

JUAN SEBASTIAN CABRERA HIDALGO



Conjunto normalizado de interfaces para los
componentes de un simulador virtual de
prótesis de mano robótica

Universidad Del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Maestría en Electrónica y Telecomunicaciones

Popayán
2009

JUAN SEBASTIAN CABRERA HIDALGO

Conjunto normalizado de interfaces para los
componentes de un simulador virtual de prótesis
de mano robótica

Tesis presentada a la Facultad de Ingeniería
Electrónica y Telecomunicaciones de la
Universidad del Cauca para la obtención del
Título de

Magíster en
Electrónica y Telecomunicaciones.

Director:
Mag. Héctor Fabio Jaramillo Ordoñez

Popayán
2009

Nota de aceptación:

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Popayán, diciembre de 2009

Dedico mi Tesis con todo mi amor y cariño a mi linda esposa Cecilia. Debido a sus ideas, colaboración y continuo apoyo este trabajo pudo hacerse realidad.

Agradecimientos

Agradezco a mi director de tesis, el Mág. Hector Fábio Jaramillo, por la confianza que tuvo en mí y sus acertadas sugerencias que contribuyeron enormemente al exitoso desarrollo del presente proyecto.

Un Especial agradecimiento a todo el Grupo de Investigación en Automática Industrial de la Universidad del Cauca por su continua colaboración y apoyo.

Igualmente, un sincero agradecimiento al grupo BIOECI de la Escuela Colombiana de Ingeniería.

Por último, a mi madre por sus oraciones y sus palabras de aliento que permitieron este logro tan importante en mi vida.

Resumen

Las dificultades que enfrenta una persona amputada son desoladoras, desde su propia autoestima al sentirse "menos-válido", hasta la consecución de un trabajo desde el cual pueda sostenerse a sí mismo y a su familia; en Colombia debido a los conflictos armados muchas personas (civiles y militares) son víctimas de amputaciones que reducen su autonomía en la sociedad, dificultando aún más las condiciones de vida en nuestro país.

En el mundo muchos centros de investigación y empresas privadas realizan estudios y desarrollos para brindar opciones de una mejor vida a las personas amputadas a través del uso de prótesis robóticas.

Entre los dispositivos prostéticos al que mayor atención se le ha prestado es la prótesis de mano, ya que nuestras manos son nuestra principal herramienta de acción creativa; sin embargo las soluciones que existen suelen ser muy costosas para nuestra sociedad, y gran parte de este costo se debe a la falta de reutilización de los avances alcanzados por otros en el área, es así como cada solución a pesar de tener una arquitectura de base similar su implementación es propietaria y suele mantener un alto acoplamiento entre los módulos de la arquitectura utilizada.

Con el presente proyecto se busca entonces contribuir a la solución del siguiente problema:

¿Cómo integrar los componentes del simulador de una prótesis de mano robótica, de tal forma que esa integración sea una tarea efectiva y debidamente documentada, con el fin de que soporte un desarrollo sostenible y permita su evolución?

Para dar respuesta a esta pregunta se evidencio la necesidad de conformar un conjunto normalizado de interfaces para los componentes de un simulador de

prótesis de mano robótica el permitiría un desarrollo sostenible y comprensible del mismo, de tal forma que si se hace necesario cambiar total o parcialmente la estrategia de control en alguno de los componentes se realice teniendo en cuenta las interfaces previamente establecidas, sin causar inconvenientes al funcionamiento del resto de la aplicación.

Por lo tanto, la presente monografía de tesis propone un conjunto normalizado de interfaces para los módulos de una arquitectura general de prótesis de mano, donde se incluyen las características lógicas y temporales de cada interfaz. Como caso de estudio se plantea su verificación a través de la implementación de un entrenador de prótesis de mano. Todo lo anterior con el fin de brindar una solución a los problemas de acoplamiento, reutilización y mantenibilidad de los diferentes módulos de la arquitectura para un dispositivo prostético de mano.

Por último, con base en diferentes propuestas para normalización de interfaces a nivel software, aplicando directrices de desarrollo hardware/software y aprovechando los avances del Grupo de Investigación en Automática Industrial, el presente trabajo realiza un aporte a la base del conocimiento en el área de la bioingeniería, particularmente, en la construcción de dispositivos prostéticos de forma que se estimula el desarrollo de soluciones modulares para este tipo de proyectos.

Palabras Claves: Arquitectura, Interfaz, Entrenador, Prótesis, Mano.

Abstract

The difficulties that an amputee hand are grim, from their self-esteem to feel like a handicapped person until the achievement of a job from which he or she can support himself and his family. In Colombia because of armed conflict, many people like civilian and military are amputees reducing their autonomy in society and making it more difficult living in our country.

Among the prosthetic devices that more attention has been paid is the prosthetic hand. Since our hands are our main tool for creative action. However, the existing solutions are often very expensive to our society and much of this cost is due to the lack of reuse of the progress made by others in the area, although each solution have a similar basic architecture, this architecture have a proprietary implementation and usually maintains a high coupling between its modules.

This project seeks to help resolve the following question:

How integrate the components of the simulator of robotic hand prosthesis, so that integration becomes an effective and properly documented task, with the aim of supporting sustainable development and enable its evolution?

To answer this question we evidenced the need to establish a standard set of interfaces for the components of a simulator of the robotic hand prosthesis would allow a sustainable and comprehensive development, so, if necessary at some time to change all or part of the components control strategy, then, this activity should be conducted by considering the previously established interfaces, without causing inconvenience to the rest of the application.

Therefore this monograph thesis proposes a standard set of interfaces for modules in a general architecture of prosthetic hand which includes the logical and temporal characteristics of each interface, and as a study case raises verification

through implementation of hand prosthesis coach, all the above in order to provide a solution to the problems of coupling, reuse and maintainability of the various modules of the architecture for a prosthetic hand device.

Based on different proposals for standardization of interfaces to standard software development guidelines applying hardware / software, and building on the progress of the Research Group in Industrial Automation, this paper contributes to the knowledge base in the area, so as to stimulates the development of modular solutions for these projects.

Key words: Architecture, Interface, Prosthesis, Hand, Robotic.

Contenido

	Pág.
Lista de Tablas	xvii
Lista de Figuras.....	xix
Lista de Abreviaturas.....	xxi
1 Introducción	23
1.1 Justificación.....	23
1.2 Contribución de la Tesis.....	24
1.3 Organización de la Tesis.....	26
2 Base Inicial de Conocimiento	27
2.1 Arquitectura para dispositivos prostéticos	27
2.2 Adquisición y procesamiento de la señal electromiográfica	28
2.2.1 Electromiografía	29
2.2.2 Sensores de EMGS	30
2.2.3 Digitalización	32
2.2.4 Análisis de la señal EMG.....	33
2.2.5 Reducción de dimensionalidad.....	35
2.2.6 Reconocimiento de patrones	35
2.2.7 Intenciones de movimiento.....	38
2.3 Sistemas de control para prótesis de mano	41
2.4 Realimentación al usuario	42

3	Conjunto normalizado de interfaces	45
3.1	Soporte teórico para la definición de interfaces.....	45
3.1.1	Interfaz Software	46
3.1.2	Interfaz Hardware.....	47
3.1.3	Estándar para la definición y documentación de Interfaces.....	49
3.1.4	Implementación de Interfaz hardware/software	51
3.2	Modelo de análisis del conjunto de interfaces Hardware/Software para simulador de prótesis de mano robótica.....	52
3.2.1	Modulo Gestor Analógico (MGA)	55
3.2.2	Diagrama UML del Conjunto Normalizado de Interfaces . ¡Error! Marcador no definido.	
3.2.3	Módulo Gestor Digital (MGD).....	58
3.2.4	Módulo Caracterizador de Señal (MCS).....	63
3.2.5	Módulo Gestor de Reconocimiento de Intención de Movimiento (MGRIM).....	74
3.2.6	Módulo Controlador del Dispositivo (MCD).....	80
3.2.7	Módulo Gestor Sensorial (MGS).....	88
3.2.8	Módulo Gestor de Realimentación (MGR).....	94
4	Verificación del Conjunto de Interfaces.....	97
4.1	Responsabilidades del conjunto normalizado de interfaces	97
4.2	Descripción del simulador virtual	99
4.2.1	Módulo Gestor Analógico (MGA)	101
4.2.2	Módulo Gestor Digital (MGD).....	102
4.2.3	Módulo Caracterizador de la Señal (MCS)	103
4.2.4	Módulo Gestor de Reconocimiento de Intención de Movimiento (MGRIM).....	103
4.2.5	Módulo Controlador del Dispositivo (MCD).....	104
4.2.6	Módulo Gestor Sensorial (MGS)	105
4.2.7	Módulo Gestor de Realimentación (MGR).....	105
4.2.8	Dispositivo protésico Virtual	105
5	Conclusiones, Aportes y Trabajos Futuros.....	109

