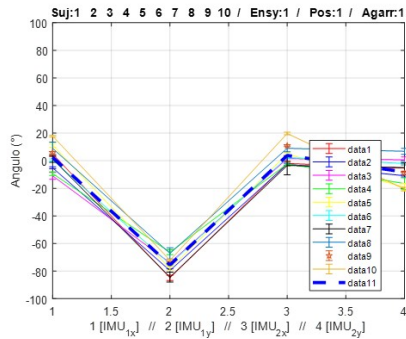


(d) Agarre 4

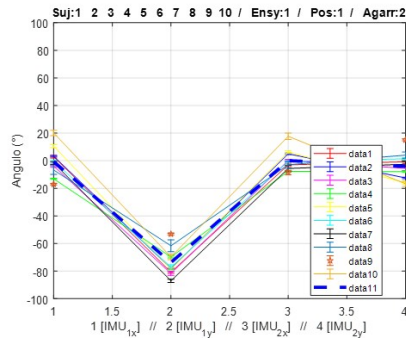
Figura 5-15.: Datos FSR ensayo 1 Posición 1 agarres 1 2 3 4

En las gráficas se representa el promedio de la muestras de los sensores FSR para cada sujeto de prueba y la varianza entre voluntarios.

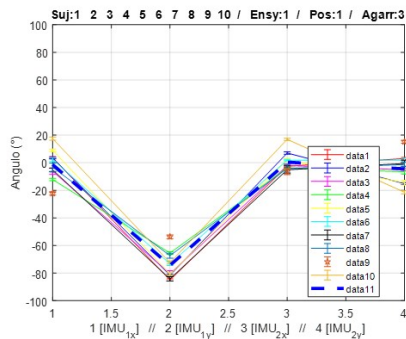
Se observa. Según las líneas de tendencia un comportamiento particular para cada agarre.



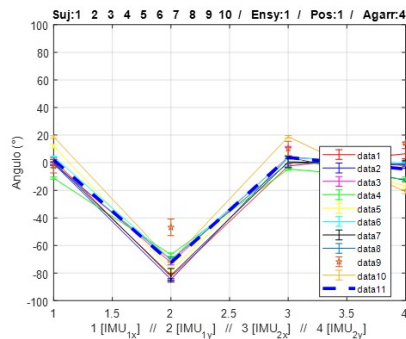
(a) Agarre 1



(b) Agarre 2



(c) Agarre 3



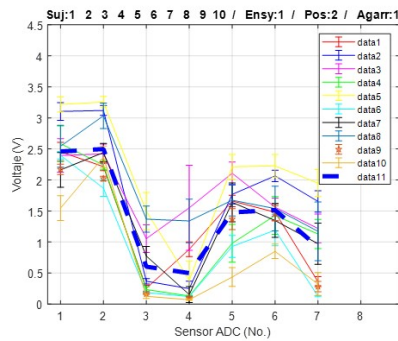
(d) Agarre 4

Figura 5-16.: Datos IMU ensayo 1 Posición 1 agarre 1 2 3 4

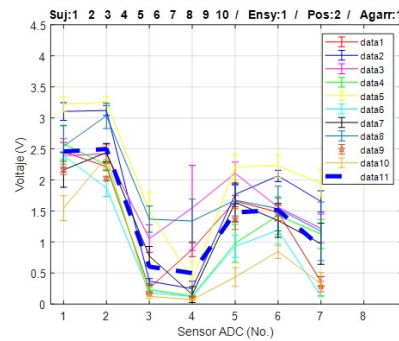
En las gráficas se representa el promedio de la muestras de los sensores IMU para cada sujeto de prueba y la varianza entre voluntarios.

Se observa. Según las líneas de tendencia un comportamiento particular para cada posición. Por otro se visualiza un patrón común en todos los sujetos.

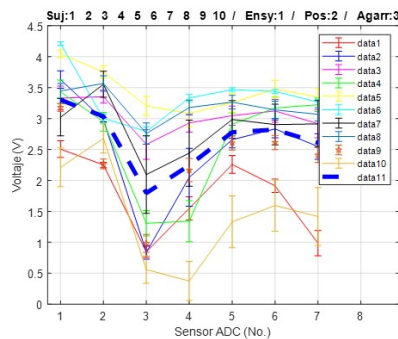
Análisis Gráfico de datos de los sensores FSR de los diez sujetos de prueba en el ensayo numero 1, Posición 2 Agarres 1 2 3 4



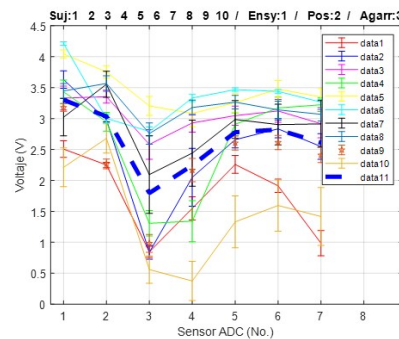
(a) Agarre 1



(b) Agarre 2



(c) Agarre 3



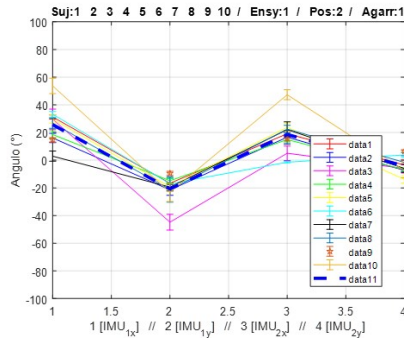
(d) Agarre 4

Figura 5-17.: Datos FSR ensayo 1 Posición 2 agarres 1 2 3 4

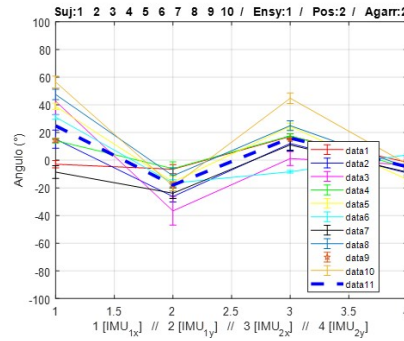
En las gráficas se representa el promedio de la muestras de los sensores FSR para cada sujeto de prueba y la varianza entre voluntarios.

Se observa. Según las líneas de tendencia un comportamiento particular para cada agarre, incluso si se compara con los mismos agarres en la posición uno.

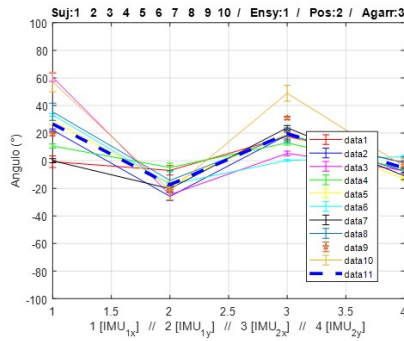
Por otro lado se visualiza una mayor dispersión de datos entre sujeto y sujeto, este comportamiento puede darse ante la complejidad de ejecutar la posición dos por determinado tiempo.



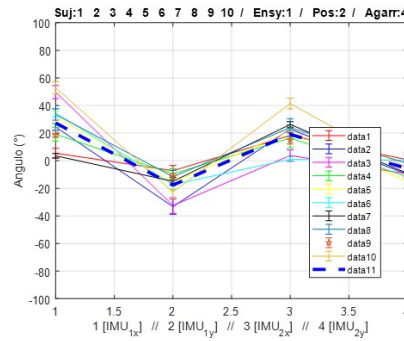
(a) Agarre 1



(b) Agarre 2



(c) Agarre 3



(d) Agarre 4

Figura 5-18.: Datos IMU ensayo 1 Posición 2 agarres 1 2 3 4

En las gráficas se representa el promedio de la muestras de los sensores IMU para cada sujeto de prueba y la varianza entre voluntarios.

Se observa. Según las líneas de tendencia un comportamiento particular para la segunda posición.

No obstante se visualiza una mayor dispersión de los datos entre sujeto y sujeto en comparación con los datos de la posición uno, este comportamiento puede darse ante la complejidad de mantener la posición dos por determinado tiempo.

Se adjunta como Anexo E el análisis gráfico correspondiente al ensayo 1 Posición 1 y 2 para los 10 sujetos y los agarres 1234 (sensores de Fuerza e Inerciales

Se adjunta como Anexo F el análisis gráfico correspondiente al ensayo 1 Posición 2 para los 10 sujetos y los agarres 1234 (sensores de Fuerza e Inerciales

Se adjunta como Anexo G el análisis gráfico correspondiente a los ensayos 1 2 3 4 Posición 1 para los 10 sujetos y agarre 1 (sensores de Fuerza e Inerciales

Se adjunta como Anexo H el análisis gráfico correspondiente a los ensayos 1 2 3 4 Posición1 para los 10 sujetos y agarre 1 234 (sensores de Fuerza e Inerciales

Se adjunta como Anexo H el análisis gráfico correspondiente a los ensayos 1 2 3 4 Posición1 para los 10 sujetos y agarre 1 234 (sensores de Fuerza e Inerciales

6. Algoritmos De Clasificación

6.1. Descripción Pruebas realizadas

Se implementaron algoritmos de Regresión Logística One Vs All. De igual manera se ejecutaron algunas herramientas de clasificación pre-construidas en Matlab, en específico el método SVM Cuadrática .

Se tuvieron en cuenta varios escenarios basándose en la base de datos obtenida durante el desarrollo del proyecto.

En los numerales 6.2.1; 6.2.2; 6.2.3 se ejecutaron validaciones incluyendo solamente el algoritmo de regresión logística One Vs All.

En los numerales 6.2.4; 6.2.5; 6.2.6; 6.2.7 se ejecuto el algoritmo de regresión logística One Vs All en comparación con el método SVM.

Los diferentes escenarios ejecutados están descritos en cada numeral junto con los resultados.

Se adjunta como Anexo K los algoritmos de clasificación de Regresión Logística.

6.2. Regresión Logística One VS All

6.2.1. Método Clasificación One Vs All Regresión Logística, un ensayo, una posición, cuatro agarres, para cada sujeto.

De la base de datos se extrajeron los datos correspondientes a un ensayo, una posición, cuatro agarres.

El conjunto de entrenamiento se construyó aleatoriamente por el setenta por ciento de las muestras seleccionadas para un total de 4200 elementos, 12 características, 4 correspondientes a señales de los dos sensores de movimiento inercial (IMU) y 8 correspondientes a los sensores de fuerza (FSR). El algoritmo predecirá cuatro clases o etiquetas.

$m=4200$; Conjunto de Validación= 1800 ; $n=12$; Clases= 4 , conformadas por la posición Ángulo recto con los cuatro agarres.

Se relaciona la exactitud y error del algoritmo en el conjunto de entrenamiento y en el conjunto de Validación.

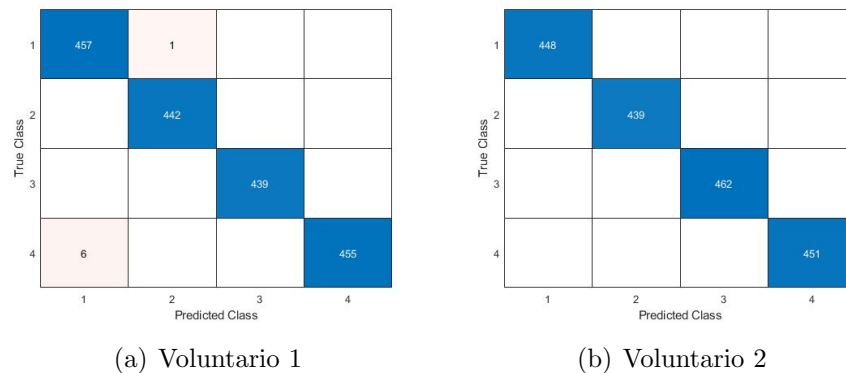


Figura 6-1.: Matriz de confusión C1 S1yS2

Clasificador 1				
2*Voluntario	Conjunto de Entrenamiento		Conjunto de Validación	
	Exactitud	Error	Exactitud	Error
1	99,21	0,79	99,61	0,39
2	100	0	100	0
3	99,66	0,34	99,61	0,39
4	100	0	100	0
5	96,33	3,67	96,16	3,84
6	100	0	100	0
7	99	1	98,83	1,17
8	100	0	100	0
9	100	0	100	0
10	98,95	1,05	99,22	0,78

Tabla 6-1.: Relación Exactitud y Error Clasificador 1

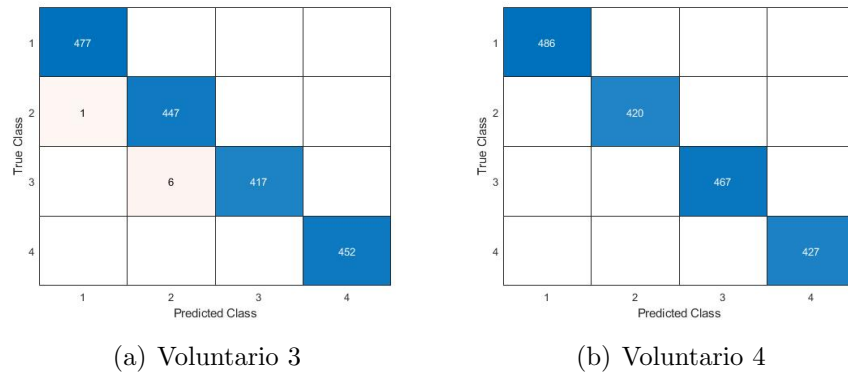
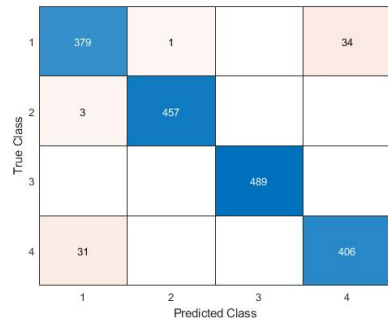
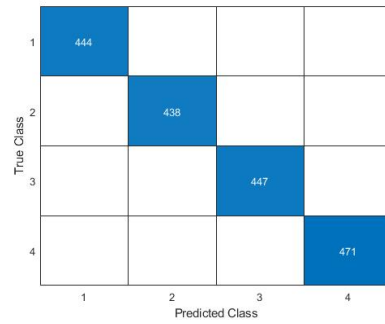


Figura 6-2.: Matriz de confusión C1 S3yS4

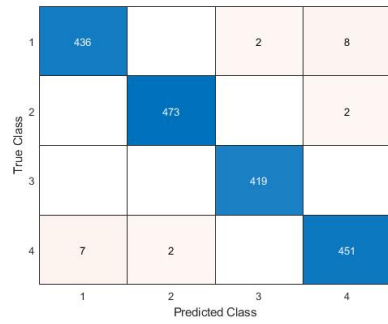


(a) Voluntario 5

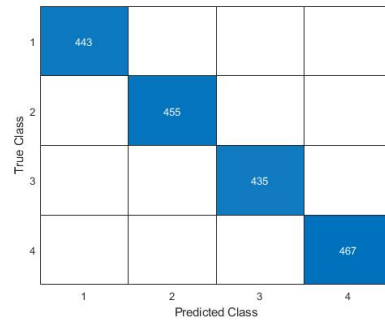


(b) Voluntario 6

Figura 6-3.: Matriz de confusión C1 S5YS6



(a) Voluntario 7



(b) Voluntario 8

Figura 6-4.: Matriz de confusión C1 S7YS8

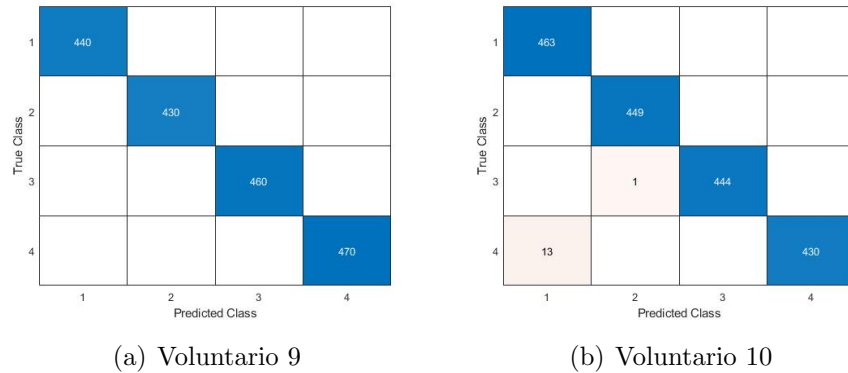


Figura 6-5.: Matriz de confusión C1 S9YS10

Según la tabla 6-1 y observando las matrices de confusión obtenidas del conjunto de validación, se aprecia que manejando una sola posición, con cuatro agarres y un solo ensayo el valor de la exactitud es alta.

6.2.2. Método Clasificación One Vs All Regresión Logística, cuatro ensayos, dos posición, cuatro agarres, para cada sujeto con IMU.

De la base de batos se extrajo los datos correspondientes a cuatro ensayos, dos posiciones, cuatro agarres.

El conjunto de entrenamiento se construyó aleatoriamente por el setenta por ciento de las muestras seleccionadas para un total de 33600 elementos, 12 características, 4 correspondientes a señales de los dos sensores de movimiento inercial (IMU) y 8 correspondientes a los sensores de fuerza (FSR). El algoritmo predecirá ocho clases o etiquetas.

m=33600; Conjunto de Validación=14400; n=12; Clases=8, conformadas

por las dos posiciones con los cuatro agarres.

Se relaciona la exactitud y error del algoritmo en el conjunto de entrenamiento y en el conjunto de Validación.

Clasificador 2 con IMU				
Voluntario	Conjunto de Entrenamiento		Conjunto de Validación	
	Exactitud	Error	Exactitud	Error
1	93,73	6,27	93,95	6,05
2	92,57	7,43	92,38	7,62
3	88,28	11,72	88,49	11,5
4	94,15	5,85	93,88	6,12
5	94,81	5,19	94,68	5,32
6	99,82	0,18	99,75	0,25
7	88,93	11,07	88,7	11,3
8	93,3	6,7	93,38	6,62
9	94,51	5,49	94,75	5,25
10	87,64	12,36	87,39	12,6

Tabla 6-2.: Relación Exactitud y Error Clasificador 2

1	1767	13		37				
2	25	1754						
3	41	5	1706	16				
4	55	1	297	1465				
5	8			1684	11	10	46	
6	1			11	1739			
7				22	63	1657	75	
8				47	24	62	1758	
	1	2	3	4	5	6	7	8

(a) Voluntario 1

1	1744	15	11					
2		1779	16	13				
3	3	10	1703	57			3	
4		76	48	1688				
5				1778	7	2	15	
6			8		3	1740		18
7			4			6	1333	470
8				6	146	19	141	1538
	1	2	3	4	5	6	7	8

(b) Voluntario 2

Figura 6-6.: Matriz de confusión C2 S1yS2

1	1565	62	95	97	3	4		
2	100	1540	34	144	17	1		
3	24	5	1766	17				
4	393	169		1198				
5				1705	29		42	
6	4	7	1	142	1566	20	80	
7				1	10	14	1728	1
8				31	21	89	1675	
	1	2	3	4	5	6	7	8

(a) Voluntario 3

1	1758	79		1				
2	64	1758		9	2			
3			1823	1				
4	2	29		1732			3	
5	1	1		1548	220		59	
6				242	1463	6	45	
7						1764	5	
8				89	16	6	1674	
	1	2	3	4	5	6	7	8

(b) Voluntario 4

Figura 6-7.: Matriz de confusión C2 S3yS4

1	1598	56	4	103				
2	19	1725		49				
3	10	6	1822	5				
4	25	30	38	1731		2		
5		1			1669	52	2	74
6		1			46	1685		32
7							1785	
8				2	143	66		1619
	1	2	3	4	5	6	7	8

(a) Voluntario 5

1	1791	4						
2	14	1786						
3			1801					
4				1858				
5					1776	4		
6					14	1789		
7							1800	
8								1763
	1	2	3	4	5	6	7	8

(b) Voluntario 6

Figura 6-8.: Matriz de confusión C2 S5yS6

1	1654	29	15	49				16
2	4	1749	6	15				
3	18	15	1620	144				18
4	37	10	38	1675				
5	9				1355	321	24	110
6	5	1		1	474	1354		8
7	2				16	6	1671	95
8					85	34	22	1695
	1	2	3	4	5	6	7	8

(a) Voluntario 7

1	1723	53		14		2		
2	132	1610	1	9				
3			1742	19				16
4	35	72	49	1679				1
5	13			3	1517	188	12	52
6				7	130	1605		59
7			13				1821	19
8					3	44	4	1750
	1	2	3	4	5	6	7	8

(b) Voluntario 8

Figura 6-9.: Matriz de confusión C2 S7yS8

1	1733	8		57				
2	23	1783		23		2		
3			1811	2				
4	299			1482	1	6		
5					1772	26		6
6		3			94	1605		44
7	3						1823	
8					48	110		1636
	1	2	3	4	5	6	7	8

(a) Voluntario 9

1	1732	26	28	9				
2	31	1682	1	118				
3	28	18	1685	39				
4	33	89	27	1703				
5	7			1	1369	265	40	109
6		2			227	1358		194
7					13		1765	31
8					190	203	86	1291
	1	2	3	4	5	6	7	8

(b) Voluntario 10

Figura 6-10.: Matriz de confusión C2 S9yS10

Según la tabla 6-2 y observando las matrices de confusión las dos posición, con cuatro agarres y cuatro ensayos el valor de la exactitud es disminuye respecto al clasificador 1. Dado a factores como el aumento significativo de muestras, y a desviaciones causadas por los intervalos entre ensayo y ensayo.

6.2.3. Método Clasificación One Vs All Regresión Logística, cuatro ensayos, dos posición, cuatro agarres, para cada sujeto sin IMU.

De la base de batos se extrajo los datos correspondientes a cuatro ensayos, dos posiciones, cuatro agarres.

El conjunto de entrenamiento se construyó aleatoriamente por el setenta por ciento de las muestras seleccionadas para un total de 33600 elementos, 8 características correspondientes a los 8 sensores de fuerza (FSR). El algoritmo predecirá ocho clases o etiquetas.

m=33600; Conjunto de Validación=14400; n=8; Clases=8, conformadas por

las dos posiciones con los cuatro agarres.

Se relaciona la exactitud y error del algoritmo en el conjunto de entrenamiento y en el conjunto de Validación.

Clasificador 3 sin IMU				
Voluntario	Conjunto de Entrenamiento		Conjunto de Validación	
	Exactitud	Error	Exactitud	Error
1	85,83	14,17	85,46	14,5
2	88,81	11,19	88,75	11,3
3	66,58	33,42	66,47	33,5
4	92,27	7,73	91,44	8,56
5	85,69	14,31	85,54	14,5
6	98,49	1,51	98,45	1,55
7	85,89	14,11	85,86	14,1
8	78,03	21,97	77,88	22,1
9	81,73	18,27	81,55	18,5
10	76,69	23,31	77,312	22,7

Tabla 6-3.: Relación Exactitud y Error Clasificador 3

1	1496	105		170			6	19	
2	218	1523		2				12	
3	12	7	1625	139	1		81		
4	144	20	673	923			61	2	
5		9	8		1735	10	2	37	
6		1			28	1768	1	1	
7			7	33	9	88	1609	21	
8		13	14	1	106	8	24	1628	
		1	2	3	4	5	6	7	8

(a) Voluntario 1

1	1666	98	15			5	3	14	
2	17	1479	8	77		101		92	
3	4	12	1775	51			1		
4		25	41	1700					
5	5				1764	3		17	
6	5	30	16	1	2	1747		12	
7		7	6				1123	668	
8		17		9	139	5	113	1527	
		1	2	3	4	5	6	7	8

(b) Voluntario 2

Figura 6-11.: Matriz de confusión C3 S1yS2

1	1141	118	62	95	39	83	169	95	
2	84	832	42	183	79	211	371	18	
3	39	110	1512	7	4		93		
4	654	313		786	13	22	13	1	
5	59			4	1655	67		20	
6	18	123	21	14	222	1281	10	113	
7	1	16	440	51	3	2	1014	210	
8	1	296	106	51	14	18	30	1351	
		1	2	3	4	5	6	7	8

(a) Voluntario 3

1	1751	35		28					
2	87	1656		13	55	2			
3			1773				1		
4	4	28		1801				1	
5	4	32			1447	233		138	
6				1	246	1449	8	41	
7		7	1	7			1747	1	
8		5		77	65	36	4	1616	
		1	2	3	4	5	6	7	8

(b) Voluntario 4

Figura 6-12.: Matriz de confusión C3 S3yS4

1	1348	169	16	241	8			
2	372	1125		218	35			14
3	28	2	1741	16			10	
4	80	41	85	1593	2	6		6
5		8		1691	63	9	34	
6		15		3	52	1714		24
7			6	2	3		1791	
8		50		31	386	47		1315
	1	2	3	4	5	6	7	8

(a) Voluntario 5

1	1815	5						
2	35	1800		1		1		
3			1731	37				
4				1811				
5				1742	58			
6		1		84	1690			
7						1769		
8								1820
	1	2	3	4	5	6	7	8

(b) Voluntario 6

Figura 6-13.: Matriz de confusión C3 S5yS6

1	1475	141	34	192			7	
2	12	1760	5	25	2		5	
3	7	26	1608	112			26	
4	74	7	54	1590	1		24	50
5	24	17		1	1298	521	14	29
6	23	39	2		240	1371	4	40
7	28		5	38	22	7	1637	37
8	14			3	53	42	28	1626
	1	2	3	4	5	6	7	8

(a) Voluntario 7

1	1320	256	1	105	91	10		7
2	283	1418	79	36				
3	150	8	1428	88	2		2	79
4	99	90	304	959	218	6	14	164
5	29	2		41	1286	385		23
6	2			1	300	1502		35
7			14	6			1790	23
8			5	158	66	1	1	1513
	1	2	3	4	5	6	7	8

(b) Voluntario 8

Figura 6-14.: Matriz de confusión C3 S7yS8

1	1628	46	1	109				
2	55	1693		28				1
3	3	2	1740	2			19	
4	899	9		884	3		9	
5				1174	642			23
6		14		490	1160			156
7		2	1	5			1787	
8				33	104			1678
	1	2	3	4	5	6	7	8

(a) Voluntario 9

1	1174	632	8	1				16
2	294	1299	8	160				14
3	52	8	1750	32	3			13
4	94	99		1538	1			68
5	6	7		1	970	465	27	237
6	2				198	1495		113
7	15	3		6	28		1737	22
8		10			199	278	147	1170
	1	2	3	4	5	6	7	8

(b) Voluntario 10

Figura 6-15.: Matriz de confusión C3 S9yS10

Según la tabla 6-3 y observando las matrices de confusión para dos posición, con cuatro agarres y cuatro ensayos el valor de la exactitud es disminuye respecto al clasificador 1 y 2. Dado a factores como el aumento significativo de muestras, y a desviaciones causadas por los intervalos entre ensayo y ensayo, adicionado la falta de cuatro características referentes a las IMU.

Comparación entre los clasificadores 2 y 3

Los clasificadores 2 y 3 son de naturaleza similar en cuanto al tamaño de la muestra. No obstante en el clasificador 3 se descarto el uso de las características suministradas por los sensores de movimiento inercial (IMU). Es decir en el clasificador dos se contemplaron 12 características para el entrenamiento y validación mientras que en el clasificador tres, solamente se contemplo ocho características.

Se relaciona la diferencia entre los valores de exactitud de los conjuntos de validación entre los clasificadores dos y tres.

Sujeto	diferencia Exactitud
1	8,49
2	3,63
3	22,02
4	2,44
5	9,14
6	1,3
7	2,84
8	15,5
9	13,2
10	10,078

Tabla 6-4.: Diferencia en la Exactitud del conjunto de validación con relación al uso de los sensores de Movimiento Inercial IMU

Con base en la información de la tabla 6-4 se evidencia que el uso de los sensores de movimiento inercial IMU mejoran la calidad del clasificador.

6.2.4. Comparación Método SVM Cuadrático con Método One Vs All Regresión Logística, cuatro ensayos, una posición, cuatro agarres, tres sujetos.

De la base de datos se extrajeron los datos correspondientes a cuatro ensayos, una posición, cuatro agarres, tres sujetos.

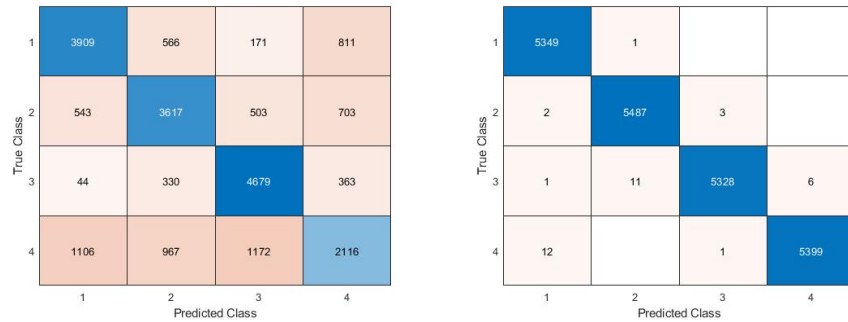
El conjunto de entrenamiento se construyó aleatoriamente por el setenta por ciento de las muestras seleccionadas para un total de 50400 elementos, 8 características correspondientes a los 8 sensores de fuerza (FSR). El algoritmo predecirá cuatro clases o etiquetas.

m=33600; Conjunto de Validación=21600; n=8; Clases=4, conformadas por la posición ángulo recto con los cuatro agarres.