5.1	Conclusiones.....	109
5.2	Aportes del proyecto	111
5.3	Trabajos Futuros	113

Anexos

A.1	Control del par del motor D.C.....	125
A.2	Tipo de Datos en Sistemas Embebidos.....	129
A.3	Lenguaje de Definición de Interfaces IDL.....	131
A.4	Características Espectrales.....	133
A.4.1	Transformada de Fourier.....	133
A.4.2	Coeficientes Cepstrales	135
A.4.3	Transformada Wavelets	135
A.4.4	Coeficiente Wavelets.....	136
A.5	Actor, Entidad Control, Interfaz.....	139
A.5.1	Actor.....	139
A.5.2	Entidad.....	139
A.5.3	Control.....	140
A.5.4	Interfaz	140

Lista de Tablas

	Pág.
Tabla 3.1 Plantilla para documentación de interfaces	51
Tabla 3.2 MGA, Sensado de Fuente de Señal, Obtener Señal.....	57
Tabla 3.3 MGA, Gestor Analógico, Obtener señal pre-procesada analógicamente ..	58
Tabla 3.4 MGD, Gestor Digital SW HW, Filtro digital. Obtener muestra de señal	62
Tabla 3.5. MGD, Gestor Digital SW HW, Obtener arreglo de muestras de la señal..	62
Tabla 3.6. MCS, Caracterizador de Señal SW, Obtener parámetros Temporales Básicos.....	66
Tabla 3.7. MCS, Caracterizador de Señal SW, Obtener valor RMS.....	67
Tabla 3.8. MCS, Caracterizador de Señal SW, Obtener la varianza	67
Tabla 3.9. MCS, Caracterizador de Señal SW, Obtener duración.....	68
Tabla 3.10 MCS, Caracterizador de Señal SW, <i>Obtener contador de inflexión</i>	68
Tabla 3.11. MCS, Caracterizador de Señal SW, Obtener longitud de onda.....	69
Tabla 3.12. MCS, Caracterizador de Señal SW, Obtener contador de cruce por cero	69
Tabla 3.13. MCS, Caracterizador de Señal SW, Obtener coeficientes AR	70
Tabla 3.14 MCS, Caracterizador de Señal SW, Obtener parámetros de frecuencia básicos	71
Tabla 3.15. MCS, Caracterizador de Señal SW, Obtener parámetros de energía básicos	72
Tabla 3.16. MCS, Caracterizador de Señal SW, Obtener parámetros de potencia básicos	72

Tabla 3.17. MCS, Caracterizador de Señal SW, Obtener coeficientes Wavelets	73
Tabla 3.18. Tipo de agarre y valor asignado.	78
Tabla 3.19. MGRIM, Movimiento HW SW. Obtener tipo de agarre.	78
Tabla 3.20. Movimientos de la muñeca	79
Tabla 3.21. MGRIM. Gestor de Reconocimiento de Intensión de Movimiento. Movimiento HW SW .Obtener tipo de movimiento de la muñeca.	79
Tabla 3.22. MCP.Controlador de prótesis. Obtener torque de los dedos	86
Tabla 3.23. MCP. Controlador de prótesis. Obtener ángulo de los dedos.....	87
Tabla 3.24. MGS. Gestor sensorial HW. Realimentación Sensorial HW. Obtener posición de los dedos	89
Tabla 3.25. MGS. Gestor sensorial HW. Realimentación Sensorial HW. Obtener posición extrema de los dedos	90
Tabla 3.26. MGS. Gestor sensorial. . Realimentación Sensorial HW. Obtener sobrecarga de los dedos	91
Tabla 3.27. MGS. Gestor sensorial. Realimentación Sensorial HW. Obtener fuerza ejercida por los dedos	92
Tabla 3.29. MGS. Gestor sensorial HW. Realimentación Sensorial HW. Obtener el valor del deslizamiento de los dedos	93
Tabla 3.28. MGS. Gestor sensorial. Realimentación Sensorial HW. Obtener fuerza ejercida por la palma de la prótesis	93
Tabla 3.30. MGR. Gestor de Realimentación, realimentación, Obtener señal de realimentación	95
Tabla 4.1 Comparación entre algunos de los sistemas operativos reconocidos en el mercado.....	100

Lista de Figuras

	Pag.
Figura 2.1. Arquitectura de control formal multifuncional de una mano artificial propuesta por Ph.D. Zecca.....	27
Figura 2.2 Esquema del amplificador diferencial.....	31
Figura 2.3. Clasificación de agarres de mano	39
Figura 2.4 Sistema de control.....	41
Figura 3.1 Diagrama UML del conjunto normalizado de interfaces para los componentes de un simulador virtual de prótesis de mano robótica.....	54
Figura 3.2 Modelo de análisis en UML del Modulo Gestor Analógico.	55
Figura 3.3. Modelo de análisis del módulo de digitalización y filtraje digital en UML.	60
Figura 3.6. Modelo de análisis del Módulo Caracterizador de Señal.....	64
Figura 3.7 Representación frecuencia media, mediana y potencia máxima	71
Figura 3.8 Movimientos Muñeca	75
Figura 3.9 Modelo de análisis en UML del Módulo Gestor de Reconocimiento de Intención de Movimiento (MGRIM).....	77
Figura 3.10. Anatomía de la mano humana	80
Figura 3.11. Esquema básico de control realimentado	83
Figura 3.12 Modelo de análisis en UML del Módulo controlador del Dispositivo.....	85
Figura 3.13. Modelo de análisis en UML del Módulo Gestor Sensorial	89

Figura 3.14. Modelo de análisis en UML del Módulo Gestor de Realimentación (MGR)	95
Figura 4.1. Arquitectura del simulador virtual	99
Figura 4.2. Electrodoes superficiales	101
Figura 4.3 Alternativa de diseño para implementar la interfaz <i>GetSxAnalogProcessed</i>	102
Figura 4.4 Ilustración de movimiento de apertura y cierre de mano	106
Figura 4.5 Modelo esquelético del dispositivo prótesis virtual	106
Figura 4.6. Código ejemplo para la descripción de interfaces en lenguaje C.	107
Figura 3.4. Obtención de los coeficientes wavelet	137
Figura A.4.5. Proceso de filtrado	137

Lista de Abreviaturas

CORBA	Common Object Request Broker Architecture.
ECG	Electro Cardiográfico
EMG	Electro Mio Gráfica.
EMGS	Electro Mio Gráfica Superficial.
FES	Funtional Electrical Stimulus.
FSR	Force Sensing Resistor.
HDL	Hardware Description Language.
ME	Mio Eléctrica.
MIDL	Microsoft Interface Description language
OMG	Object Management Group.
PZT	Piezoelectric Transducer.
SX	Señal o en Ingles, Signal.
UML	Unified Modeling Language
IDL	Interface Description language.
WSDL	Web Service Description Language.
LCM	Módulo de control de bajo nivel (LowLevel Controlling Module)
SPM	Módulo de Procesamiento Sensorial (Sensory Processing Module)
SFM	Módulo de Retroalimentación Sensorial (Sensory Feedback Module)
STFT	Short time Fourier Transform
DWT	Discrete Wavelet Transform
WPT	Wavelet Packet Transform
ANN	Artificial Neural Networks
DV	Dedo Virtual
TTL	Transistor-Transistor Logic
PID	Proporcional Integral Derivativo

A/D	Analógico/Digital
USB	Universal Serial Bus
SCSI	Small Computers System Interface
I²C	Inter-Integrated Circuit
SPI	Serial Peripheral Interface
UART	Universal Asynchronous ReceiverTransmitte
MGA	Módulo Gestor Analógico
MGD	Módulo Gestor Digital
MCS	Módulo Caracterizador de Señal.
MGRIM	Módulo Gestor de Reconocimiento de Intención de Movimiento
MCD	Módulo Controlador del Dispositivo.
MCP	Articulación Metacarpofalange.
MGR	Módulo Gestor de Realimentación
MGS	Módulo Gestor Sensorial
FIR	Finite Impulse Response
IIR	Infinite Impulse Response
AR	Auto-Regresivo
ARMA	Auto-Regressive Mobile Average
LMS	Least Mean Square algorithm
RMS	Recursive Least Squares algorithm
FFT	Fast Fourier Transform
FES	Functional Electrical Stimulation
PWM	Pulse Width Modulation

Capítulo 1

1 Introducción

Con el presente proyecto se busca a la solución del siguiente problema:

¿Cómo integrar los componentes del simulador de una prótesis de mano robótica, garantizando la modularidad del sistema y de tal forma que esa integración sea una tarea efectiva, debidamente documentada, que soporte un desarrollo sostenible y permita su evolución?

Para dar respuesta a esta pregunta, se evidencia la necesidad de la aplicación de conceptos enmarcados en la ingeniería del software a arquitecturas de sistemas robóticos, consolidándose como un escenario multidisciplinario que vale la pena estudiar.

1.1 Justificación

Un conjunto normalizado de interfaces para los componentes de un simulador de prótesis de mano robótica permitiría un desarrollo sostenible y comprensible del mismo, de tal forma que si se hace necesario cambiar total o parcialmente la estrategia de control en alguno de los componentes se realice teniendo en cuenta las interfaces previamente establecidas, sin causar inconvenientes al funcionamiento de los demás componentes. Esto facilitaría la integración y depuración de los

componentes hardware/software que conforman el simulador de prótesis de mano robótica y permitiría llevar de una manera metódica un sistema de control de versiones, el cual registre los cambios realizados con miras a la puesta a punto, mantenimiento y evolución del sistema.

En general este trabajo servirá como marco de referencia para desarrollos e investigaciones futuras en el área de la simulación de las prótesis robóticas, el cual sería un importante aporte al conocimiento científico global en sistemas biónicos. Específicamente, el conjunto normalizado de interfaces permitirá a posteriores iniciativas en el área optimizar la fase de integración en el proceso de desarrollo de un simulador de prótesis robótica, así como la creación de una base teórica para aplicaciones en otras áreas.

Cumpliendo su objetivo de responsabilidad social, este proyecto se enfoca en lograr soluciones accesibles a poblaciones de bajos recursos. Por otro lado, desde la perspectiva tecnológica se enmarca en la optimización de algoritmos de control y procesamiento de señales, dado que se pretende realizar un aporte en el campo de las arquitecturas de integración de componentes hardware/software al proponer un conjunto normalizado de interfaces para los componentes de una prótesis de mano y especificar los principales criterios para su diseño.

Las especificaciones precisas de la interfaz del componente son importantes debido a que los desarrolladores de éste deben escribir código y/o diseñar hardware utilizando los servicios de otros componentes antes de que éstos se implementen. Las especificaciones de la interfaz suministrarán información a los desarrolladores del componente para que conozcan qué servicios están disponibles en otros componentes y cómo pueden acceder a ellos. Las especificaciones claras y no ambiguas de la interfaz del componente reducen las oportunidades de malas interpretaciones entre componentes que suministran un servicio y los que podrían utilizar ese servicio (Montilva, y otros, 2003)

1.2 Contribución de la Tesis

Los trabajos previos identificados en la revisión bibliográfica se enfocan en realizar un diagrama arquitectónico de bloques sin efectuar una definición adecuada

entre las interfaces de los componentes hardware/software. Esta tesis de maestría, partiendo de la arquitectura definida por el proyecto “Prótesis robótica para amputados de mano”, realiza una adecuada definición de las interfaces, especificando las responsabilidades de cada componente, el tipo y formato de datos en que se transmiten los mensajes y las restricciones temporales para su transmisión y respuesta, de forma que faciliten su distribución y reutilización.

Otro aporte importante del proyecto se hace al Grupo de Investigación en Automática Industrial de la Universidad del Cauca, particularmente en el proyecto: “Prótesis robótica para amputados de mano”, ya que contribuye a la fase de integración de los componentes hardware/software derivados del mismo.

Una adecuada investigación de un conjunto de interfaces para los componentes de una prótesis robótica realiza un significativo aporte a la manera de realizar documentación de proyectos relacionados con electrónica y fomenta la adopción de buenas prácticas de análisis y diseño de sistemas electrónicos modulares.

Particularmente, al interior de la Universidad del Cauca, al aprovechar las ventajas propuestas por la ingeniería de software se permite que en futuros proyectos de gran magnitud donde su desarrollo se plantee de manera modular, por componentes, se cuente con un referente que permita la organización y estructuración del trabajo por medio de un acuerdo formal en el establecimiento de las responsabilidades, garantizando su comprensión, evolución y mantenimiento.

Cabe resaltar que uno de los productos derivados de este proyecto es un entrenador virtual de prótesis de mano robótica el cual podrá ser utilizado por pacientes de escasos recursos, específicamente, del Centro Integral de Rehabilitación de Colombia CIREC.

1.3 Organización de la Tesis

Capítulo 2. -Base inicial de conocimiento. En este capítulo se registra en forma sistemática el conocimiento identificado que está relacionado con la simulación de prótesis de mano robóticas. Este capítulo está constituido por una síntesis de cada uno de los núcleos temáticos definidos.

Capítulo 3. -Conjunto normalizado de interfaces. En este capítulo se especifica el conjunto normalizado de interfaces para un simulador de prótesis de mano robótica. Se documentarán cada interfaz y se explicará la forma de interacción entre cada uno de los componentes que conforman el simulador.

Capítulo 4. -Validación del conjunto de interfaces. En este capítulo se documenta los resultados de la integración de los componentes del simulador virtual de prótesis de mano robótica, usando el conjunto normalizado de interfaces propuesto.

Capítulo 5. -Conclusiones y recomendaciones. Este capítulo contiene el balance de los resultados obtenidos por el proyecto y los aspectos más importantes relacionados con su desarrollo. Se incluyen también las sugerencias para trabajos futuros

Capítulo 2

2 Base Inicial de Conocimiento

2.1 Arquitectura para dispositivos prostéticos

En la tesis doctoral de (Zecca, 2003) desarrollada en Scuola Superiore Sant'Anna y apoyada por los gobiernos de Italia y Japón en el marco del programa ejecutivo 2002 - 2006 para la cooperación internacional entre dichos países, se propuso un esquema de control formal multifuncional de una mano artificial, ver Figura 2.1.

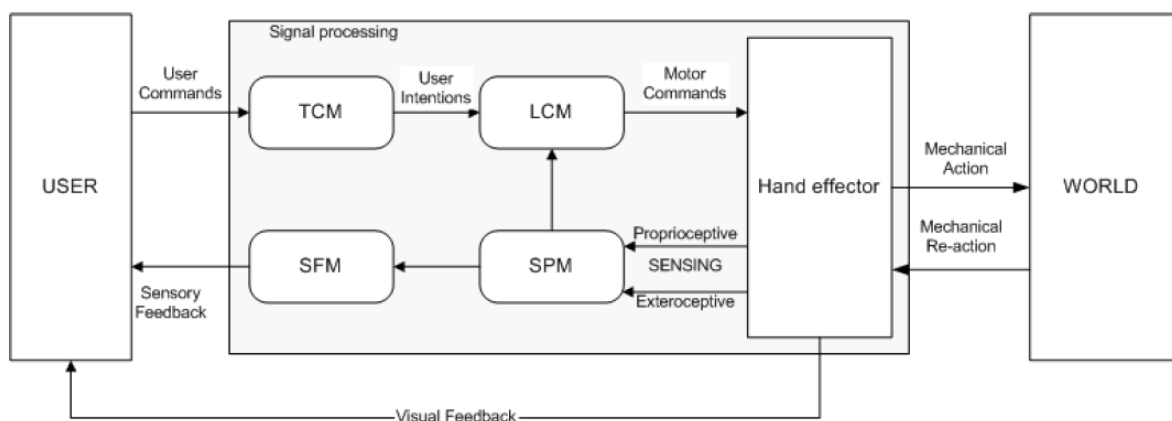


Figura 2.1. Arquitectura de control formal multifuncional de una mano artificial propuesta por Ph.D. Zecca

A continuación, se describe abreviadamente los componentes del sistema de control propuesto:

TCM - Módulo de control de alto nivel (Top-Level Controlling Module)

Este módulo adquiere la señal de entrada desde el usuario y genera los comandos de control o intenciones de movimiento que serán las señales de entrada para el módulo de control de bajo nivel.

LCM - Módulo de control de bajo nivel (Low-Level Controlling Module)

Este módulo recibe como entradas la señal generada por el TCM y las señales provenientes del Módulo de Procesamiento Sensorial (SPM), para controlar los movimientos de la mano y la fuerza de agarre.

SPM - Módulo de Procesamiento Sensorial (Sensory Processing Module)

Este módulo adquiere la señal proveniente de los sensores localizados en la mano; convierte y entrega la información al módulo LCM cerrando el ciclo de control de la prótesis de mano.

SFM - Módulo de Retroalimentación Sensorial (Sensory Feedback Module)

Este módulo adquiere la información generada por el SPM y la presenta al usuario utilizando una codificación apropiada.

Los anteriores módulos descritos por Zecca, son fácilmente reconocibles en otras investigaciones relacionadas con el tema. Entre dichas investigaciones se encuentran (Alonso, y otros, 2002), (Sebelius, y otros, 1998), (Ferguson, y otros, 2002), (Hauschild, y otros, 2007), (Farrell, y otros, 2007), (Birkedal, y otros). Los módulos no necesariamente deberían estar todos presentes en un dispositivo de prótesis sensorial debido a que su uso es de acuerdo a la complejidad del dispositivo protésico a implementar.

Teniendo como referencia los módulos del esquema de control propuesto en (Zecca, 2003) a continuación se presenta una síntesis del estado del arte de las prótesis de mano.

2.2 Adquisición y procesamiento de la señal electromiográfica

La Adquisición y procesamiento de la señal electromiográfica hace parte del módulo de control de alto nivel especificado en (Zecca, 2003). Otros autores han denominado a este módulo acondicionamiento y procesamiento (Stegeman, y otros, 2007), cabecera analógica y sistema de adquisición de datos (Alonso, y otros, 2002), captación de la señal, rectificación y filtrado (Sebelius, y otros, 1998), módulo de adquisición y pre-procesamiento (Birkedal, y otros), entre otras. Al igual que (Zecca, 2003) y (Romo, y otros, 2007), los anteriores autores coinciden en que las señales Electromiográficas Superficiales o EMGS son una fuente de información muy apropiada para el control de dispositivos prostéticos.