Para ambos algoritmos se utilizaron los mismos conjuntos de entrenamiento y validación. En Matlab mediante las herramientas de clasificación se entrenó el sistema mediante validación cruzada con el algoritmo SVM.

Regresión Logística		SVM Cuadrático	
Exactitud CE	Exactitud CV	Exactitud CE	Exactitud CV
66,33	66,3	99,7	99,82

Tabla 6-5.: Comparación Método SVM Cuadrático con Método One Vs All Regresión Logística, cuatro ensayos, una posición, cuatro agarres, tres sujetos



(a) Matriz de Confusión Regresión Logística (b) Matriz de Confusión SVM

Figura 6-16.: Matriz de Confusión comparación Regresión Logística con SVM

De acuerdo a la información contenida en la tabla 6-5 se evidencia que la exactitud del algoritmo de regresión logística One Vs All presenta disminución dado el aumento de muestras y la inclusión de tres sujetos de prueba. No obstante el algoritmos SVM presenta un óptimo comportamiento.

6.2.5. Comparación Método SVM Cuadrático con Método One Vs All Regresión Logística, cuatro ensayos, dos posiciones, cuatro agarres, tres sujetos con IMU.

De la base de batos se extrajo los datos correspondientes a cuatro ensayos, dos posiciones, cuatro agarres, tres sujetos.

El conjunto de entrenamiento se construyó aleatoriamente por el setenta por ciento de las muestras seleccionadas para un total de 100800 elementos, 12 características, 4 correspondientes a señales de los dos sensores de movi-

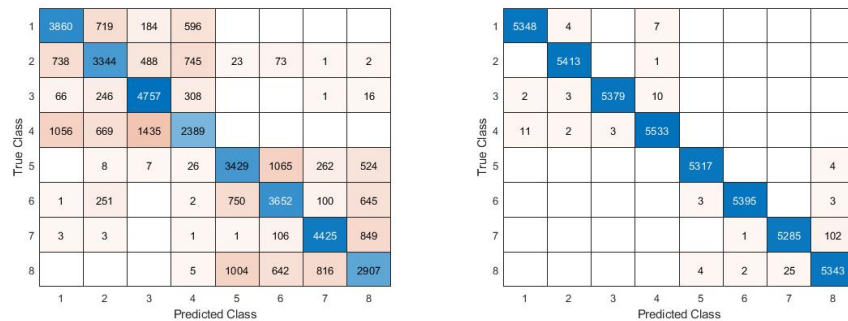
miento inercial (IMU) y 8 correspondientes a los sensores de fuerza (FSR). El algoritmo predecirá ocho clases o etiquetas.

m=100800; Conjunto de Validación=43200; n=12; Clases=8, conformadas por las dos posiciones con los cuatro agarres.

Para ambos algoritmos se utilizo los mismos conjuntos de entrenamiento y validación. En Matalab mediante las herramientas de clasificación se entrena el sistema mediante validación cruzada con el algoritmo SVM.

Regresión Logística		SVM Cuadrático	
Exactitud CE	Exactitud CV	Exactitud CE	Exactitud CV
66,27	65,93	99,4	99,59

Tabla 6-6.: Comparación Método SVM Cuadrático con Método One Vs All Regresión Logística, cuatro ensayos, dos posiciones , cuatro agarres, tres sujetos



(a) Matriz de Confusión Regresión Logística (b) Matriz de Confusión SVM SVM

Figura 6-17.: Matriz de Confusión comparación Regresión Logística con SVM

De acuerdo a la información contenida en la tabla 6-6 se evidencia una mayor eficiencia en el Método SVM Cuadrático en relación con el Método One Vs All Regresión Logística.

6.2.6. Comparación Método SVM Cuadrático con Método One Vs All Regresión Logística, cuatro ensayos, dos posiciones, cuatro agarres, tres sujetos sin IMU.

De la base de datos se extrajo los datos correspondientes a cuatro ensayos, dos posiciones, cuatro agarres, tres sujetos.

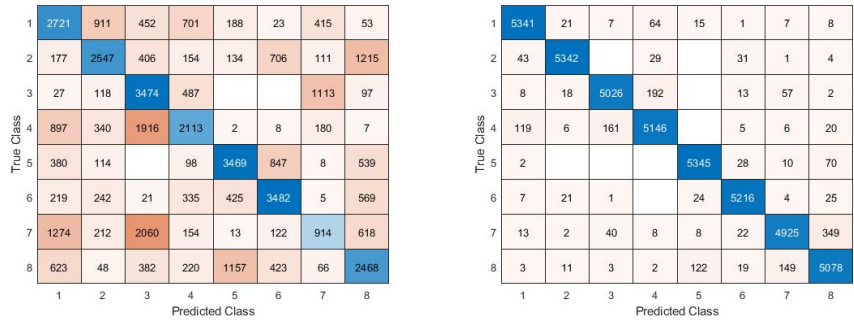
El conjunto de entrenamiento se construyó aleatoriamente por el setenta por ciento de las muestras seleccionadas para un total de 100800 elementos, 8 características correspondientes a 8 sensores de fuerza (FSR). El algoritmo predecirá ocho clases o etiquetas.

$m=100800$; Conjunto de Validación= 43200 ; $n=8$; Clases= 8 , conformadas por las dos posiciones con los cuatro agarres.

Para ambos algoritmos se utilizaron los mismos conjuntos de entrenamiento y validación. En Matlab mediante las herramientas de clasificación se entrenó el sistema mediante validación cruzada con el algoritmo SVM.

Regresión Logística		SVM Cuadrático	
Exactitud CE	Exactitud CV	Exactitud CE	Exactitud CV
49,35	49,04	95,6	95,87

Tabla 6-7.: Comparación Método SVM Cuadrático con Método One Vs All Regresión Logística, cuatro ensayos, dos posiciones , cuatro agarres, tres sujetos, Sin IMU



(a) Matriz de Confusión Regresión Logística (b) Matriz de Confusión SVM SVM

Figura 6-18.: Matriz de Confusión comparación Regresión Logística con SVM

Como se ilustra en la figura 6-7 la ausencia de las IMU afecto de manera considerable el Método de regresión logística One Vs All.

6.2.7. Comparación Método SVM Cuadrático con Método One Vs All Regresión Logística, Tres ensayos de entrenamiento un ensayo de validación , dos posiciones, cuatro agarres, tres sujetos.

De la base de datos se extrajo los datos correspondientes a tres Ensayos , dos posiciones, cuatro agarres, tres sujetos, para el conjunto de entrenamiento.

De igual manera se extrajo los datos correspondientes a un Ensayo , dos posiciones, cuatro agarres, tres sujetos, para el conjunto de validación.

Conjunto de entrenamiento:

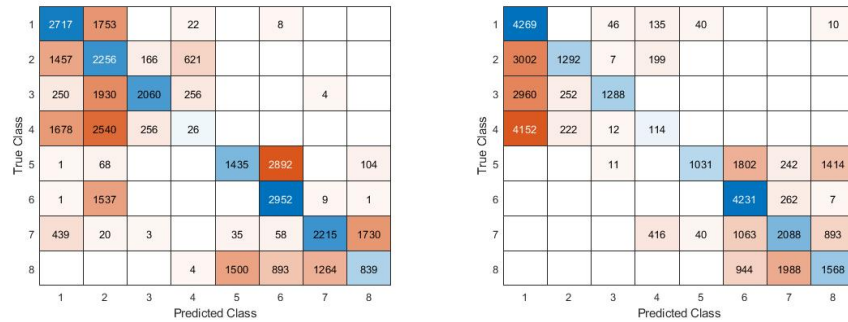
m=144000; n=8; Clases=8, conformadas por las dos posiciones con los cuatro agarres.

Conjunto de Validación:

m=48000; n=8; Clases=8, conformadas por las dos posiciones con los cuatro agarres.

Regresión Logística		SVM Cuadrático	
Exactitud CE	Exactitud CV	Exactitud CE	Exactitud CV
77.59	40.27	99.9	45,87

Tabla 6-8.: Comparación Método SVM Cuadrático con Método One Vs All Regresión Logística, Tres ensayos de entrenamiento un ensayo de validación , dos posiciones, cuatro agarres, tres sujetos



(a) Matriz de Confusión Regresión Logística (b) Matriz de Confusión SVM

Figura 6-19.: Matriz de Confusión comparación Regresión Logística con SVM

Al ver la información contenida en la tabla 6-8 podemos informa varios factores.

- El método de SVM frente al de regresión logística One vs All presenta mayor exactitud en la predicción de las clases.
- En este experimento el conjunto de entrenamiento se formo con los tres primeros ensayos, mientras que la validación se formo con el cuarto ensayo. Se evidencia que tanto el método de regresión logística One Vs ALL como el método SVM presentan un alto error en la predicción del conjunto de validación.

7. Aplicación para Validación Online

Hasta el momento se ha evaluado de manera off-line el sistema implementado, es decir en un intervalo de tiempo se registró toda la información de los diferentes sujetos (de acuerdo al protocolo presentado) y posteriormente se evaluó la exactitud de los algoritmos de clasificación. A pesar de la utilidad de este esquema procedimental, realmente no es el que se utilizaría en el control de una prótesis, en cuyo caso el sistema debería realizar todas las actividades de forma continua y en un corto lapso de tiempo, es decir registrar las señales, procesarlas, clasificarlas y con base en los resultados generar una orden que conlleve al desarrollo de una acción particular en la prótesis (abrir la mano, cerrarla etc).

Esta última fase del proyecto tenía por objetivo demostrar el funcionamiento del sistema de manera on-line, básicamente siguiendo el esquema de pasos que serían necesarios ejecutar para el control de una prótesis real. Para ello se elaboró una aplicación de usuario en Matlab a través de la cual se realizan de manera continua las siguientes actividades: registrar las señales que entrega el dispositivo electrónico, procesarlas y clasificarlas para finalmente generar una salida que simule la acción que debería efectuar la prótesis.

Durante la fase destinada a la construcción de la base de datos (para el procesamiento off-line de la información) se optó por elaborar otra aplicación en Matlab para facilitar el procedimiento de registro y disminuir las probabilidades de error en el proceso, esta aplicación fue presentada en el

7. Aplicación para Validación Online

capítulo 5 y constituyó el punto de partida para la aplicación de validación on-line.

En la figura 7-1 se presenta la ventana a través de la cual el usuario puede observar el funcionamiento del sistema implementado. A diferencia de la aplicación anterior, esta no cuenta con los campos destinados a recopilar la información del sujeto de prueba, en su lugar se incluyó un nuevo campo a través del cual se selecciona el clasificador a utilizar (el cual ha sido previamente entrenado).

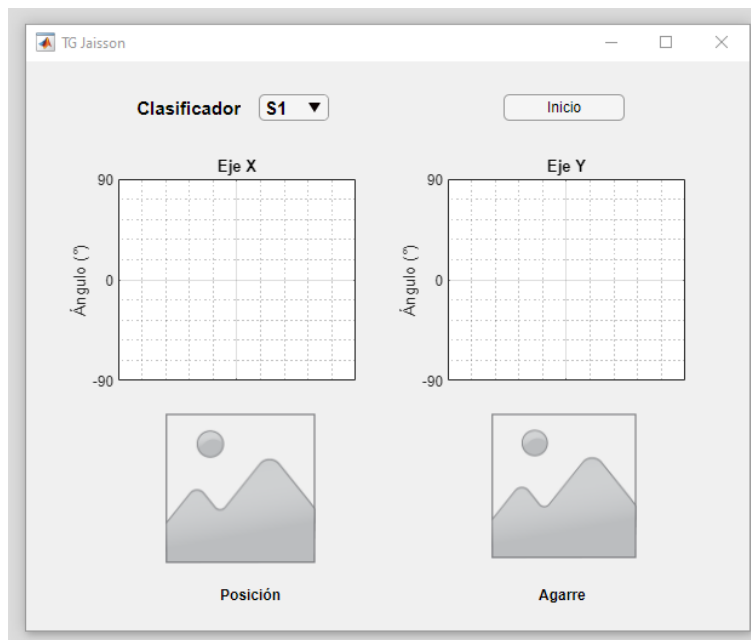


Figura 7-1.: Ventana - Aplicación para validación On-line

Los ensayos realizados con la aplicación, evidenciaron que el desempeño on-line del sistema implementado es inferior a los obtenidos en los análisis off-line, en lo que concierne a la exactitud de los agarres predichos por

7. Aplicación para Validación Online

el sistema. Esto es evidente aun cuando la evaluación se realiza de forma visual y no se cuantifica. De acuerdo a los resultados off-line se esperaría que el acierto en la predicción de los agarres fuese cercano al 90 %, es decir que por cada 10 combinaciones de agarre-posición diferentes el sistema debería equivocarse solo 1 vez, sin embargo se encuentra que a pesar de que la exactitud en la predicción de la posición es casi del 100 %, la exactitud en la predicción de los agarres es mucho menor, incluso podría decirse que entorno al 50 %.

Con base en los resultados visuales obtenidos, se optó por realizar un análisis, off-line, del desempeño de los clasificadores simulando las condiciones a las cuales se vería expuesto el sistema durante una prueba on-line. A continuación se describen las pruebas realizadas y se presentan algunas tablas con los resultados obtenidos.

Todas las pruebas fueron realizadas empleando clasificadores supervisados basados en máquinas de soporte vectorial, los cuales fueron configurados para utilizar el 100 % de los datos que se ingresan y realizan validación cruzada con 5 iteraciones durante el entrenamiento para reducir el sobreajuste.

1. Se seleccionó como sujeto de prueba el número 4, debido a que éste fue el utilizado inicialmente para las pruebas on-line. Se tomaron todas las medidas del Ensayo 1 para entrenar el clasificador y se utilizaron todas las medidas del ensayo 4 para validar su desempeño. De esta forma se simula el retiro del brazalete y su posterior reubicación. En esta prueba se obtuvo una exactitud del 40.47 % en la predicción del agarre, lo cual corresponde aproximadamente con lo observado en las pruebas on-line.
2. Se repitió el proceso descrito en la prueba No. 1 pero se adiciono el conjunto de muestras del Ensayo 2 al conjunto de entrenamiento. En esta prueba se obtuvo una exactitud del 50.48 % en la predicción del agarre.
3. Se adicionaron las muestras del Ensayo 3 al conjunto de entrenamiento y se repitió el proceso descrito en las pruebas 1 y 2. En esta prueba se obtuvo

7. Aplicación para Validación Online

una exactitud del 67.01 % en la predicción del agarre.

S	Conjunto de Entrenamiento					Conjunto de Validación		
	Ensayos			Exactitud	Muestras	Ensayos	Exactitud	Muestras
4	1			100	12000	4	40.47	12000
	1	2		99.9	24000	4	50.48	12000
	1	2	3	99	36000	4	67.01	11200

Tabla 7-1.: Relación exactitud y Error Clasificador 1

Un análisis preliminar de estas pruebas permite inferir que el incremento en el número de ensayos podría mejorar el desempeño de las medidas online, tomando en cuenta que el conjunto completo de medidas empleadas para la etapa de validación corresponden a un ensayo completamente nuevo.