2.2.1 Electromiografía

La electromiografía según (Zecca, 2003) es la disciplina que trata la detección, el análisis y el uso de señales eléctricas que emiten las contracciones musculares. Esta señal, referida como señal electromiográfica (EMG), es un término más apropiado en el pasado que en el presente. En el pasado la única manera para capturar la señal para un estudio subsecuente era obteniendo una representación gráfica. Hoy en día es posible almacenar la señal en una cinta magnética, en un disco compacto o en dispositivos electrónicos. Un número creciente de profesionales en el área ha adoptado el término de señal mioeléctrica o señal ME, aunque el término EMG es aún muy usado sobre todo en el ámbito clínico.

El problema paradójico con las señales EMG es que son unas de las señales electro-fisiológicas más fáciles de medir, pero también unas de las más difíciles de interpretar cuantitativamente. Con un osciloscopio y dos objetos metálicos conectados a la entrada, es posible registrar señales EMG. Según (De Luca, 2006) "Para su detrimento, la electromiografía es tan fácil de usar y consecuentemente tan fácil de abusar".

La electromiografía puede clasificarse en dos tipos según los electrodos y la técnica de grabación aplicada: EMG de aguja, en Inglés Needle EMG y de superficie o EMGS.

El tipo de EMG utilizada en la práctica clínica depende de la clase de estudios a realizar: La EMG de aguja es utilizada cuando se sospecha la presencia de trastornos miógenos y neurógenos y es primordial un estudio cuidadoso de la señal

evaluada. La técnica de EMGS se utiliza únicamente en clínica en estudios de velocidad de conducción aunque ha sido fuertemente difundida en el campo de la rehabilitación donde sirve como señal de control electrónico.

2.2.2 Sensores de EMGS

El término sensor en lugar de electrodo, se utiliza para subrayar el hecho de que cada señal EMGS necesita de la medición de un conjunto de al menos dos electrodos que a menudo se integra junto al pre-amplificador en el mismo encapsulado (Stegeman, y otros, 2007). Según (De Luca, 2007) un electro es la superficie metálica del sensor la cual realiza el contacto eléctrico con la piel, también conocido como sensor pasivo. Por otro lado, un sensor activo es la unidad que contiene los electrodos y la circuitería electrónica para el amplificador diferencial de las señales EMG y reduce la contribución de varias fuentes de ruido. A este tipo de sensores también se los conoce como sensores activos.

La primera cuestión concierne a la configuración de los electrodos. Debido a que la señal EMG es de baja amplitud respecto a otras señales del ambiente sobre la superficie de la piel es necesario y conveniente detectarla con una configuración diferencial, ver Figura 3.8. Es decir, utilizar dos superficies de la piel con el fin de detectar su diferencia de potencial y a este resultado posteriormente amplificarlo.

El arreglo diferencial actúa como un filtro pasa banda para la señal vista desde la superficie de detección. Por otra parte, en (De Luca, 1997) muestra como la distribución de frecuencias en el espectro es afectada por la distancia de separación entre las superficies.

Para la adquisición de las señales EMG es muy común utilizar las aplicaciones desarrolladas para el registro de señales electro-cardiográficas o ECG. En (Analog Device, 1999) se presenta una aplicación para la elaboración de sensores activos de señales ME basados en el amplificador de instrumentación AD620 y utilizando electrodos desechables como los fabricados por (ConMed, 2009) y 3M Health Care (Care, M. H., 2009). Las empresas Delsys (Delsys, 2009) y B&L Engineering (B & Engineering, 2009), ofrecen sensores activos especialmente diseñados para aplicaciones en EMG; este tipo de sensores no son desechables. Para ver las ventajas y desventajas de la utilización de uno u otro tipo de sensor referirse a (De Luca, 2006).

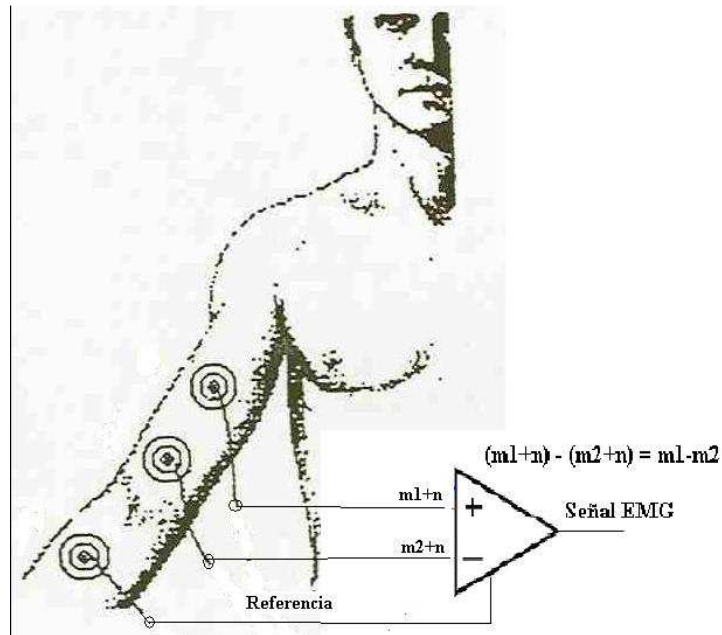


Figura 2.2 Esquema del amplificador diferencial.

Actualmente se ha establecido un acuerdo sobre la configuración de la superficie de detección y las dimensiones de un electrodo, a través del proyecto SENIAM (Stegeman, y otros, 2007), el cual tiene como objetivo principal integrar las investigaciones básicas y aplicadas sobre señales electromiográficas de superficie, particularmente en el territorio Europeo con el fin de establecer una cooperación Europea y resolver los principales temas que actualmente impiden un útil intercambio de datos y experiencia clínica. Particularmente, en (De Luca, 1997) se realizan algunas recomendaciones que se presentan a continuación.

- Configuración de los electrodos diferenciales:
 1. Las superficies de detección consisten de dos barras paralelas, cada una de 1.0 cm de largo y entre 1 y 2 mm de ancho, con 1.0 cm de separación.
 2. El ancho de banda de 20 - 500 Hz con un roll-off de al menos 12dB por octava.
 3. Tasa de rechazo de modo común CMRR >80dB.
 4. Ruido <2 uV en la banda de 20 - 400Hz.
 5. Impedancia de entrada >100 Mega Ohms.

Localizar los electrodos en el centro del vientre muscular, entre la unión miotendinosa y la zona de la inervación más cercana, con las superficies de detección orientadas paralela a la longitud de las fibras musculares.

Utilizar el valor RMS de la señal para medir la amplitud de la señal EMG provocadas voluntariamente.

2.2.3 Digitalización

La introducción del procesamiento digital en el análisis de señales EMGS sugiere su digitalización mediante un proceso de muestreo. De acuerdo con (Stegeman, y otros, 2007) alrededor del 95% de la potencia proveniente de armónicos se localiza por encima de los 400Hz, y la mayoría del porcentaje restante es generado por los electrodos y ruido proveniente de equipos electrónicos cercanos a la muestra.

La frecuencia de corte es usualmente escogida cercana a los 500Hz por lo tanto la frecuencia de muestreo debe ser al menos 1000 muestras/segundo¹.

Para el análisis de las señales, lo más común es segmentar los registros en segmentos de longitud de 256 muestras, es decir 256 ms de grabación y aplicar las técnicas de extracción de características a cada segmento (Romo, y otros, 2007).

Los requerimientos de digitalización de la señal EMG dependen de tamaño de las amplitudes esperadas. La señal ME capturada a través de electrodos de plata/cloruro de plata varía entre 0 y 6mV (Ferguson, y otros, 2002). Por lo tanto la detección de detalles en bajas amplitudes está limitada por el nivel de ruido del sistema amplificador. Los amplificadores modernos tienen un nivel de ruido del orden de unos pocos milivoltios, por lo tanto una digitalización al rededor de $0.5mV/bit$ es suficientemente precisa (Stegeman, y otros, 2007). Un conversor de 16 bits, tiene $2^{16} = 65536$ posibles niveles; esto significa que sin una etapa de adaptación de ganancia, el rango de medida con un conversor de este tipo es +16mV, el cual es suficiente para la mayoría de aplicaciones.

¹ Teorema de muestreo: el muestreo debe ser al menos 2 veces más grande que la frecuencia más alta de la señal.