Tomando en cuenta las implicaciones que conlleva el protocolo para adquirir nuevas muestras, principalmente el tiempo de registro y la disponibilidad de tiempo de los sujetos, se planteó evaluar otra alternativa. Para entrenar el clasificador se utilizaron todas las muestras de los ensayos No. 1, 2 y 3 mas una pequeña porción del ensayo No. 4 (destinado completamente a la validación en las pruebas anteriores). Para ello se separó en dos partes el conjunto de muestras del ensayo No. 4, un subconjunto fue utilizado para complementar las muestras empleadas durante el entrenamiento, y el subconjunto restante fue utilizado para evaluar el desempeño del clasificador. En las siguientes tablas se presentan los resultados obtenidos, inicialmente se evaluó el desempeño de los clasificadores con los registros del sujeto No. 4 y posteriormente se complementó el estudio realizando las mismas pruebas con los sujetos 1, 2 y 3.

En estas tablas se evidencia que al momento de incluir datos del ensayo de validación en el conjunto de entrenamiento, se produce un cambio positivo en la exactitud de la predicción. Los cambios son tan notables, que el

7. Aplicación para Validación Online

solo hecho de incluir una muestra conlleva a un incremento de alrededor del 20 % o mas en la exactitud de predicción del clasificador, y cuando se incluye 4 muestras se consiguen exactitudes del orden del 85 % en promedio.

Se adjunta como Anexo L el algoritmo de la Aplicación de validación Online

S	Conjunto de Entrenamiento					Conjunto de Validación			
	Ensayos				Exactitud	Muestras	Ensayos	Exactitud	Muestras
4	1	4 [1]			100	12800	4	67,14	11600
	1	2	4 [1 3]		99,9	24400	4	69,2	11600
	1	2	3	4 [1]	99	36400	4	83,22	11600

Tabla 7-2.: Relación exactitud y Error Clasificador 1

S	Conjunto de Entrenamiento					Conjunto de Validación			
	Ensayos				Exactitud	Muestras	Ensayos	Exactitud	Muestras
4	1	4 [1 3]			100	12800	4	84,19	11200
	1	2	4 [1 3]		99,8	24800	4	79,17	11200
	1	2	3	4 [1 3]	99	36800	4	83,74	11200

Tabla 7-3.: Relación exactitud y Error Clasificador 1

Desde luego desde el punto de vista de la finalidad la práctica del sistema, esto implicaría destinar un tiempo para el reentrenamiento cada vez que el sujeto se instale el brazalete o la prótesis, sin embargo esta alternativa podría considerarse viable tomando en cuenta el tiempo necesario que conllevaría este proceso y que solo se requeriría realizar una vez (alrededor de 5 minutos para registrar una medida completa, realizar el procesamiento de la información y reentrenar el algoritmo de clasificación).

8. Conclusiones

El desarrollo de prótesis y el estudio de mecanismos para su control, sigue siendo un campo relevante y abierto, la complejidad de la problemática obliga a un análisis desde diversas perspectivas y áreas del conocimiento, lo cual desde luego desborda los objetivos del presente trabajo, sin embargo las actividades realizadas y los resultados obtenidos constituyen un aporte en esta línea, particularmente en lo concerniente a la identificación de intencionalidad de movimiento, considerando esta etapa como un paso fundamental para el posterior control de prótesis.

El trabajo se dividió en 4 fases, (1) construcción de un dispositivo electrónico [conjunto: brazaletes, sensores, microcontrolador] para registrar información correlacionada con el movimiento de brazo y agarres efectuados por un sujeto de prueba, (2) elaboración de un protocolo de medida y construcción de una base de datos con la información recolectada de 10 sujetos de prueba, (3) uso y validación de herramientas computacionales para la predicción automática del movimiento realizado por los sujetos de prueba, y (4) elaboración de una aplicación para demostrar el desempeño on-line del sistema desarrollado.

El dispositivo electrónico fue construido empleando elementos de bajo costo, tanto los sensores de fuerza, sensores de movimiento inercial y el microcontrolador son de gama baja, sin embargo las medidas recopiladas y el algoritmo implementado, basado en interrupciones, reflejan un desempeño adecuado para el registro de las variables físicas abordadas. Desde luego es conveniente recordar que este prototipo se enfocó al estudio de identificación

8. Conclusiones

de intención de movimiento con posiciones y agarres estables, manteniendo fija la configuración durante cada periodo de registró (por al menos 5 segundos).

El protocolo implementado, el número de sujetos, ensayos y medidas, fueron adecuados para la construcción de una base de datos acorde con las necesidades establecidas por los objetivos del presente proyecto, dicha información permitió desarrollar apropiadamente los procesos de identificación basados en técnicas de aprendizaje automático. La base de datos construida puede ser utilizada como punto de partida en estudios posteriores, los cuales se orienten a la optimización de los algoritmos de identificación, análisis estadísticos de las señales etc.

Los resultados obtenidos por los algoritmos de identificación automática, Regresión Logística y máquinas de Soporte Vectorial, permitieron inferir un comportamiento adecuado del sistema en conjunto, resaltándose los siguientes dos aspectos: (1) las señales entregadas por los sensores de fuerza permitieron identificar adecuadamente los 4 agarres abordados en el presente estudio, y (2) el uso de las IMU mejoró el desempeño de los sistemas de identificación. Desde luego los resultados también reflejan algunas limitantes del presente prototipo, las cuales se deberían abordar para el desarrollo de un sistema mas robusto; particularmente se observó que el aumento de ensayos con cada sujeto contribuye positivamente a disminuir el error en la predicción de los agarres y de esta forma mejora el desempeño de las pruebas on-line, aspecto que queda en evidencia en los ensayos realizados con la aplicación de demostración elaborada en Matlab.

9. Trabajos futuros

Con base en la realización de este trabajo se desglosan actividades y trabajos que pueden dar continuidad al presente proyecto en búsqueda de optimizar el desempeño del sistema desarrollado.

- Analizar a profundidad el desempeño de los algoritmos de clasificación (ajustar los parámetros) y probar otros.
- Emplear una IMU con 9 DOFS, tomando en cuenta el desplazamiento del ángulo en el plano.
- Realizar medidas con mas ángulos del brazo y mas agarres.
- Realizar medidas con variaciones dinámicas del ángulo del brazo (no estáticas o a ángulos fijos).
- Realizar medidas incluyendo como variable la fuerza ejercida en cada agarre (por ejemplo ubicar sensores de presión en los dedos).
- Aumentar el número de sensores de presión (por ejemplo diseñar un brazalete que contenga una matriz de sensores en lugar de un solo arreglo lineal)

9. Trabajos futuros

- Incluir otra IMU para realizar medidas variando el giro de la muñeca y/o el antebrazo.
- Realizar ensayos registrando de forma simultánea señales de electromiografía.
- Realizar ensayos registrando de forma simultánea señales de electroencefalografía.
- Realizar pruebas con sujetos que hayan sufrido la amputación del miembro superior.
- Aumentar el número de sujetos y hacer un análisis tomando parámetros como el género, la edad.
- Definir un método para normalizar las medidas de presión entre los diferentes sujetos, y repetir los análisis con este ajuste.
- Acondicionar externa o internamente las señales entregadas por los sensores de fuerza, bien sea para trabajarlos en margen lineal o calcular exactamente la presión que esta ejerciendo cada sensor de presión.

Anexos

A. Anexos: General

Para facilitar la consulta, análisis y reproducción, de todas las actividades descritas en el presente proyecto, la base de datos y los programas desarrollados se encuentran disponibles a través del siguiente enlace:

<https://drive.google.com/drive/folders/1Ff9WYExTgec7-Iq63g4i0w...GIAeHbzUXN?usp=sharing>

B. Anexo I: Código en lenguaje C – Programa del microcontrolador

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/wdt.h>

#define n_muestras 20
#define IMU1_DIR 0x68
#define IMU2_DIR 0x69
#define DIR_WHO_AM_I 0x75
#define DIR_ACCEL 0x3b
#define DIR_GYRO 0x43
#define DIR_TEMP 0x41
#define DIR_CONFIG 0x1A
#define DIR_ACCEL_CONF 0x1C
#define DIR_GYRO_CONF 0x1B
#define DIR_FIFO_EN 0x23
#define DIR_FIFO_COUNTH 0x72
#define DIR_FIFO_COUNTL 0x73
#define DIR_FIFO_DATA 0x74
```

B. Anexo I: Código en lenguaje C – Programa del microcontrolador

```
#define DIR_PWR_MGMT_1 0x6B
#define DIR_PWR_MGMT_2 0x6C
#define DIR_SMPLRT 0x19
#define DIR_INT_ENABLE 0x38

#define I2C_escribiendo 0
#define I2C_leyendo 1

//----- DEFINICIÓN DE VARIABLES GLOBALES
volatile unsigned int n_comp;
volatile unsigned char x;

volatile unsigned int sensor_ADC_Temp[16];
volatile unsigned char sensor_ADC[16];
volatile unsigned char sensor_buffer[8];
volatile unsigned char datos_buffer[52];
volatile unsigned char canal_sensor;
volatile unsigned char c_dato_Tx;
volatile unsigned int n_medida;
volatile unsigned char n_medidas_ADC;
volatile unsigned char reading;
volatile unsigned char i,j,q;
volatile unsigned char IMU_WriteMode,..
    IMU_ReadMode,IMU1_WriteMode, IMU1_ReadMode,...
    IMU2_WriteMode, IMU2_ReadMode;
volatile unsigned char accel[6];
volatile unsigned char giro[6];
volatile unsigned char temp[2];
volatile unsigned char datoPrueba[14];
volatile unsigned char datoR_I2C[14];
volatile unsigned char dato_IMU1[14];
volatile unsigned char dato_IMU2[14];
```

B. Anexo I: Código en lenguaje C – Programa del microcontrolador

```
volatile unsigned char datoW_I2C;
volatile unsigned char contDatoPrueba;
volatile unsigned char proceso_I2C;
volatile unsigned char estado_I2C;
volatile unsigned char dir_reg_I2C;
volatile unsigned char num_dato_leido;
volatile unsigned char num_datos_leer;
volatile unsigned char num_datos_Tx;
volatile unsigned char band_Datos_IMU_Completo;
volatile unsigned char guardando_datos;
volatile unsigned char Cuenta_datos_guardados;
volatile unsigned char cuenta_leds;
volatile unsigned char LEDS_G;
volatile unsigned char n_comp2,led_segundos;

// ----- PROTOTIPOS DE FUNCIÓN
void config_serial(void);
void config_ADC(void);
void config_puertos(void);
void config_Int_Ext(void);
void config_watchdog(void);
void config_temp();
void config_I2C(void);
void config_IMU(void) ;
void nada(void);
void delay_mseg(unsigned int m_segundos);

void IMU_GetData(unsigned char N_IMU, unsigned char dir_reg,...
                unsigned char n_datos);
void IMU_PutData(unsigned char N_IMU, unsigned char dir_reg,...
                unsigned char dato_write);
```

B. Anexo I: Código en lenguaje C – Programa del microcontrolador

```
// ----- RUTINAS DE INTERRUPCIÓN

// ----- RUTINA DE SERVICIO DE INTERRUPCIÓN ISR...
// POR RECEPCIÓN DE DATOS POR USART -----

ISR(USART2_RX_vect) // Si el modulo USART detecta una recepción...
                    // completa se ejecuta esta rutina de interrupción
{
    reading=UDR2; // lee lo que recibió (al leer el valor recibido...
                 // se limpia el buffer)
    if(reading==0) // Verifica si recibió el valor 0
    {
        canal_sensor=0; // se inicializa la variable que controla...
                        // el canal del ADC
        n_medidas_ADC=0; // Se inicializa la variable que controla...
                        // el numero de medidas para obtener un valor...
                        // promedio por cada canal del ADC
        for (q=0;q<52;q++)
        {
            datos_buffer[q]=0;
        }

        for (q=0;q<16;q++)
        {
            sensor_ADC_Temp[q]=0;
        }

        TCCR2B= 0b00000110; // Activamos el temporizador para que genere...
                            // interrupciones cada 4ms (este controla cada...
                            // cuanto se tomaran medidas y se enviaran al PC,...
                            // se configura el programa para que tome medidas...
                            // cada 20ms).
```

B. Anexo I: Código en lenguaje C – Programa del microcontrolador

```
band_Datos_IMU_Completo=0;
IMU_GetData(1,DIR_ACCEL,14); // Se inicia la primera lectura de...
    datos de la IMU
ADCSRA=ADCSRA|0b01000000; // Se inicia la primera conversión del ADC
n_medida=0;

}
if (reading==10)
{
    guardando_datos=1;
    Cuenta_datos_guardados=0;
    cuenta_leds=0;
    LEDES_G=0;
}

if (reading==20)
{
    TCCR2B= 0b00000110;
    PORTA=~(0b11000011);
}

}

// ----- RUTINA DE SERVICIO DE INTERRUPCIÓN ISR POR TRANSMISIÓN...
// DE DATOS POR USART -----

ISR(USART2_TX_vect) // Esta rutina de interrupción se ejecuta cuando...
                    // el modulo USART detecta que ha finalizado...
                    // la transmisión de datos,
{
    UCSR2A=UCSR2A|0b01000000; // Limpia la bandera de final de Transmisión
    c_dato_Tx++;
}
```

B. Anexo I: Código en lenguaje C – Programa del microcontrolador

```
    if(c_dato_Tx<num_datos_Tx)
    {
        UDR2=datos_buffer[c_dato_Tx]; // transmite los valores...
                                         almacenados en el buffer de datos
    }
}

// ----- RUTINA DE SERVICIO DE INTERRUPCIÓN ISR POR...
//                FINAL DE CONVERSIÓN DE ADC -----

ISR(ADC_vect)
{
    sensor_ADC_Temp[(canal_sensor*2)+1] = ...
    sensor_ADC_Temp[(canal_sensor*2)+1]+ADCL;
    sensor_ADC_Temp[(canal_sensor*2)] = ...
    sensor_ADC_Temp[(canal_sensor*2)]+ADCH;
    canal_sensor++;

    ADMUX=ADMUX&0b11100000; // *** Limpia los bits del canal

    if(canal_sensor<8)
    {
        ADMUX=ADMUX|canal_sensor; // Cambia de canal
        ADCSRA=ADCSRA|0b01000000; // Se inicia la conversión...
                                     colocando un 1 en el bit ADSC,...
                                     sin modificar los otros bits
    }
    else
    {
        n_medidas_ADC++;
        canal_sensor=0;
    }
}
```

B. Anexo I: Código en lenguaje C – Programa del microcontrolador