2.2.4 Análisis de la señal EMG

En algunas investigaciones como (Romo, y otros, 2007), (Sebelius, y otros, 1998), (Ferguson, y otros, 2002), (Farrell, y otros, 2007), (Englehart, y otros, 1999) a este componente le denominan: Caracterización y Reconocimiento de Patrones. En miras a realizar una contribución en esta área, no es posible ir al detalle respecto al procesamiento de la señal EMG superficial luego de que ha sido digitalizada. Las posibilidades son virtualmente ilimitadas (Stegeman, y otros, 2007). Sin embargo, la mayoría de aplicaciones usan como características principales de la señal ME las siguientes:

Análisis temporal

Las características basadas en estadísticos temporales se evalúan rápidamente, y son extraídas de cada segmento de tiempo para crear un conjunto de características que representan el patrón de la señal EMGS (Romo, y otros, 2007). Entre las características se encuentran:

1. Promedio o media de la señal rectificada.
2. Valor del área bajo la curva de la señal ME o Integración de la señal ME.
3. Valor raíz cuadrática media o *rms*.
4. Valor medio absoluto de la señal.
5. Valor medio absoluto de pendientes.
6. Longitud de la forma de onda.
7. Coeficientes modelo paramétrico auto regresivo, AR.

En el dominio del tiempo, existen dos parámetros que son principalmente utilizados: En (De Luca, 1997) se muestra el cálculo de la raíz cuadrática media o *rms* y el valor rectificado promedio. Ambos son apropiados y proveen una medida muy útil de la amplitud de la señal.

Análisis espectral

La representación tiempo-frecuencia de la señal provee información de la distribución de su energía en los dos dominios, lográndose una descripción más completa del fenómeno físico (Romo, y otros, 2007). Sin embargo, una

transformación de la señal implica un costo computacional adicional y un espacio de características de mayor dimensión. Entre las características se tiene:

1. Mediana, media y moda del espectro de frecuencia.
2. Ancho de banda del espectro de potencia.
3. Coeficientes transformada wavelet y transformada de paquetes wavelet.
4. Coeficientes ceptrales.

Para el análisis del espectro de frecuencia es particularmente útil la transformada de Fourier y sus derivaciones como la Transformada de Fourier de Tiempo Corto (STFT) o Transformada de Fourier Ventaneada (Cortez, y otros, 2007) y la transformada de Gabor (Azor).

Por otro lado, la transformada wavelets supera en muchos aspectos la transformada de Fourier, especialmente en relación a la resolución de la frecuencia. Sin embargo, los recursos computacionales necesarios para realizar esta transformación son mucho mayores que los necesarios para la Transformada de Fourier. Para tiempo discreto existe la transformada discreta wavelet o DWT. Una forma generalizada de la DWT es la Transformada Paquete Wavelet (WPT) (Lat).

De Luca en (Delsys, 2009), recomienda la frecuencia media argumentando que es menos sensible al ruido, menos sensible al fenómeno de *aliasing* y en la mayoría de los casos es más sensible a los factores fisiológicos y biomecánicos que ocurren al interior de los músculos durante una contracción sostenida. Sin embargo, afirma que la estimación de la frecuencia media es más variable debido a la inestabilidad del espectro de la señal en las frecuencias bajas.

Por otro lado, una manera de realizar un análisis de frecuencia sin tener que usar una herramienta de transformación, es a partir de la medición de velocidad de las fibras musculares. Entre las características utilizadas para este fin se encuentran:

1. El valor de cruces por cero.
2. El número de inflexiones o cambios de signo.
3. σ .

Para ver una descripción detallada de cada característica, referirse a (Romo, y otros, 2007), (De Luca, 2006) y (Salazar, y otros, 2006). Del análisis de las señales ME, la idea es conformar un vector descriptor utilizando las características

anteriormente mencionadas o realizar algún tipo de manipulación sobre estos y así obtener las características descriptoras de los patrones EMGS.

Una importante distinción se presenta entre las contracciones musculares dinámicas y no-dinámicas. Varias clases de herramientas para procesamiento de señales han sido probadas basándose en el supuesto de estacionariedad. Esto significa que las características de la señal se asumen constantes durante el transcurso del tiempo. Específicamente cuando se aplica análisis espectral, se asume estacionariedad de la señal EMG.

2.2.5 Reducción de dimensionalidad

Después de obtener un vector descriptor el cual puede estar compuesto por todas o algunas de las características previamente mencionadas, se debe hacer que el módulo de control de alto nivel analice sus propiedades con el fin de reconocer alguno de los movimientos de interés (Reconocimiento de patrones). Debido a que la dimensión del vector descriptor en algunos casos es muy grande, es de interés eliminar información repetitiva y que no aporta para el reconocimiento de la intención de movimiento generada por el usuario. En esta etapa entra la reducción de dimensionalidad, la cual permite preservar la información más relevante, al tiempo que se reduce la dimensión de dicho vector. Entre las técnicas más utilizadas se encuentran:

1. El análisis de componente principal o transformada de Karhuemmn Loeve.
2. Descomposición de valores singulares.
3. Algoritmos genéticos.

2.2.6 Reconocimiento de patrones

En el reconocimiento de patrones se usan los resultados de la extracción de características como un conjunto de entrenamiento ya sea para una red neuronal o como una base de identificación estadística (Ferguson, y otros, 2002). En (Salazar, y otros, 2006) se considera la técnica de reconocimiento de patrones como una

disciplina de la inteligencia artificial, por lo tanto se consideran los siguientes elementos:

Adquisición y representación del conocimiento

Implica la transferencia y transformación de conocimientos de una o más fuentes como: libros, manuales, expertos, a un sistema informático. Para el dominio de los dispositivos prostéticos se interpreta como la adquisición de patrones de señales ME conocidas.

Aprendizaje

Implica cambios en el sistema el cual se adapta para permitir llevar a cabo la misma tarea a partir de las mismas condiciones. Busca calcular el patrón prototipo o base o el conjunto de patrones prototipo que caracterizan cada una de las señales ME a discriminar.

Clasificación

Consiste en proporcionar nuevos prototipos al sistema, independientes de los utilizados en el aprendizaje, para que éste los etiquete utilizando el conjunto de señales ME disponibles.

Existen una variedad de herramientas para el reconocimiento o clasificación de patrones de las cuales se pueden destacar:

Clasificador Bayesiano

Basado en el teorema de Bayes. Utiliza el concepto de probabilidad *a-posteriori*, probabilidad *a-priori* y probabilidad condicional.

Análisis discriminante lineal

Es otro método estadístico que discrimina la separabilidad de clases en forma lineal, lo cual no se ajusta de manera eficiente a los patrones EMGS por su dinámica estocástica. Sin embargo ha sido utilizado por Englehart y Hudgins en aplicaciones con señales ME (Englehart, y otros, 1999).

Redes neuronales artificiales

Esta es una de las herramientas más utilizadas, su aplicación puede reconocerse en (Alonso, y otros, 2002), (Hargrove, y otros, 2005) (Betancourt, y otros, 2004). Es un sistema computacional inspirado en las características de aprendizaje y estructura de la red neuronal biológica. Las redes ANN tienen características como: generalización y habilidad para aprender de la experiencia sin requerir un modelo matemático *a-priori* de las características de la señal, adaptación a los cambios de condiciones ambientales y habilidad para procesar datos degradados o incompletos, lo que las hace apropiadas para señales críticas como las señales EMGS

Lógica difusa

Son sistemas de inferencia, y al igual que las ANN simulan el comportamiento de decisión humana. Con posibilidad de introducir la experiencia de un experto en la elaboración de reglas de decisión, en un esquema *If-then-else*. Estos sistemas se pueden clasificar en dos categorías: la primera, incluye modelos lingüísticos basado en colección de reglas *If-then-else*, cuyos antecedentes y consecuentes usan valores *fuzzy* conocido como modelo *Mamdani* y la segunda categoría, está basada en sistemas tipo *Tagagi Sugeno* y usa una estructura de reglas que tienen antecedentes *fuzzy* y consecuentes funcionales.

Sistemas Neurofuzzy

La combinación de redes neuronales y lógica difusa permite una relación simbiótica en la cual se aprovecha el conocimiento de un experto y la capacidad de aprendizaje y eficiencia computacional de la red neuronal, logrando un sistema de decisiones más inteligentes.

2.2.7 Intenciones de movimiento

El sistema de reconocimiento de patrones tiene como finalidad reconocer las intenciones de movimiento del paciente. Para una prótesis de mano los tipos de agarres serían los movimientos a ejecutar. Sin embargo, de acuerdo con (Bing, y otros, 1991), (Gaviria, y otros, 2008), el agarre no es un movimiento único, existe un proceso el cual se divide en tres fases como:

La fase previa de agarre.

Ésta es la primera fase de la tarea de agarre y precede al agarre real. Cada opción de agarre representa un movimiento de los dedos de la mano como extensión, flexión, rotación, etc. La combinación de varios tipos de movimientos que deben realizar los dedos para asir un objeto define una planificación del agarre. Cuando la geometría del objeto se especifica, es necesario definir o calcular las trayectorias aproximadas de los dedos que contactan con el objeto.

La fase del agarre

Consiste en el agarre propiamente dicho del objeto, es decir, el conjunto de dedos que participan en el agarre y su posición y localización respecto al objeto a sujetar. Empieza en el momento que la mano toca y sostiene el objeto de una manera estable. En esta fase debe identificarse el tipo de agarre a ser empleado.