```
    if (n_medidas_ADC<3)
    {
        ADMUX=ADMUX|canal_sensor; // Cambia de canal
        ADCSRA=ADCSRA|0b01000000; // Se inicia la conversión...
                                   colocando un 1 en el bit ADSC,...
                                   sin modificar los otros bits
    }
    else
    {
        for (q=0;q<16;q++)
        {
            sensor_ADC[q]=sensor_ADC_Temp[q]/3;
            // Para tomar un promedio de 3...
            // lecturas de cada canal del ADC
        }
    }
}

// ----- RUTINA DE SERVICIO DE INTERRUPCIÓN ISR POR I2C

ISR(TWI_vect)
{
    if(proceso_I2C == I2C_leyendo)
    {
        switch (estado_I2C)
        {
            case 1 : // Ingresa en este caso cuando completa el START
                    // Envía la dirección de la IMU con indicación de...
                    // escritura (para indicar cual es el registro a leer)

```


B. Anexo I: Código en lenguaje C – Programa del microcontrolador

```
        TWDR = IMU_WriteMode ;
        TWCR = (1<<TWEN) | (1<<TWIE) | (1<< TWINT);
        estado_I2C++;
        break;
case 2: // Ingresa en este caso cuando termino de enviar...
        dirección de la IMU, ahora debe enviar la dirección...
        del registro a leer
        TWDR = dir_reg_I2C ;
        TWCR = (1<<TWEN) | (1<<TWIE) | (1<< TWINT);
        estado_I2C++;
        break;
case 3: // Ingresa aquí cuando termino de enviar la dirección...
        del registro a leer, ahora debe proceder a enviar un re-start
        TWCR = (1<<TWEN) | (1<<TWIE) | (1<<TWSTA) | (1<<TWINT);
        estado_I2C++;
        break;
case 4: // Ingresa aquí cuando termino de enviar el re-start,...
        ahora debe proceder a enviar la dirección de la IMU....
        pero con la condición de lectura
        TWDR = IMU_ReadMode ;
        TWCR = (1<<TWEN) | (1<<TWIE) | (1<< TWINT);
        estado_I2C++;
        break;
case 5: // Ingresa aquí cuando termino de enviar la dirección...
        de la IMU pero con la condición de lectura, ahora debe...
        proceder a iniciar la lectura de datos
        if (num_datos_leer>1)
        {
            TWCR = (1<<TWEN) | (1<<TWEA) | (1<<TWIE) | (1<< TWINT);
        }
        else
        {
```

B. Anexo I: Código en lenguaje C – Programa del microcontrolador

```
        TWCR = (1<<TWEN) | (1<<TWIE) | (1<< TWINT);
    }
    num_dato_leido=0;
    estado_I2C++;
    break;
case 6: // Ingresa aqui cada vez que recibe un dato
    datoR_I2C[num_dato_leido]=TWDR;
    num_dato_leido++;
    if (num_dato_leido < (num_datos_leer-1))
    {
        TWCR = (1<<TWEN) | (1<<TWEA) | (1<<TWIE) | (1<< TWINT) ;
    }
    else
    {
        if (num_dato_leido < (num_datos_leer))
        {
            TWCR = (1<<TWEN) | (1<<TWIE) | (1<< TWINT);

        }
        else
        {
            estado_I2C++; // Se sale de este caso cuando...
            ha terminado de recibir todos los datos solicitados
        }
    }
    break;
case 7: // Finaliza la comunicacion y libera el bus I2C
    TWCR = (1<<TWEN) | (1<<TWSTO) | (1<<TWIE) | (1<< TWINT);
    if (IMU_WriteMode==IMU1_WriteMode)
    {
        for (q=0;q<14;q++)
        {
```

B. Anexo I: Código en lenguaje C – Programa del microcontrolador

```
        dato_IMU1[q]=datoR_I2C[q];
    }
    IMU_GetData(2,DIR_ACCEL,14);
}
else
{
    for (q=0;q<14;q++)
    {
        dato_IMU2[q]=datoR_I2C[q];
    }
    band_Datos_IMU_Completo=1;
}
break;
}
}
else
{
    //PORTA=64;
    //PORTA=TWSR;
    //nada();
    switch (estado_I2C)
    {
        case 1 : // Ingresa en este caso cuando completa el START
            // Envía la dirección de la IMU con indicación...
            de escritura (para indicar cual es el registro a leer)
            //PORTA=TWSR;
            TWDR = IMU_WriteMode ;
            TWCR = (1<<TWEN) | (1<<TWIE) | (1<< TWINT);
            estado_I2C++;
            break;
        case 2: // Ingresa en este caso cuando termino de enviar...
```

B. Anexo I: Código en lenguaje C – Programa del microcontrolador

```
        dirección de la IMU, ahora debe enviar la dirección...
        del registro a leer
        TWDR = dir_reg_I2C ;
        TWCR = (1<<TWEN) | (1<<TWIE) | (1<< TWINT);
        estado_I2C++;
        break;
    case 3: // Ingresa aquí cuando termino de enviar la dirección...
            del registro a escribir, ahora debe proceder a enviar...
            el dato respectivo
            TWDR = datoW_I2C ;
            TWCR = (1<<TWEN) | (1<<TWIE) | (1<< TWINT);
            estado_I2C++;
            break;
    case 4: // Finaliza la comunicación y libera el bus I2C
            TWCR = (1<<TWEN) | (1<<TWSTO)| (1<<TWIE) | (1<< TWINT);
            break;
    }
}

// ----- RUTINA DE SERVICIO DE INTERRUPCIÓN ISR POR INTERRUPCIÓN EXTERNA

ISR(INT4_vect)
{
    guardando_datos=1;

    Cuenta_datos_guardados=0;
    cuenta_leds=0;
    LEDS_G=0;
    while ((PINE&0b00010000)==0){}
}
```

B. Anexo I: Código en lenguaje C – Programa del microcontrolador

```
// ----- RUTINA DE SERVICIO DE INTERRUPCIÓN ISR POR COMPARACIÓN TIMERO

ISR(TIMERO_COMPA_vect)
{
    n_comp++;
}

// ----- RUTINA DE SERVICIO DE INTERRUPCIÓN ISR POR COMPARACIÓN TIMER2

ISR(TIMER2_COMPA_vect) // Esta rutina se ejecuta cada 4ms
{

    TIFR2=0b00000010;
    n_comp++; // Cada que se produce la interrupción por comparación...
              se incrementa en uno (1) el valor de la variable n_comp,...
              en el modo CTC el valor de TCNT retorna automáticamente a 0.
    n_comp2++;
    if (n_comp2>=250) // Enciende y apaga un led cada 1 segundo
    {
        if (led_segundos!=0)
        {
            PORTA=PINA|0b10000000;
            led_segundos=1;
        }
        else
        {
            PORTA=PINA&0b01111111;
            led_segundos=0;
        }
    }
}
```

B. Anexo I: Código en lenguaje C – Programa del microcontrolador

```
    n_comp2=0;
}
if(n_comp>=10) // con este condicional se garantiza...
                la toma de medidas cada 20 ms
{
    //IMU_GetData(DIR_WHO_AM_I,vectx,1);

    // Actualiza los datos del buffer para su transmisión
    if (band_Datos_IMU_Completo==1)
    {
        if (guardando_datos==1)
        {
            if (LEDS_G==0)
            {
                PORTA=~(0b10000000);
                LEDS_G=1;
            }
            else
            {
                PORTA=~(0b10011000);
                LEDS_G=0;
            }

            if (Cuenta_datos_guardados<50)
            {
                Cuenta_datos_guardados++;
            }
            else
            {
                guardando_datos=0;
                Cuenta_datos_guardados=0;
            }
        }
    }
}
```

B. Anexo I: Código en lenguaje C – Programa del microcontrolador

```
        datos_buffer[0]=0;
        datos_buffer[1]=0;
        datos_buffer[2]=1;
        datos_buffer[3]=1;
    }
else
{
    PORTA=~0b00011000;
    datos_buffer[0]=0;
    datos_buffer[1]=0;
    datos_buffer[2]=0;
    datos_buffer[3]=0;
}

for (i=4;i<18;i++)
{
    datos_buffer[i]=dato_IMU1[i-4]; // AcelX (2 bytes)...
        - AcelY (2 bytes) - AcelZ (2 bytes) - TempH...
        - TempL - GyroX (2 bytes) - GyroY (2 bytes)...
        - GyroZ (2 bytes)
}
for (i=18;i<32;i++)
{
    datos_buffer[i]=dato_IMU2[i-18]; // AcelX (2 bytes)...
        - AcelY (2 bytes) - AcelZ (2 bytes) - TempH...
        - TempL - GyroX (2 bytes) - GyroY (2 bytes)...
        - GyroZ (2 bytes)
}
for (i=32;i<48;i++)
{
    datos_buffer[i]=sensor_ADC[i-32]; // ADC0 a ADC7...
        (8 bytes) // Medidas con precision de 8 bits,...
```

B. Anexo I: Código en lenguaje C – Programa del microcontrolador

```
        se podria aumentar a 10
    }
    datos_buffer[48]=(n_medida+1)>>8; // Numero de la medida
    datos_buffer[49]=(n_medida+1)&0x00FF;
}
else
{
    PORTA=~0b00110011; // Indica que ocurrió un error de...
                        comunicación con las IMU
    datos_buffer[0]=1;
    datos_buffer[1]=1;
    datos_buffer[2]=0;
    datos_buffer[3]=0;
}

for (i=50;i<52;i++) // Bits de finalización en 255
{
    datos_buffer[i]=255;
}

c_dato_Tx=0;
num_datos_Tx=52;
UDR2=datos_buffer[0]; // Inicia la transmisión de datos

canal_sensor=0;
n_medidas_ADC=0;
for (q=0;q<16;q++) // Reinicia el vector acumulador de...
                    medidas del ADC
{
    sensor_ADC_Temp[q]=0;
}
ADCSRA=ADCSRA|0b01000000; // Se reinicia la conversión del ADSC
```


B. Anexo I: Código en lenguaje C – Programa del microcontrolador

```
        band_Datos_IMU_Completo=0; // Con esta bandera se identifica...
            si hubo alguna desconexión de la IMU antes de...
            terminar de leer todos los datos
        IMU_GetData(1,DIR_ACCEL,14); // Se reinicia la lectura de...
            datos de la IMU

        n_medida++;

        n_comp=0;

    }

}

// ----- RUTINA PRINCIPAL -----

int main()
{
    cli();
    n_comp=0;
    n_medida=0;
    canal_sensor=0;

    config_puertos();
    config_Int_Ext();
    config_serial();
    config_temp();
    config_ADC();
    config_watchdog();
    sei();
}
```

B. Anexo I: Código en lenguaje C – Programa del microcontrolador

```
config_I2C();
config_IMU();

PORTA=~1; // Indica que terminó la etapa de configuración de periféricos

n_comp2=0;
led_segundos=0;

while(1)
{

}

return 0;
}

// ----- SUB_RUTINAS -----

// ----- RUTINA DE CONFIGURACIÓN DE LOS PUERTOS I/O -----
void config_puertos(void)
{
    PORTE=0b00010000;
    DDRE=0b00000000;

    PORTA=0b00000000;
    DDRA=0b11111111;

    PORTA=0b01010101; // Indica que se reinicio y acaba....
                      de configurar los puertos
}
}
```

B. Anexo I: Código en lenguaje C – Programa del microcontrolador

```
// ----- RUTINA DE CONFIGURACIÓN DE LAS INTERRUPCIONES EXTERNAS
void config_Int_Ext(void)
{
    EICRB=0b000000010;
    EIMSK=0b00010000;
}

// ----- RUTINA DE CONFIGURACIÓN DEL TIMER -----
void config_temp()
{
    // Esta rutina programa el TIMER2 para que ejecute una...
    // interrupción por comparación cada que transcurre 1ms,...
    // en ella se incrementa en uno (1) el valor de la variable...
    // n_comp, luego espera a que el valor de n_comp coincida ...
    // con el valor "m_segundos", antes de retornar al flujo normal...
    // del programa.
    // En resumen realiza un retardo de 1ms*m_segundos.
    n_comp=0; // Inicializamos la bandera de conteo en 0, esta se...
    // incrementara en uno (1) cada que transcurra 1 ms (mediante la...
    // rutina de interrupción)

    TCNT2=0; // Inicializamos la cuenta en 0
    OCR2A=249; // Utilizando un cristal de 16 MHz y el preescalador ...
    // de 64, con este valor de OCR se producirá una interrupción por...
    // comparación cada 1ms
    TIMSK2=0b00000010; //|= (1<<OCIE0A); // habilitamos las interrupciones...
    // del timer por comparación
    TCCR2A = 0b01000010; //(1<<COM0A0)|(1<<WGM01); // Configuramos el...
    // TIMER0 en modo CTC (Cler Timer on Compare), con el pin OCOA ...
    // activado en modo Toggle y OCOB desconectado. Cada 1ms cambiara ...
}
```

B. Anexo I: Código en lenguaje C – Programa del microcontrolador

```
de estado el pin OCOA

}

// ----- RUTINA DE CONFIGURACIÓN DEL TIMER PARA CREAR RETARDOS...
// DE 'm' MILISEGUNDOS (NECESARIOS EN LA CONFIGURACIÓN DE LA IMU)

void delay_mseg(unsigned int m_segundos)
{
    // Esta rutina programa el TIMERO para que ejecute una...
    // interrupción por comparación cada que transcurre 1ms,...
    // en ella se incrementa en uno (1) el valor de la variable...
    // n_comp luego espera a que el valor de n_comp coincida...
    // con el valor "m_segundos", antes de retornar al flujo...
    // normal del programa. En resumen realiza un retardo de ...
    // 1ms*m_segundos.
    n_comp=0; // Inicializamos la bandera de conteo en 0, esta...
    // se incrementara en uno (1) cada que transcurra 1 ms ....
    // (mediante la rutina de interrupción)

    TCNT0=0; // Inicializamos la cuenta en 0
    OCROA = 249; // Utilizando un cristal de 16 MHz y el ....
    // preescalador de 64, con este valor de OCR se producirá ...
    // una interrupción por comparación cada 1ms
    TIMSK0 |= (1<<OCIEOA); // habilitamos las interrupciones...
    // del timer por comparación
    TCCR0A = (1<<COMOAO)|(1<<WGM01); // Configuramos el TIMERO...
    // en modo CTC (Cler Timer on Compare), con el pin OCOA...
    // activado en modo Toggle y OCOB desconectado. Cada 1ms...
    // cambiará de estado el pin OCOA
    TCCR0B=(1<<CS01)|(1<<CS00); // Configuramos un preescalador...
    // de 64, esto tomando en cuenta que se empleara un cristal...
```

B. Anexo I: Código en lenguaje C – Programa del microcontrolador