La fase de manipulación

La fase de manipulación se caracteriza por movimientos que resultan como consecuencia de mover el objeto con relación al ambiente. El agarre se selecciona en base a la movilidad y la destreza requerida para manipular el objeto.

En (Bing, y otros, 1991), se presentan las clasificaciones propuestas para los agarres de una mano humana, entre ellas están: Schlesinger, Napier y Cutovsky. Sin embargo, para los propósitos del presente proyecto se tendrá en cuenta la clasificación de Schlesinger debido a que las demás clasificaciones se concentran en diferenciar los tipos de agarre de acuerdo a la precisión y la fuerza, y no a la intención de movimiento.

Clasificación de Schelsinger

La clasificación de Schelsinger se propuso desde la perspectiva del diseño de una mano protésica, basado en los parámetros de la forma del objeto, superficie de la mano y forma de la mano. La Figura 2.3 ilustra los tipos de agarre: cilíndrico, pinza, gancho, palmar, esférico y lateral.

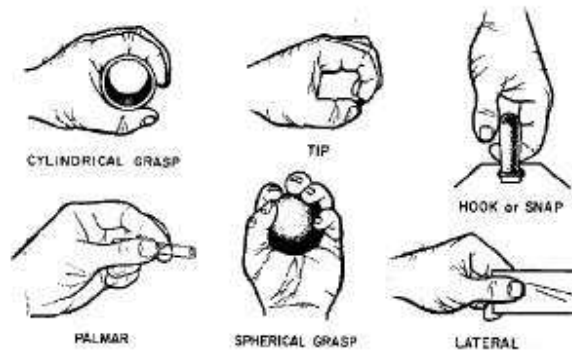


Figura 2.3. Clasificación de agarres de mano. Fuente (Bing, y otros, 1991).

Con el fin de seleccionar un conjunto estandarizado de intenciones de movimiento para una prótesis de mano controlada a partir de señales ME, es importante conocer los tipos de movimientos que pueden ejecutar las prótesis existentes. En conversación con el Ph.D. Rodríguez Cheu², hizo una precisión entre prótesis robóticas y prótesis clínicas, y enfatizando en que la diferencia radica en el número de grados de libertad del dispositivo: “Se consideran prótesis robóticas a los dispositivos prostéticos con más de dos grados de libertad y a las cuales se les puede aplicar los fundamentos robóticos como la planificación de trayectorias y su control, convirtiéndose en dispositivos adecuados para realizar más que un agarre, una manipulación del objeto. Por otro lado, las prótesis clínicas son mecanismos de tan solo uno o dos grados de libertad, las cuales por su limitación siguen siempre la misma trayectoria y ejecutando el mismo tipo agarre”. Para las prótesis clínicas de un grado de libertad las intenciones de movimiento a reconocer son generalmente: abrir, cerrar y detener, para prótesis con dos grados de libertad se adiciona el movimiento

² Profesor de la **Escuela Colombiana de Ingeniería** y reconocido investigador en el área de **biomecanismos**

de rotación. Entre las prótesis que tiene este tipo de movimientos se encuentran (Iberall, y otros, 2008), (Bionics), (Sánchez, y otros, 2006).

Un concepto muy adecuado para el diseño de una prótesis robótica es el Dedo Virtual (DV). En (Bing, y otros, 1991) y (Gaviria, y otros, 2008) se define como una representación abstracta usada para describir una colección de dedos o superficies de la mano aplicando fuerzas de oposición. Los dedos reales se agrupan en un dedo virtual (DV). Una clasificación de agarre muy útil para las prótesis clínicas es la considerada por Arbib (Bing, y otros, 1991), (Iberall, y otros, 2008), quien considera que la aprehensión involucra al menos dos fuerzas aplicadas en oposición contra las superficies de un objeto. Según este concepto se distinguen tres tipos básicos:

Oposición a las yemas

La fuerza se aplica a lo largo de un eje paralelo a la palma. Ofrece flexibilidad para manipulación a expensas de estabilidad y fuerza.

Oposición de palma

La fuerza se aplica a lo largo de una dirección perpendicular a la palma. Sacrifica flexibilidad en favor de estabilidad.

Oposición lateral

Las fuerzas se aplican a lo largo de una dirección transversal a la palma. Es un compromiso entre flexibilidad y estabilidad.

Para oposición a la palma, la palma es el primer dedo virtual (DV1), y los cuatro dedos son el segundo dedo virtual (DV2). DV1 termina en cualquier lugar sobre la superficie palmar, mientras que DV2 termina sobre la unión de la articulación de los dedos llamada metacarpofalange o MCP. Para oposición a las yemas, la oposición ocurre entre el pulgar (DV1) y uno o más dedos (DV2). Para oposición lateral, la oposición ocurre entre el pulgar (DV1) y el lado del dedo índice (DV2). Aquí el eje de oposición es ortogonal a DV1 y DV2. El concepto de dedo virtual es útil para diferenciar movimientos básicos como el de extensión y flexión en el que solo interviene un grado de libertad.

2.3 Sistemas de control para prótesis de mano

En (Zecca, 2003) se afirma que la principal limitación en el uso de una prótesis artificial es el control del sistema. Dicho control debe suficientemente intuitivo y que a su vez implique un escaso esfuerzo mental como sea posible. Lo anterior con el fin de hacer de la prótesis una herramienta práctica, que permita su uso por largos periodos de tiempo.

Esta área se relaciona estrechamente con el control de los motores de la prótesis de mano, a partir de la intención de movimiento del usuario y la información que proveniente del módulo de procesamiento sensorial previsto para cerrar el lazo de control.

Existen dos tipos de controles generalmente usados, estos son:

1. Control simple *On-Off*: en este caso el usuario envía todos los comandos como señales digitales TTL (0V o 5V) al sistema de control. Por ejemplo el comando abrir correspondería a un bit de una palabra de control.
2. Apertura y cierre proporcional: la entrada al sistema de control podrían ser señales analógicas en un rango de 0V a 5V o señales digitales de 16 bits. En este caso el mensaje no solo contiene información del tipo de movimiento si no también del grado de proporción a ejecutar.

Dependiendo de los sensores y actuadores implantados en la prótesis de mano, el sistema de control podría ser en lazo abierto o lazo cerrado. Una ilustración del sistema de control se presenta en la Figura 2.4.

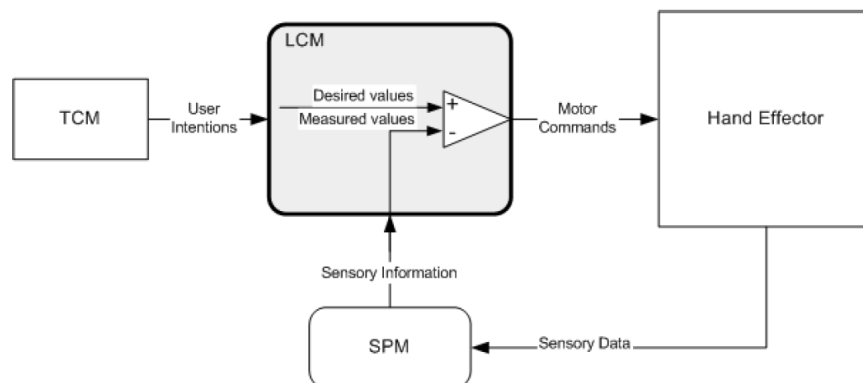


Figura 2.4 Sistema de control

Dado que los sistemas de control *On-Off* son sistemas muy sencillos e intuitivos, sin dejar de ser prácticos y eficientes, no se profundiza al respecto.

Por otro lado, en el ámbito de la robótica se prefieren los sistemas de control proporcional. Entre los más relevantes para el desarrollo de un robot se encuentran:

- Control PID.
- Control por par calculado.

Control PID

Es el control de mayor aplicación en los robots industriales actuales, el mecanismo es considerado como un sistema lineal y cada una de sus articulaciones es gobernada por un control descentralizado de tipo PID. Su ventaja es la facilidad de implementación y el bajo costo de cálculo matemático, pero presenta problemas de precisión de movimientos rápidos.

Control por par calculado

Dado que en la práctica se exigen evoluciones rápidas del robot con una gran precisión dinámica, es necesario utilizar un sistema de control más sofisticado que permita contar con la mayor parte de las fuerzas de interacción dinámica. La utilización del control por par calculado constituye una excelente solución teórica ya que este tipo de control asegura el desacople y la linealización de las ecuaciones del modelo las cuales son no lineales y acopladas.

2.4 Realimentación al usuario

En general, la mayoría de prótesis mioeléctricas no proveen ninguna retroalimentación directamente al usuario. Los usuarios deben controlar la operación de la mano con sus ojos.

El proceso básico parte de una electromiografía que mide la contracción y relajación del músculo en estudio y las traduce a señales auditivas o visuales de aumento o decremento dependiendo de la actividad electromiográfica.

En la literatura, existen diversos ejemplos de retroalimentación aplicada en dispositivos prostéticos, en las cuales la información que viene de los sensores de la prótesis son transmitidos al usuario a través de una estimulación en la piel, ya sea como estímulos mecánicos, eléctricos o térmicos. Sin embargo, estos sistemas no han sido exitosos debido a que la sensación que experimenta un paciente no es natural y el dispositivo resulta muy molesto para el paciente.