```
externo de 16MHz

while(n_comp!= m_segundos){};
TCCR0B=0; // Detenemos el reloj del contador
}

// ----- RUTINA DE CONFIGURACIÓN DE LA COMUNICACIÓN SERIAL
void config_serial(void)
{
    // USART inicialización - Modulo USART2
    // Parámetros de la Comunicación: 8 Data, 1 Stop, No Parity
    // Transmisión habilitada, Transmisión Sincrónica, Recepción...
    // habilitada, Interrupción por recepción y transmisión...
    // habilitada, Rata de Transmisión de 38.4 kbps
    UCSR2B=(1<<RXCIE2)|(1<<TXCIE2)|(1<<RXEN2)|(1<<TXEN2);
    // Habilitamos la recepción y transmisión de datos, ...
    // habilitamos la interrupción por recepción, se emplea...
    // el modulo USART 0
    UBRR2=25; // Con estas linea y la siguiente se configura...
    // la velocidad de transmisión a utilizar: para un cristal...
    // de 16MHz da una velocidad de transmisión de 38400 bps
}

// ----- Funcion para configurar el ADC -----
void config_ADC(void)
{
    ADCSRA=0b10001011; // *** La distribucion de bits en el registro...
    // ADCSRA es: [ADEN,ADSC,ADATE,ADIF,ADIE,ADPS2,ADPS1,ADPS0] ***
    ADMUX =0b01000000; // *** La distribucion de bits en el registro...
    // ADMUX es: [REFS1,REFSO,ADLAR,MUX4,MUX3,MUX2,MUX1,MUX0] ***
    ADCSRB=0b00000001; // *** La distribucion de bits en el registro...
    // ADCSRB es: [NC,ACME,NC,NC,MUX5,ADTS2,ADTS1,ADTS0] ***
}
```

B. Anexo I: Código en lenguaje C – Programa del microcontrolador

```
DIDR0=0b11111111; // *** La distribucion de bits en el registro...
    DIDR0 es: [ADC7D,ADC6D,ADC5D,ADC4D,ADC3D,ADC2D,ADC1D,ADC0D] ***
}

// ----- Funcion para configurar el I2C -----
void config_I2C(void)
{
    TWSR=0x00; //set prescaler bits to zero
    TWBR=0x46; //SCL frequency is 50K for 16Mhz
    TWCR=0b00000101; //enable TWI module
}

// ----- Funcion para configurar la IMU -----
void config_IMU(void)
{
    IMU1_WriteMode=IMU1_DIR<<1;
    IMU1_ReadMode=(IMU1_DIR<<1)|1;

    IMU2_WriteMode=IMU2_DIR<<1;
    IMU2_ReadMode=(IMU2_DIR<<1)|1;

    delay_mseg(100);
    IMU_PutData(1,DIR_SMPLRT,0X07);
    delay_mseg(100);
    IMU_PutData(1,DIR_PWR_MGMT_1,0b0000001); // 0b0000001
    delay_mseg(100);
    IMU_PutData(1,DIR_CONFIG,0b00000000);
    delay_mseg(100);
    IMU_PutData(1,DIR_GYRO_CONF,0b00000000); // +- 200/s
    delay_mseg(100);
    IMU_PutData(1,DIR_ACCEL_CONF,0b11100000); // +- 2g
    delay_mseg(100);
}
```

B. Anexo I: Código en lenguaje C – Programa del microcontrolador

```
IMU_PutData(1,DIR_INT_ENABLE,0X01);
delay_mseg(100);

delay_mseg(100);
IMU_PutData(2,DIR_SMPLRT,0X07);
delay_mseg(100);
IMU_PutData(2,DIR_PWR_MGMT_1,0b0000001); // 0b0000001
delay_mseg(100);
IMU_PutData(2,DIR_CONFIG,0b00000000);
delay_mseg(100);
IMU_PutData(2,DIR_GYRO_CONF,0b00000000); // +- 200/s
delay_mseg(100);
IMU_PutData(2,DIR_ACCEL_CONF,0b11100000); // +- 2g
delay_mseg(100);
IMU_PutData(2,DIR_INT_ENABLE,0X01);
delay_mseg(100);

//IMU_PutData(DIR_FIFO_EN,0b10001000);

for (q=0;q<52;q++)
{
    datos_buffer[q]=8;
}

}

// ----- Función para leer datos de los registros de la IMU

void IMU_GetData(unsigned char N_IMU, unsigned char dir_reg,...
```

B. Anexo I: Código en lenguaje C – Programa del microcontrolador

```
        unsigned char n_datos)
{
    if (N_IMU == 1)
    {
        IMU_WriteMode=IMU1_WriteMode;
        IMU_ReadMode=IMU1_ReadMode;
    }
    else
    {
        IMU_WriteMode=IMU2_WriteMode;
        IMU_ReadMode=IMU2_ReadMode;
    }

    proceso_I2C = I2C_leyendo;
    estado_I2C = 1;
    num_datos_leer = n_datos;
    num_dato_leido = 0;
    dir_reg_I2C = dir_reg;

    TWCR = (1<<TWEN) | (1<<TWSTA) | (1<<TWIE) | (1<<TWINT); //
}

// ----- Función para escribir en un registro de la IMU -----

void IMU_PutData(unsigned char N_IMU, unsigned char dir_reg, unsigned...
                char dato_write)
{
    if (N_IMU == 1)
    {
        IMU_WriteMode=IMU1_WriteMode;
        IMU_ReadMode=IMU1_ReadMode;
    }
}
```


B. Anexo I: Código en lenguaje C – Programa del microcontrolador

```
    }
    else
    {
        IMU_WriteMode=IMU2_WriteMode;
        IMU_ReadMode=IMU2_ReadMode;
    }

    proceso_I2C = I2C_escribiendo;
    estado_I2C = 1;
    datoW_I2C = dato_write;
    dir_reg_I2C = dir_reg;

    TWCR = (1<<TWEN) | (1<<TWSTA) | (1<<TWIE) | (1<<TWINT);

}

// ----- Función para hacer depuración (detiene por completo...
//                   los procesos) -----
void nada(void)
{
    while(1)
    {
        cli();
    }
}
```

C. Anexo II: Código – Aplicación en Matlab para el registro de datos

```
% Properties that correspond to app components
properties (Access = public)
    TG_Jaisson_principal    matlab.ui.Figure
    Inicio                  matlab.ui.control.Button
    GrabarDatos             matlab.ui.control.Button
    PosicinDropDownLabel   matlab.ui.control.Label
    Posicion                 matlab.ui.control.DropDown
    SujetoEditFieldLabel   matlab.ui.control.Label
    Sujeto                   matlab.ui.control.EditField
    EdadEditFieldLabel     matlab.ui.control.Label
    Edad                     matlab.ui.control.NumericEditField
    GeneroDropDownLabel    matlab.ui.control.Label
    Genero                   matlab.ui.control.DropDown
    EnsayoNoEditFieldLabel matlab.ui.control.Label
    Ensayo                   matlab.ui.control.NumericEditField
    RegistroNoEditFieldLabel matlab.ui.control.Label
    Registro                 matlab.ui.control.NumericEditField
    AgarreDropDownLabel    matlab.ui.control.Label
    Agarre                   matlab.ui.control.DropDown
```

C. Anexo II: Código – Aplicación en Matlab para el registro de datos

```
end

properties (Access = public)
    %Property % Description
    SerMicro = serialport("COM8",38400);
    i;
    grabar;
    cuenta_grabar;
end

methods (Access = public)

    function InicioRegistro (app)

        %SerMicro = serialport("COM6",38400);
        %% %% ----- Envia el valor 0 al micro para que el empiece ...
        write(app.SerMicro,0,'uint8')

        %% IMU1
        acelx0_IMU1=8170;
        acely0_IMU1=7692;
        acelz0_IMU1=25782-16384; % offset menos 16384
        gyrox0_IMU1=-826.85;
        gyroy0_IMU1=57.34;
        gyroz0_IMU1=-83.79;

        datos_calibracion_IMU1=[acelx0_IMU1 acely0_IMU1 acelz0_IMU1 ...
        gyrox0_IMU1 gyroy0_IMU1 gyroz0_IMU1];

        %% IMU2
        acelx0_IMU2 = 9791;
```

C. Anexo II: Código – Aplicación en Matlab para el registro de datos

```
acely0_IMU2 = 8776;
acelz0_IMU2 = 26567-16384; % offset menos 16384
gyrox0_IMU2 = -290;
gyroy0_IMU2 = -59.56;
gyroz0_IMU2 = 39.20;

datos_calibracion_IMU2=[acelx0_IMU2 acely0_IMU2 acelz0_IMU2 ...

%% Para calibrar -->

datos_para_calibrar_IMU=[0 0 0 0 0 0];

%% --- Proceso de recepcion de datos
b=uint16(zeros(1000,52));
archivo_medidas=uint16(zeros(100,52));

Angle_IMU1=[0 0 0];
Angle_IMU2=[0 0 0];

vect_datos_IMU1=zeros(100,10);
vect_datos_IMU2=zeros(100,10);
vect_datos_ADC=zeros(100,8);
datosRx=ones(1,52);

datos_nuevos_IMU1=zeros(1,10);
datos_nuevos_IMU2=zeros(1,10);

while app.i==1
    datosRxN=uint16(read(app.SerMicro,52,'uint8'));
    datos_cabecera=datosRxN(1:4);
    if((datos_cabecera(1)+datos_cabecera(2))==0) % verifica...
        que no haya errores
```

C. Anexo II: Código – Aplicación en Matlab para el registro de datos

```
datosRx=datosRxN;
if app.grabar==1
    app.GrabarDatos.Enable=0;
    archivo_medidas(app.cuenta_grabar,:)=datosRx;
    app.cuenta_grabar = app.cuenta_grabar + 1;
    if app.cuenta_grabar>50
        nombre_archivo=strcat...
            ("medidas\",app.Sujeto.Value,"-",num2str(ap...
            p.Edad.Value)," - ",app.Ge...
            lue)," - Med_",num2str(app.Registro.Value),...
            " - Pos_",num2str(app.Posicion.Value)," - Ag_",num2st...
        writematrix(archivo_medidas,nombre_archivo);
        nombre_archivo2=strcat("medidas\",app.Sujet...

        nsayo.Value)," - Med_",num2str(app.Registr ...
        ue)," - Ag_",num2str(app.Agarre.Value)," - parametros...
        writematrix([strcat("nombre: ",app.Sujeto.Value);strc...
        ro: ",num2str(app.Registro.Value));strcat("...
        Posicion: ",num2str(app.Posicion.Value));st...
        rcat("Agarre: ",num2str(app.Agarre.Value))] ,nombre_ar...
        o2);
        archivo_parametros=uint16(zeros(100,52));
        app.cuenta_grabar=1;
        app.grabar=0;
        app.GrabarDatos.Enable=1;
        % guardar el vector datos_Archivo en un arc...
        hivo
        % incluyendo las variables del experimento
    end
end
end
end
```

C. Anexo II: Código – Aplicación en Matlab para el registro de datos

```
b=[b(2:1000,:);datosRxN];

datos_IMU1=datosRx(5:18);
datos_IMU2=datosRx(19:32);
datos_ADC=datosRx(33:48);
n_medidaH=datosRx(49);
n_medidaL=datosRx(50);

n_medidaT=conv_uint16_double(n_medidaH,n_medidaL); % 1 ...
valor

Angle_Ant_IMU1=Angle_IMU1;
[Acel_IMU1 , Gyro_IMU1 , Angle_IMU1, Temp_IMU1] = ...
compute_datos_IMU(datos_IMU1,datos_calibracion_IMU1,Ang...
le_Ant_IMU1); % 10 valores
datos_nuevos_IMU1 = [Acel_IMU1 , Gyro_IMU1 , Angle_IMU1...
, Temp_IMU1];

Angle_Ant_IMU2=Angle_IMU2;
[Acel_IMU2 , Gyro_IMU2 , Angle_IMU2, Temp_IMU2] = ...
compute_datos_IMU(datos_IMU2,datos_calibracion_IMU2,An...
gle_Ant_IMU2); % 10 valores
datos_nuevos_IMU2 = [Acel_IMU2 , Gyro_IMU2 , Angle_IMU...
2, Temp_IMU2];

sens_ADC = ((compute_ADC(datos_ADC).*5)./1023); % 8 va...
lores [ADC0 ADC1 ADC2 ADC3 ADC4 ADC5 ADC6 ADC7]

[vect_datos_IMU1, vect_datos_IMU2, ...
vect_datos_ADC]=plot_datos_IMU(vect_datos_IMU1 , ...
datos_nuevos_IMU1,vect_datos_IMU2 , datos_nuevos_IMU2,...
```

C. Anexo II: Código – Aplicación en Matlab para el registro de datos

```
        vect_datos_ADC, sens_ADC);

    end

    close all
    clear all

end

end

% Callbacks that handle component events
methods (Access = private)

    % Code that executes after component creation
    function startupFcn(app)
        clc
        close all
        clear all

        % ----- Muestra los puertos disponibles y configura uno
        disp("Puertos seriales disponibles: ")
        serialportlist("available")
        app.Inicio.Enable=1;
    end

    % Button pushed function: Inicio
    function InicioPushed(app, event)
%         clc
%         close all
%         clear all
```

C. Anexo II: Código – Aplicación en Matlab para el registro de datos

```
        app.i=1;
        app.Inicio.Enable=0;
        InicioRegistro (app)

    end

    % Button pushed function: GrabarDatos
    function GrabarDatosPushed(app, event)
        write(app.SerMicro,10,'uint8')
        app.grabar=1;
        app.cuenta_grabar=1;
    end
end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

    % Create TG_Jaisson_principal and hide until all component...
    % s are created
    app.TG_Jaisson_principal = uifigure('Visible', 'off');
    app.TG_Jaisson_principal.Position = [100 100 494 322];
    app.TG_Jaisson_principal.Name = 'TG Jaisson';

    % Create Inicio
    app.Inicio = uibutton(app.TG_Jaisson_principal, 'push');
    app.Inicio.ButtonPushedFcn = createCallbackFcn(app, @Inici...
    oPushed, true);
    app.Inicio.Position = [371 20 100 22];
```


C. Anexo II: Código – Aplicación en Matlab para el registro de datos

```
app.Inicio.Text = 'Inicio';

% Create GrabarDatos
app.GrabarDatos = uibutton(app.TG_Jaisson_principal, 'push...');
app.GrabarDatos.ButtonPushedFcn = createCallbackFcn(app, ...
@GrabarDatosPushed, true);
app.GrabarDatos.FontSize = 24;
app.GrabarDatos.FontWeight = 'bold';
app.GrabarDatos.Position = [309 157 176 90];
app.GrabarDatos.Text = 'Grabar datos';

% Create PosicinDropDownLabel
app.PosicinDropDownLabel = uilabel(app.TG_Jaisson_principa...
1);
app.PosicinDropDownLabel.HorizontalAlignment = 'right';
app.PosicinDropDownLabel.Position = [27 93 54 22];
app.PosicinDropDownLabel.Text = 'Posición ';