En el ámbito de la realimentación sensorial, aparece el término “mio-feedback” refiriéndose al proceso de detección de la actividad muscular voluntaria y entrega de dicha información al paciente a través de electro-estimulación; de acuerdo al grado de potencia de la señal eléctrica enviada al paciente se provoca una contracción muscular involuntaria.

Propiamente el mio-feedback no es un método de realimentación sensorial para dispositivos prostéticos, sino más bien una herramienta clínica para rehabilitación en pacientes con parálisis en sus extremidades, como se afirma en (Gomez, y otros, 2007). Sin embargo en trabajos como (Birkedal, y otros) y (Brittain, y otros, 1980) se hace uso de la electro-estimulación para realimentar al paciente con la información sensorial implantada en la prótesis.

Según el Ph.D. Rodríguez³, actualmente se trabaja en la conexión de un sistema de realimentación sensorial con los hilos nerviosos presentes en el miembro amputado e introduciendo en el sistema nervioso una corriente eléctrica proporcional a la información detectada por los sensores. Aún no se registran resultados acerca de este procedimiento, pero se espera que en los próximos años se encuentre información al respecto.

³ Director Grupo de Investigación en Sistemas Biomecánicos, adscrito a la Escuela Colombiana de Ingeniería

Capítulo 3

3 Conjunto normalizado de interfaces

3.1 Soporte teórico para la definición de interfaces

Para lograr que dos componentes hardware o software interoperen se hace necesario conocer la interfaz de cada uno de ellos. Es decir, qué tipo de información se requiere como insumo para la ejecución de las operaciones y qué tipo de información se espera como resultado. Para comprender la relación entre la información de entrada y salida de un componente se acompaña dicha especificación con una descripción detallada del funcionamiento y las posibles restricciones o modos de fallo del componente.

Actualmente, los sistemas electrónicos se conforman por componentes puramente hardware o una combinación entre componentes hardware y software como los sistemas basados en procesadores digitales en los cuales se ejecuta secuencialmente un conjunto de instrucciones.

Dado que el presente proyecto pretende definir un conjunto de interfaces para los componentes hardware y software de una prótesis robótica de mano, a continuación se expone la forma como regularmente se especifica una interfaz para dichos componentes

3.1.1 Interfaz Software

De acuerdo con (Ganssle, 2007) un módulo o componente software es un archivo de código fuente que contiene funciones o rutinas y a su vez las variables para soportar dichas funciones. Cada módulo contiene un número de funciones relativas. Por ejemplo, el módulo de conversión A/D podría incluir todos los controladores A/D en un simple archivo. El agrupamiento de funciones de esta manera, hace más fácil encontrar secciones relevantes de código y permite un encapsulamiento más efectivo.

El encapsulamiento busca entonces esconder los detalles de la operación de una función y procura mantener las variables usadas por la función de una manera local. Aunque lenguajes como C y ensamblador no soportan el encapsulamiento de una manera explícita, es posible obtener los beneficios del encapsulamiento al igual que en la programación orientada a objetos, simplemente realizando una codificación cuidadosa en el momento de asignar el alcance de las variables de cada función. Por ejemplo, si la responsabilidad de una función es intercambiar la información entre dos espacios de memoria, la variable necesaria para almacenar temporalmente dicha información debe ser declarada de manera local; dándole a esta variable un carácter de privacidad ya que no es de interés para la ejecución de otros componentes.

Con los anteriores conceptos claros y de acuerdo con (Informática, 2009), la interfaz de un componente software es la parte de un programa que sirve para comunicarse con otro programa, con una parte del hardware o con el usuario. En un sentido más amplio, puede verse la interfaz como el conjunto de comandos y/o métodos que permiten la intercomunicación del programa con cualquier otro programa o entre partes del propio programa, o incluso con componentes hardware.

Así entendido, la interfaz de un componente describe todas las responsabilidades de las funciones que están incluidas en él. Los lenguajes de programación basados en la descripción de código, es decir los lenguajes “no gráficos”, en su mayoría coinciden en la definición de una interfaz de una función software en la cual se especifica el nombre y el tipo de los argumentos de entrada y salida de ella:

TipoRetorno ***nombreFuncion***(***tipo*** Argumento1, ***tipo*** Argumento2, ... ***tipo*** ArgumentoN);

En el *Firmware Development Standard*, en español, estándar de desarrollo firmware propuesto en (Ganssle, 2004) la documentación para las interfaces de los componentes de software debe contener los siguientes aspectos: el nombre de la interfaz, los valores de entrada y salida del componente, y una descripción detallada de su funcionalidad, restricciones y modos de fallo. Del mismo modo sugiere que dicha información debe registrarse en el mismo código del programa a manera de comentario y previamente a la definición del prototipo de la función o componente.

Para la definición de interfaces existen diferentes lenguajes tales como: MIDL, del Inglés Lenguaje de Definición de Interfaces de Microsoft, *Open Service Interface Definition*, WSDL, en español lenguaje de descripción de servicios web, *Platform Independent Component Moledling Language* y el más importante de ellos y del cual se han derivado en su mayoría es el Lenguaje de Descripción de Interfaces o IDL por su nombre en Inglés.

El IDL es un lenguaje desarrollado por el *Object Management Group*, OMG de sus siglas en inglés, con el fin de contar con un mecanismo de abstracción para la implementación de sistemas distribuidos basados en CORBA. La principal función es la de separar la implementación de la definición a través de un contrato entre el cliente y el servidor en el que se describen los datos y las interfaces de los objetos de una aplicación. IDL es independiente del lenguaje de programación, siendo esta una característica muy importante a tener en cuenta para la definición del conjunto de interfaces que se pretende generar en el presente proyecto.

3.1.2 Interfaz Hardware

De acuerdo con (Informática, 2009), una interfaz hardware es el punto de contacto entre dos sistemas o módulos de un sistema. Es una conexión física y funcional entre dos aparatos o sistemas independientes. Particularmente en el campo de la electrónica y las telecomunicaciones, una interfaz es el puerto o circuito físico a través del que se envían o reciben señales desde un sistema o subsistemas hacia otros. No existe una interfaz universal, existen diferentes estándares tan complejas como: Interfaz USB o interfaz SCSI o mucho más sencillas como las interfaces que definen los pines de entradas y salidas de un dispositivo electrónico.

En general, en las interfaces se establecen las especificaciones técnicas concretas para que un componente se integre con otro y desempeñe la función para la cual fue desarrollado.

La interconexión sólo es posible utilizando el mismo tipo de datos y un protocolo común entre los componentes a conectar. Es decir, si la entrada requerida en un componente es de tipo analógica, no es posible conectarle una salida digital de otro componente. El tipo de información no sería común entre ambos ocasionando una falla en la interconexión.

Así también, una interfaz puede ser definida como un intérprete de condiciones externas al sistema, por ejemplo a través de transductores y otros dispositivos, permitiendo una comunicación con actores externos, como personas u otros sistemas. Para el caso de una prótesis mioeléctrica, caso de estudio del presente proyecto, la interfaz entre el hombre y la mano protésica son simplemente los sensores EMG.

No existe una manera estandarizada de definir una interfaz hardware. Sin embargo, entre los lenguajes más difundidos para la descripción de componentes hardware se encuentran *VHDL* y *Verilog*. Dichos lenguajes definen la interfaz de un componente hardware a través del concepto de *entity* en el caso de *VHDL* y *module* en *Verilog*. Dicha entidad o módulo se asemejan a la definición de la interfaz o prototipo de una función de software.

Otra forma de comprender la entidad es verla como una caja negra en la que simplemente se especifican las entradas y salidas y no se describe su funcionamiento interno. Por ejemplo, a continuación se especifica la entidad de un multiplexor hardware

```

Entity mux
  PORT
  (
    control      :IN      bit;
    entrada1     :IN      bit;
    entrada2     :IN      bit;
    salida       :OUT     bit
  );
End mux;
```


La entidad *mux* anteriormente descrita especifica el nombre de cada señal, si es de entrada o salida y el tipo de dato al que pertenece dicha información. Si no se conoce previamente el funcionamiento de un multiplexor no es posible deducir de la definición de la entidad cual será su comportamiento.

Frecuentemente, desarrolladores de hardware se encuentran con documentos que describen un componente hardware a través de su interfaz, este tipo de documentos se conoce como Hoja Técnica o en Inglés Datasheet, el cual presenta una descripción detallada del comportamiento de un dispositivo hardware, sus limitaciones, modos de fallo, condiciones eléctricas entre otra información.

3.1.3 Estándar para la definición y documentación de Interfaces

De acuerdo a lo expuesto anteriormente, tanto una interfaz hardware como software tienen el mismo objetivo; definir las responsabilidades de un componente sin especificar la manera en que dichas responsabilidades se llevan a cabo.