% Create Posicion
app.Posicion = uidropdown(app.TG_Jaisson_principal);
app.Posicion.Items = {'Ángulo recto', 'Completamente horiz...
ontal'};
app.Posicion.ItemsData = {'AR', 'CH'};
app.Posicion.Position = [108 93 185 22];
app.Posicion.Value = 'AR';

% Create SujetoEditFieldLabel
app.SujetoEditFieldLabel = uilabel(app.TG_Jaisson_principa...
1);
app.SujetoEditFieldLabel.HorizontalAlignment = 'right';
app.SujetoEditFieldLabel.Position = [34 277 40 22];
```

C. Anexo II: Código – Aplicación en Matlab para el registro de datos

```
app.SujetoEditFieldLabel.Text = 'Sujeto';

% Create Sujeto
app.Sujeto = uieditfield(app.TG_Jaisson_principal, 'text');
app.Sujeto.Position = [108 277 57 22];

% Create EdadEditFieldLabel
app.EdadEditFieldLabel = uilabel(app.TG_Jaisson_principal);
app.EdadEditFieldLabel.HorizontalAlignment = 'right';
app.EdadEditFieldLabel.Position = [37 246 34 22];
app.EdadEditFieldLabel.Text = 'Edad';

% Create Edad
app.Edad = uieditfield(app.TG_Jaisson_principal, 'numeric');
app.Edad.Limits = [0 100];
app.Edad.Position = [109 246 57 22];

% Create GeneroDropDownLabel
app.GeneroDropDownLabel = uilabel(app.TG_Jaisson_principal);
app.GeneroDropDownLabel.HorizontalAlignment = 'right';
app.GeneroDropDownLabel.Position = [31 210 46 22];
app.GeneroDropDownLabel.Text = 'Genero';

% Create Genero
app.Genero = uidropdown(app.TG_Jaisson_principal);
app.Genero.Items = {'Femenino', 'Masculino '};
app.Genero.ItemsData = {'F', 'M'};
app.Genero.Position = [109 210 100 22];
app.Genero.Value = 'F';

% Create EnsayoNoEditFieldLabel
app.EnsayoNoEditFieldLabel = uilabel(app.TG_Jaisson_princi...
```

C. Anexo II: Código – Aplicación en Matlab para el registro de datos

```
pal);
app.EnsayoNoEditFieldLabel.HorizontalAlignment = 'right';
app.EnsayoNoEditFieldLabel.Position = [18 175 71 22];
app.EnsayoNoEditFieldLabel.Text = 'Ensayo No. ';

% Create Ensayo
app.Ensayo = uieditfield(app.TG_Jaisson_principal, 'numeric');
app.Ensayo.Limits = [0 Inf];
app.Ensayo.Position = [108 175 100 22];

% Create RegistroNoEditFieldLabel
app.RegistroNoEditFieldLabel = uilabel(app.TG_Jaisson_pri...
ncipal);
app.RegistroNoEditFieldLabel.HorizontalAlignment = 'right';
app.RegistroNoEditFieldLabel.Position = [16 136 76 22];
app.RegistroNoEditFieldLabel.Text = 'Registro No. ';

% Create Registro
app.Registro = uieditfield(app.TG_Jaisson_principal, 'nume...
ric');
app.Registro.Limits = [0 Inf];
app.Registro.Position = [108 136 100 22];

% Create AgarreDropDownLabel
app.AgarreDropDownLabel = uilabel(app.TG_Jaisson_principal);
app.AgarreDropDownLabel.HorizontalAlignment = 'right';
app.AgarreDropDownLabel.Position = [31 53 42 22];
app.AgarreDropDownLabel.Text = 'Agarre';

% Create Agarre
app.Agarre = uidropdown(app.TG_Jaisson_principal);
app.Agarre.Items = {'Neutral', 'Pinza', 'Fuerte', 'Palma Ab...
```

C. Anexo II: Código – Aplicación en Matlab para el registro de datos

```
        ierta'});
        app.Agarre.ItemsData = {'N', 'P', 'F', 'A'};
        app.Agarre.Position = [108 53 100 22];
        app.Agarre.Value = 'N';

        % Show the figure after all components are created
        app.TG_Jaisson_principal.Visible = 'on';
    end
end

% App creation and deletion
methods (Access = public)

    % Construct app
    function app = TG_Jaisson_app

        % Create UIFigure and components
        createComponents(app)

        % Register the app with App Designer
        registerApp(app, app.TG_Jaisson_principal)

        % Execute the startup function
        runStartupFcn(app, @startupFcn)

        if nargin == 0
            clear app
        end
    end

end

% Code that executes before app deletion
function delete(app)
```

C. Anexo II: Código – Aplicación en Matlab para el registro de datos

```
        % Delete UIFigure when app is deleted
        delete(app.TG_Jaisson_principal)
    end
end
end
```

D. Anexo III: Código – Aplicación en Matlab para la validación On-Line

```
classdef Validacion_On_Line_app < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        ValidacionOnLine          matlab.ui.Figure
        Fig_EjeY                   matlab.ui.control.UIAxes
        Fig_EjeX                   matlab.ui.control.UIAxes
        Fig_Sens_ADC               matlab.ui.control.UIAxes
        Inicio                     matlab.ui.control.Button
        ClasificadorDropDownLabel  matlab.ui.control.Label
        ClasificadorDropDown       matlab.ui.control.DropDown
        Fig_Pos                    matlab.ui.control.Image
        Fig_Agarr                  matlab.ui.control.Image
        PosicinLabel               matlab.ui.control.Label
        AgarreLabel                matlab.ui.control.Label
    end

    properties (Access = public)
```

D. Anexo III: Código – Aplicación en Matlab para la validación On-Line

```
%Property % Description
SerMicro = serialport("COM12",38400);
i;
grabar;
cuenta_grabar;
P;
A;
SVM;
end

methods (Access = public)

function InicioRegistro (app)

    %% %% ----- Envia el valor 0 al micro para que el ...
    empiece a captura muestras y enviar -----
    write(app.SerMicro,0,'uint8')

    %% IMU1
    acelx0_IMU1=8170;
    acely0_IMU1=7692;
    acelz0_IMU1=25782-16384; % offset menos 16384
    gyrox0_IMU1=-826.85;
    gyroy0_IMU1=57.34;
    gyroz0_IMU1=-83.79;

    datos_calibracion_IMU1=[acelx0_IMU1 acely0_IMU1 ...
    acelz0_IMU1 gyrox0_IMU1 gyroy0_IMU1 gyroz0_IMU1];

    %% IMU2
    acelx0_IMU2 = 9791;
    acely0_IMU2 = 8776;
```

D. Anexo III: Código – Aplicación en Matlab para la validación On-Line

```
acelz0_IMU2 = 26567-16384; % offset menos 16384
gyrox0_IMU2 = -290;
gyroy0_IMU2 = -59.56;
gyroz0_IMU2 = 39.20;

datos_calibracion_IMU2=[acelx0_IMU2 acely0_IMU2 ...
acelz0_IMU2 gyrox0_IMU2 gyroy0_IMU2 gyroz0_IMU2];

%% Para calibrar -->

datos_para_calibrar_IMU=[0 0 0 0 0 0];

%% --- Proceso de recepcion de datos
b=uint16(zeros(1000,52));
archivo_medidas=uint16(zeros(50,52));
archivo_medidas_proc=zeros(50,15);

Angle_IMU1=[0 0 0];
Angle_IMU2=[0 0 0];

vect_datos_IMU1=zeros(100,10);
vect_datos_IMU2=zeros(100,10);
vect_datos_ADC=zeros(100,8);
datosRx=ones(1,52);

datos_nuevos_IMU1=zeros(1,10);
datos_nuevos_IMU2=zeros(1,10);

load('SVM_S4_E1234p3_P12_A1234_con_IMU_Clas_PA_CV.mat', ...
      'SVM_S4_E1234p3_P12_A1234_con_IMU_Clas_PA_CV');
```


D. Anexo III: Código – Aplicación en Matlab para la validación On-Line

```
app.SVM = SVM_S4_E1234p3_P12_A1234_con_IMU_Clas_PA_CV;

while app.i==1
    datosRxN=uint16(read(app.SerMicro,52,'uint8'));
    datos_cabecera=datosRxN(1:4);
    if((datos_cabecera(1)+datos_cabecera(2))==0)
        % verifica que no haya errores
        datosRx=datosRxN;
    end

    b=[b(2:1000, :);datosRxN];

    datos_IMU1=datosRx(5:18);
    datos_IMU2=datosRx(19:32);
    datos_ADC=datosRx(33:48);
    n_medidaH=datosRx(49);
    n_medidaL=datosRx(50);

    n_medidaT=conv_uint16_double(n_medidaH,n_medidaL); % 1 valor

    Angle_Ant_IMU1=Angle_IMU1;
    [Acel_IMU1 , Gyro_IMU1 , Angle_IMU1, Temp_IMU1] = ...
        compute_datos_IMU(datos_IMU1,datos_calibracion_IMU1, ...
            Angle_Ant_IMU1); % 10 valores
    datos_nuevos_IMU1 = [Acel_IMU1 , Gyro_IMU1 , ...
        Angle_IMU1, Temp_IMU1];

    Angle_Ant_IMU2=Angle_IMU2;
    [Acel_IMU2 , Gyro_IMU2 , Angle_IMU2, Temp_IMU2] = ...
        compute_datos_IMU(datos_IMU2,datos_calibracion_IMU2,...
            Angle_Ant_IMU2); % 10 valores
    datos_nuevos_IMU2 = [Acel_IMU2 , Gyro_IMU2 , ...
```

D. Anexo III: Código – Aplicación en Matlab para la validación On-Line

```
Angle_IMU2, Temp_IMU2];

sens_ADC = ((compute_ADC(datos_ADC).*5)./1023);
    % 8 valores [ADC0 ADC1 ADC2 ADC3 ADC4 ADC5 ADC6 ADC7]

%[vect_datos_IMU1, vect_datos_IMU2, ...
vect_datos_ADC]=plot_datos_IMU(vect_datos_IMU1 , ...
datos_nuevos_IMU1,vect_datos_IMU2 , datos_nuevos_IMU2, ...
vect_datos_ADC, sens_ADC);

%% Para graficar los ángulos en términos de líneas (forma 2):

posx_IMU1=[0 7];
posy_IMU2=[0 7];

posx1_IMU1=[0 (cos(Angle_IMU1(1)*pi/180)*7)];
posx2_IMU1=[0 (cos(Angle_IMU1(2)*pi/180)*7)];
posx3_IMU1=[0 (cos(Angle_IMU1(3)*pi/180)*7)];

posy1_IMU1=[0 -(sin(Angle_IMU1(1)*pi/180)*7)];
posy2_IMU1=[0 -(sin(Angle_IMU1(2)*pi/180)*7)];
posy3_IMU1=[0 (sin(Angle_IMU1(3)*pi/180)*7)];

posx1_IMU2=[0 -(sin((Angle_IMU2(1)+90)*pi/180)*7)];
posx2_IMU2=[0 -(sin((Angle_IMU2(2)+90)*pi/180)*7)];
posx3_IMU2=[0 (sin((Angle_IMU2(3)+90)*pi/180)*7)];

posy1_IMU2=[0 (cos((Angle_IMU2(1)+90)*pi/180)*7)];
posy2_IMU2=[0 -(cos((Angle_IMU2(2)+90)*pi/180)*7)];
posy3_IMU2=[0 -(cos((Angle_IMU2(3)+90)*pi/180)*7)];
```

D. Anexo III: Código – Aplicación en Matlab para la validación On-Line

```
plot(app.Fig_EjeX, posx2_IMU1, posy2_IMU1, 'r', ...
      posx2_IMU2, posy2_IMU2, 'r'), xlim(app.Fig_EjeX, [-8,8]) ...
      , ylim(app.Fig_EjeX, [-8,8])
plot(app.Fig_EjeY, posx1_IMU1, posy1_IMU1, 'b', posx1_IMU2, ...
      posy1_IMU2, 'b'), xlim(app.Fig_EjeY, [-8,8]) , ...
      ylim(app.Fig_EjeY, [-8,8])
bar(app.Fig_Sens_ADC, sens_ADC), xlim(app.Fig_Sens_ADC, ...
      [0,9]) , ylim(app.Fig_Sens_ADC, [-0.5,5.5])

medidas_proc=[Angle_IMU1(1:2), Angle_IMU2(1:2), sens_ADC];

y_pred = app.SVM.predictFcn(medidas_proc);

switch y_pred
    case 1 % Angulo Recto
        app.Fig_Pos.ImageSource='Im_Pos_AR.jpg';
        app.Fig_Agarr.ImageSource='Im_Agarr_N.jpg';

    case 2 % Angulo Recto
        app.Fig_Pos.ImageSource='Im_Pos_AR.jpg';
        app.Fig_Agarr.ImageSource='Im_Agarr_P.jpg';

    case 3 % Angulo Recto
        app.Fig_Pos.ImageSource='Im_Pos_AR.jpg';
        app.Fig_Agarr.ImageSource='Im_Agarr_F.jpg';

    case 4 % Angulo Recto
        app.Fig_Pos.ImageSource='Im_Pos_AR.jpg';
        app.Fig_Agarr.ImageSource='Im_Agarr_A.jpg';
```

D. Anexo III: Código – Aplicación en Matlab para la validación On-Line

```
case 5 % Completamente horizontal
    app.Fig_Pos.ImageSource='Im_Pos_CH.jpg';
    app.Fig_Agarr.ImageSource='Im_Agarr_N.jpg';

case 6 % Completamente horizontal
    app.Fig_Pos.ImageSource='Im_Pos_CH.jpg';
    app.Fig_Agarr.ImageSource='Im_Agarr_P.jpg';

case 7 % Completamente horizontal
    app.Fig_Pos.ImageSource='Im_Pos_CH.jpg';
    app.Fig_Agarr.ImageSource='Im_Agarr_F.jpg';

otherwise % Completamente horizontal
    app.Fig_Pos.ImageSource='Im_Pos_CH.jpg';
    app.Fig_Agarr.ImageSource='Im_Agarr_A.jpg';

end

%-----

end

close all
clear all

end
end

% Callbacks that handle component events
```

D. Anexo III: Código – Aplicación en Matlab para la validación On-Line

```
methods (Access = private)

% Code that executes after component creation
function startupFcn(app)
    clc
    close all
    clear all

    % --- Muestra los puertos disponibles y configura uno
    disp("Puertos seriales disponibles: ")
    serialportlist("available")
    app.Inicio.Enable=1;

end

% Button pushed function: Inicio
function InicioPushed(app, event)
%       clc
%       close all
%       clear all

    app.i=1;
    app.Inicio.Enable=0;
    InicioRegistro (app)

end

% Callback function
function GrabarDatosPushed(app, event)
    write(app.SerMicro,10,'uint8')
    app.grabar=1;
    app.cuenta_grabar=1;
```

D. Anexo III: Código – Aplicación en Matlab para la validación On-Line

```
end

% Value changed function: ClasificadorDropDown
function ClasificadorDropDownValueChanged(app, event)

    load('SVM_S4_E1_P12_A1234_con_IMU_Clas_PA_CV.mat', ...
        'SVM_S4_E1_P12_A1234_con_IMU_Clas_PA_CV');
    load('SVM_S4_E12_P12_A1234_con_IMU_Clas_PA_CV.mat', ...
        'SVM_S4_E12_P12_A1234_con_IMU_Clas_PA_CV');
    load('SVM_S4_E123_P12_A1234_con_IMU_Clas_PA_CV.mat', ...
        'SVM_S4_E123_P12_A1234_con_IMU_Clas_PA_CV');
    load('SVM_S4_E1234p3_P12_A1234_con_IMU_Clas_PA_CV.mat', ...
        'SVM_S4_E1234p3_P12_A1234_con_IMU_Clas_PA_CV');
    value = app.ClasificadorDropDown.Value

    if strcmp(value, 'S4_E1_P12_A1234_con_IMU_Clas_PA_CV')
        SVM = SVM_S4_E1_P12_A1234_con_IMU_Clas_PA_CV
    elseif strcmp(value, 'S4_E12_P12_A1234_con_IMU_Clas_PA_CV')
        SVM = SVM_S4_E1234p3_P12_A1234_con_IMU_Clas_PA_CV
    elseif strcmp(value, 'S4_E123_P12_A1234_con_IMU_Clas_PA_CV')
        SVM = SVM_S4_E1234p3_P12_A1234_con_IMU_Clas_PA_CV
    else
        SVM = SVM_S4_E1234p3_P12_A1234_con_IMU_Clas_PA_CV
    end

end

end

% Component initialization
methods (Access = private)

    % Create UIFigure and components
```

D. Anexo III: Código – Aplicación en Matlab para la validación On-Line

```
function createComponents(app)

    % Create ValidacionOnLine and hide until all components ...
    % are created
    app.ValidacionOnLine = uifigure('Visible', 'off');
    app.ValidacionOnLine.Position = [100 100 804 472];
    app.ValidacionOnLine.Name = 'Validación On Line';

    % Create Fig_EjeY
    app.Fig_EjeY = uiaxes(app.ValidacionOnLine);
    title(app.Fig_EjeY, 'Eje Y')
    app.Fig_EjeY.XTick = [];
    app.Fig_EjeY.XTickLabel = '';
    app.Fig_EjeY.YTick = [];
    app.Fig_EjeY.YTickLabel = '';
    app.Fig_EjeY.XGrid = 'on';
    app.Fig_EjeY.XMinorGrid = 'on';
    app.Fig_EjeY.YGrid = 'on';
    app.Fig_EjeY.YMinorGrid = 'on';
    app.Fig_EjeY.ZGrid = 'on';
    app.Fig_EjeY.ZMinorGrid = 'on';
    app.Fig_EjeY.Box = 'on';
    app.Fig_EjeY.Position = [279 25 241 194];

    % Create Fig_EjeX
    app.Fig_EjeX = uiaxes(app.ValidacionOnLine);
    title(app.Fig_EjeX, 'Eje X')
    app.Fig_EjeX.XTick = [];
    app.Fig_EjeX.XTickLabel = '';
    app.Fig_EjeX.YTick = [];
    app.Fig_EjeX.YTickLabel = '';
    app.Fig_EjeX.XGrid = 'on';
```

D. Anexo III: Código – Aplicación en Matlab para la validación On-Line

```
app.Fig_EjeX.XMinorGrid = 'on';
app.Fig_EjeX.YGrid = 'on';
app.Fig_EjeX.YMinorGrid = 'on';
app.Fig_EjeX.ZGrid = 'on';
app.Fig_EjeX.ZMinorGrid = 'on';
app.Fig_EjeX.Box = 'on';
app.Fig_EjeX.Position = [27 25 241 194];

% Create Fig_Sens_ADC
app.Fig_Sens_ADC = uiaxes(app.ValidacionOnLine);
title(app.Fig_Sens_ADC, 'Sens ADC')
app.Fig_Sens_ADC.XTick = [1 2 3 4 5 6 7 8];
app.Fig_Sens_ADC.XTickLabel = {'1'; '2'; '3';...
    '4'; '5'; '6'; '7'; '8'};
app.Fig_Sens_ADC.YTick = [0 1 2 3 4 5];
app.Fig_Sens_ADC.YTickLabel = {'0'; '1'; '2'; ...
    '3'; '4'; '5'};
app.Fig_Sens_ADC.XGrid = 'on';
app.Fig_Sens_ADC.XMinorGrid = 'on';
app.Fig_Sens_ADC.YGrid = 'on';
app.Fig_Sens_ADC.YMinorGrid = 'on';
app.Fig_Sens_ADC.ZGrid = 'on';
app.Fig_Sens_ADC.ZMinorGrid = 'on';
app.Fig_Sens_ADC.Box = 'on';
app.Fig_Sens_ADC.Position = [536 11 249 208];

% Create Inicio
app.Inicio = uibutton(app.ValidacionOnLine, ...
    'push');
app.Inicio.ButtonPushedFcn = createCallbackFcn(app,...
    @InicioPushed, true);
app.Inicio.FontSize = 24;
```


D. Anexo III: Código – Aplicación en Matlab para la validación On-Line

```
app.Inicio.FontWeight = 'bold';
app.Inicio.Position = [111 338 142 65];
app.Inicio.Text = 'Inicio';

% Create ClasificadorDropDownLabel
app.ClasificadorDropDownLabel = uilabel(app.ValidacionOnLine);
app.ClasificadorDropDownLabel.HorizontalAlignment = 'right';
app.ClasificadorDropDownLabel.FontSize = 15;
app.ClasificadorDropDownLabel.FontWeight = 'bold';
app.ClasificadorDropDownLabel.Position = [57 248 91 22];
app.ClasificadorDropDownLabel.Text = 'Clasificador';

% Create ClasificadorDropDown
app.ClasificadorDropDown = uidropdown(app.ValidacionOnLine);
app.ClasificadorDropDown.Items = ...
    {'S4_E1_P12_A1234_con_IMU_Clas_PA_CV', ...
     'S4_E12_P12_A1234_con_IMU_Clas_PA_CV', ...
     'S4_E123_P12_A1234_con_IMU_Clas_PA_CV', ...
     'S4_E1234p3_P12_A1234_con_IMU_Clas_PA_CV'};
app.ClasificadorDropDown.ItemsData = ...
    {'S4_E1_P12_A1234_con_IMU_Clas_PA_CV', ...
     'S4_E12_P12_A1234_con_IMU_Clas_PA_CV', ...
     'S4_E123_P12_A1234_con_IMU_Clas_PA_CV', ...
     'S4_E1234p3_P12_A1234_con_IMU_Clas_PA_CV'};
app.ClasificadorDropDown.ValueChangedFcn = ...
    createCallbackFcn(app, @ClasificadorDropDownValueChanged, ..
    true);
app.ClasificadorDropDown.FontWeight = 'bold';
app.ClasificadorDropDown.Position = [163 248 357 22];
app.ClasificadorDropDown.Value = ...
    'S4_E1_P12_A1234_con_IMU_Clas_PA_CV';
```

D. Anexo III: Código – Aplicación en Matlab para la validación On-Line

```
% Create Fig_Pos
app.Fig_Pos = uiimage(app.ValidacionOnLine);
app.Fig_Pos.Position = [360 310 125 124];

% Create Fig_Agarr
app.Fig_Agarr = uiimage(app.ValidacionOnLine);
app.Fig_Agarr.Position = [595 310 124 120];

% Create PosicinLabel
app.PosicinLabel = uilabel(app.ValidacionOnLine);
app.PosicinLabel.FontWeight = 'bold';
app.PosicinLabel.Position = [395 433 56 22];
app.PosicinLabel.Text = 'Posición';

% Create AgarreLabel
app.AgarreLabel = uilabel(app.ValidacionOnLine);
app.AgarreLabel.FontWeight = 'bold';
app.AgarreLabel.Position = [635 433 44 22];
app.AgarreLabel.Text = 'Agarre';

% Show the figure after all components are created
app.ValidacionOnLine.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

% Construct app
function app = Validacion_On_Line_app

% Create UIFigure and components
```

D. Anexo III: Código – Aplicación en Matlab para la validación On-Line

```
        createComponents(app)

        % Register the app with App Designer
        registerApp(app, app.ValidacionOnLine)

        % Execute the startup function
        runStartupFcn(app, @startupFcn)

        if nargin == 0
            clear app
        end
    end

    % Code that executes before app deletion
    function delete(app)

        % Delete UIFigure when app is deleted
        delete(app.ValidacionOnLine)
    end
end
end
end
```

Bibliografía

- [1] “Sala situacional de las Personas con Discapacidad (PCD) Ministerio de Salud y Protección Social Oficina de Promoción Social,” Tech. Rep., 2017.
- [2] “Observatorio Discapacidad.” [Online]. Available: <http://rssvr2.sispro.gov.co/ObservatorioDiscapacidad/>
- [3] “Páginas - Nombre del Observatorio.” [Online]. Available: <http://ondiscapacidad.minsalud.gov.co/Paginas/Inicio.aspx>
- [4] V. Ravindra and C. Castellini, “A comparative analysis of three non-invasive human-machine interfaces for the disabled,” *Frontiers in Neurorobotics*, vol. 8, no. JAN, pp. 1–10, 2014.
- [5] E. Cho, R. Chen, L. K. Merhi, Z. Xiao, B. Pousett, and C. Menon, “Force myography to control robotic upper extremity prostheses: A feasibility study,” *Frontiers in Bioengineering and Biotechnology*, vol. 4, no. MAR, pp. 1–12, 2016.
- [6] R. N. Khushaba, A. H. Al-Timemy, A. Al-Ani, and A. Al-Jumaily, “A Framework of Temporal-Spatial Descriptors based Feature Extraction for Improved Myoelectric Pattern Recognition,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 4320, no. c, pp. 1–1, 2017. [Online]. Available: <http://ieeexplore.ieee.org/document/7886279/>

Bibliografia

- [7] A. Fougner, E. Scheme, A. D. C. Chan, K. Englehart, and Ø. Stavdahl, “Resolving the limb position effect in myoelectric pattern recognition,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, p. 8, 2011.
- [8] S. Benatti, B. Milosevic, E. Farella, E. Gruppioni, and L. Benini, “A Prosthetic Hand Body Area Controller Based on Efficient Pattern Recognition Control Strategies,” *Sensors*, vol. 17, no. 4, p. 869, 2017. [Online]. Available: <http://www.mdpi.com/1424-8220/17/4/869>
- [9] F. E. R. Mattioli, E. A. Lamounier, A. Cardoso, A. B. Soares, and A. O. Andrade, “Classification of EMG signals using artificial neural networks for virtual hand prosthesis control,” *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pp. 7254–7257, 2011.
- [10] K. Englehart and B. Hudgins, “A robust, real-time control scheme for multifunction myoelectric control,” *IEEE Transactions on Bio-Medical Engineering*, vol. 50, no. 7, pp. 848–54, 2003.
- [11] L. Pan, D. Zhang, N. Jiang, X. Sheng, and X. Zhu, “Improving robustness against electrode shift of high density EMG for myoelectric control through common spatial patterns.” *Journal of neuroengineering and rehabilitation*, vol. 12, no. 1, p. 110, 2015. [Online]. Available: <http://jneuroengrehab.biomedcentral.com/articles/10.1186/s12984-015-0102-9>
- [12] A. Fougner, O. Stavdahl, P. J. Kyberd, Y. G. Losier, and P. A. Parker, “Control of upper limb prostheses: Terminology and proportional myoelectric control a review,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 20, no. 5, pp. 663–677, 2012.
- [13] M. Ortiz-Catalan, “Cardinality as a highly descriptive feature in myoe-

Bibliografia

- lectric pattern recognition for decoding motor volition,” *Frontiers in Neuroscience*, vol. 9, no. OCT, pp. 1–7, 2015.
- [14] A. D. C. Chan and K. B. Englehart, “Continuous myoelectric control for powered prostheses using hidden Markov models,” *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 1, pp. 121–124, 2005.
- [15] N. Jiang, K. B. Englehart, and P. A. Parker, “Extracting simultaneous and proportional neural control information for multiple-dof prostheses from the surface electromyographic signal,” *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 4, pp. 1070–1080, 2009.
- [16] E. J. Earley, A. A. Adewuyi, and L. J. Hargrove, “Optimizing pattern recognition-based control for partial-hand prosthesis application,” *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference*, vol. 2014, pp. 3574–3577, 2014.
- [17] G. W. Favieiro, K. O. A. Moura, and A. Balbinot, “Novel method to characterize upper-limb movements based on paraconsistent logic and myoelectric signals,” *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, no. 3, pp. 395–398, 2016. [Online]. Available: <http://ieeexplore.ieee.org/document/7590723/>
- [18] L. Hargrove, Y. Losier, B. Lock, K. Englehart, and B. Hudgins, “A real-time pattern recognition based myoelectric control usability study implemented in a virtual environment,” *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, pp. 4842–4845, 2007.
- [19] Z. Lu, K.-y. Tong, H. Shin, S. Li, and P. Zhou, “Advanced Myoelectric Control for Robotic Hand-Assisted Train-

Bibliografía

- ning: Outcome from a Stroke Patient,” *Frontiers in Neurology*, vol. 8, no. March, pp. 1–5, 2017. [Online]. Available: <http://journal.frontiersin.org/article/10.3389/fneur.2017.00107/full>
- [20] L. J. Hargrove, B. A. Lock, and A. M. Simon, “Pattern recognition control outperforms conventional myoelectric control in upper limb patients with targeted muscle reinnervation,” *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pp. 1599–1602, 2013.
- [21] C. Lauretti, A. Davalli, R. Sacchetti, E. Guglielmelli, and L. Zollo, “Fusion of M-IMU and EMG signals for the control of trans-humeral prostheses,” *Proceedings of the IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics*, vol. 2016-July, pp. 1123–1128, 2016.
- [22] T. A. Kuiken, G. A. Dumanian, R. D. Lipschutz, L. A. Miller, and K. A. Stubblefield, “Targeted muscle reinnervation for improved myoelectric prosthesis control,” *2nd International IEEE EMBS Conference on Neural Engineering*, vol. 2005, pp. 396–399, 2005.
- [23] D. Ferigo, L. K. Merhi, B. Pousett, Z. G. Xiao, and C. Menon, “A Case Study of a Force-myography Controlled Bionic Hand Mitigating Limb Position Effect,” *Journal of Bionic Engineering*, vol. 14, no. 4, pp. 692–705, 2017.
- [24] “Aprendizaje Automático — Coursera.” [Online]. Available: <https://www.coursera.org/learn/machine-learning?>
- [25] A. Mu, *Statistical Pattern Recognition*, 2019, vol. 53, no. 9.