Al definir un conjunto de interfaces se realiza un aporte a la especificación de la arquitectura del sistema más que a la manera de realizar su implementación. Por ejemplo, en el inicio de la etapa de diseño de un sistema hardware/software, no es relevante la estrategia a seguir para cumplir con determinada funcionalidad o la herramienta con que se realizará dicha implementación. Lo que realmente importa es que se defina esa función y se especifique el tipo de información que es requerida como entrada al componente y la información que éste entrega luego de su ejecución. Sin embargo, es de gran interés para los desarrolladores de sistemas hardware/software que además de la información de entrada y salida del componente se especifique el tiempo de respuesta de una función, las condiciones eléctricas para las señales hardware y el protocolo de comunicación.

Teniendo en cuenta lo anterior y considerando la necesidad de definir un conjunto de interfaces hardware/software documentadas e independientes de la plataforma de implementación se decide utilizar el lenguaje de definición de interfaces, IDL, para especificar tanto interfaces hardware como software y realizar su documentación por medio *Firmware Development Standard* propuesto por (Ganssle, 2004). Sin embargo, dado que esta solución por sí misma no cumple con los requerimientos para la definición y documentación de componentes hardware, en el presente documento se proponen ciertas modificaciones, principalmente al estándar de documentación más que al lenguaje de definición.

A continuación se describe la manera como este proyecto propone realizar la definición y documentación de las interfaces hardware y software haciendo uso de lenguaje de definición de interfaces y el estándar de desarrollo firmware propuesto por (Ganssle, 2004). Para ello, en el Anexo 3 se presenta de una manera resumida las principales características del IDL, posteriormente se expone una plantilla para la documentación de interfaces software.

Ahora bien, evidenciando que el aspecto más importante en la definición de una interfaz es la especificación de la función, entonces es a ella a quien se le hará una descripción detallada. En la Tabla 3.1 se presenta una plantilla para la documentación de dichas funciones.

Inicialmente, la plantilla se propuso para la documentación de interfaces de componentes software y no de componentes hardware. Por lo tanto, se completará la documentación agregando la descripción de las características propias de una interfaz hardware, tales como, el tipo de señal analógica o digital y el protocolo de comunicación de las señales de entrada o salida del componente. De esta manera será posible unificar la definición y documentación para interfaz tanto hardware como software.

Si en una función software los valores de entrada y salida se conocen como argumentos de la función, para el caso hardware estos valores se conocen como señales del sistema. Tanto los argumentos como las señales son equivalente en el ámbito del IDL a los parámetros de la operación.

Dado que se pretende definir las interfaces hardware y software de un sistema electrónico haciendo uso del Lenguaje de Definición de Interfaces, IDL, y teniendo en cuenta que dicho lenguaje fue desarrollado especialmente para describir interfaces de componentes software, en el presente proyecto se propone definir un nuevo tipo de dato el cual describa de una mejor manera a una señal analógica y a su vez permita modelar una interfaz de un componente hardware. El tipo de dato propuesto se denominará *Analog*, el cual, al igual que cualquier otro tipo de dato, por ejemplo un *Integer*, tiene sus propias características que serán descritas a continuación.

Un tipo de dato *Analog* corresponde a todo tipo de señal eléctrica que varía de manera continua e infinitesimalmente ya sea en voltaje o en corriente. Por ejemplo, una onda de voltaje senoidal continua es una señal analógica de una sola frecuencia y podría ser considerada como un dato de tipo Analog.

Un ítem importante para adicionar a la documentación de una interfaz de un componente hardware o software es el máximo tiempo de respuesta de una función.

Su importancia radica en que muchos de los sistemas hardware/software deben responder en tiempo real, es decir, responder antes de un tiempo especificado.

Por ejemplo, si en la interfaz del componente “Caracterizador” se define la responsabilidad de “caracterizar señal ME”, en el momento de su implementación pueden existir diversas estrategias que cumplan con la misma funcionalidad. Sin embargo, algunas de ellas pueden ser muy complejas aunque probablemente muy precisas pero demandan mucho tiempo de ejecución. Si una de estas estrategias excede el tiempo máximo de ejecución haría que todo sistema se ralentice y la respuesta no se genere en el tiempo adecuado.

Módulo:	Nombre del módulo al que pertenece la interfaz.
Interfaz:	Nombre de la interfaz a la que pertenece la responsabilidad.
Responsabilidad	Nombre con que se define la operación.
Retornos:	Descripción del valor de retorno o salida de la operación.
Argumento 1:	Descripción del primer argumento o señal.
Argumento 2:	Descripción del segundo argumento o señal.
Argumento n:	Descripción del n-ésimo argumento o señal.
Tiempo Respuesta:	Tiempo máximo de respuesta de la operación
Creador:	Nombre del autor, fabricante, referencia comercial.
Fecha de creación:	Fecha en la que se desarrolla o fabrica la operación.
Descripción:	Descripción de las responsabilidades de la operación.
Nota:	Se especifican las restricciones de ejecución y los posibles modos de fallo.

Tabla 3.1 Plantilla para documentación de interfaces

3.1.4 Implementación de Interfaz hardware/software

Un método para intercomunicar dos módulos hardware de forma paralela es asignar un pin digital de salida por cada parámetro de información a transmitir. Es decir, para el conjunto de formas de agarre, al agarre Palmar se le asignaría un pin digital o bit de transmisión. Si el pin se encuentra activo quiere decir que la intención de movimiento es un agarre de tipo Palmar; por lo tanto, por cada intención de movimiento se debe contar con un pin de transmisión.

Otra forma de implementar la comunicación paralela es asignar un pin de salida por cada bit de una palabra digital, es decir, si se desea transmitir un byte, palabra digital de 8bits, se asignarían 8 pines; de esta manera se posibilitan 256 órdenes o intenciones de movimiento diferentes.

Para la comunicación serial, tan solo es necesario contar con dos o tres pines para transmitir toda la información, dado que los bits viajan uno tras otro por la misma línea de comunicación sincronizados con una línea de reloj. En algunos casos se utilizan dos líneas independientes para la transmisión y recepción de datos.

Para el intercambio de información entre dos circuitos integrados se sugiere el uso de los siguientes buses comunicación:

- **I²C**. Su nombre viene de Ingles, *Inter-Integrated Circuit* o integrador de Circuitos Integrados.
- **SPI**. Su nombre viene del Ingles Serial Peripheral Interface o interfaz periférica serial.
- **UART**. son las siglas de Universal Asynchronous Receiver-Transmitter. En Español, Transmisor-Receptor Asíncrono Universal
- **USB**. Viene de *Universal Serial Bus* o en español, bus universal de comunicación serial.

Si la interfaz es software, simplemente se transmite la palabra digital representada en un dato de tipo *byte*; concluyendo que, sin importar el tipo de interfaz, hardware o software, la información a transmitir es la misma. Por lo tanto, se define un protocolo de comunicación de acuerdo a la información a transmitir, en este caso las intenciones de movimiento y el tipo interfaz a utilizar. Si la interfaz es hardware se debe especificar detalladamente las características del bus de comunicación, tales como, frecuencia de sincronización, voltajes de las líneas de comunicación, entre otras características propias de cada bus.

3.2 Modelo de análisis del conjunto de interfaces Hardware/Software para simulador de prótesis de mano robótica

El modelo de análisis identifica las clases Entidad, Interfaz y Control, así como los actores que interactúan en el sistema. En el Anexo 5 se amplía la información relacionada con las clases que conforman dicho modelo. Específicamente para el presente proyecto, el modelo de análisis tiene como finalidad acercar la información

conceptual del dominio a los sistemas hardware/software independientemente de las herramientas de implementación.

Para la realización del modelo de análisis del sistema: “Simulador Virtual de Prótesis Robótica de Mano” se usó el lenguaje de modelado unificado, UML por sus siglas en inglés, y específicamente de los estereotipos para las clases Interfaz, Control y Entidad. A continuación se presentará una descripción de dicho modelo con base en el trabajo expuesto en el Capítulo 2, Base Inicial de Conocimiento, y de una variedad de sugerencias de personas expertas en el área.

Es presuntuoso pensar que en dicho modelo se encuentran todas las posibles estrategias de soluciones. Sin embargo, si es posible afirmar que es un modelo muy completo que se adecúa a una gran cantidad de soluciones.

Debido a que el modelo general de análisis abarca una descripción completa del sistema, su grado de complejidad y tamaño son significativos. Por esta razón se decidió fragmentar en sub-modelos limitados por las clases interfaz. Con este criterio se identificaron los siguientes siete módulos:

1. Módulo Gestor Analógico (MGA).
2. Módulo Gestor Digital (MGD).
3. Módulo Caracterizador de Señal (MCS).
4. Módulo Gestor de Reconocimiento de Intención de Movimiento (MGRIM)
5. Módulo Controlador del Dispositivo (MCD)
6. Módulo Gestor Sensorial (MGS)
7. Módulo Gestor de Realimentación (MGR)

A partir de los anteriores módulos se especifica el conjunto de interfaces que interconectan dichos módulos. En la Figura 3.1 se presenta un diagrama general del conjunto de interfaces.

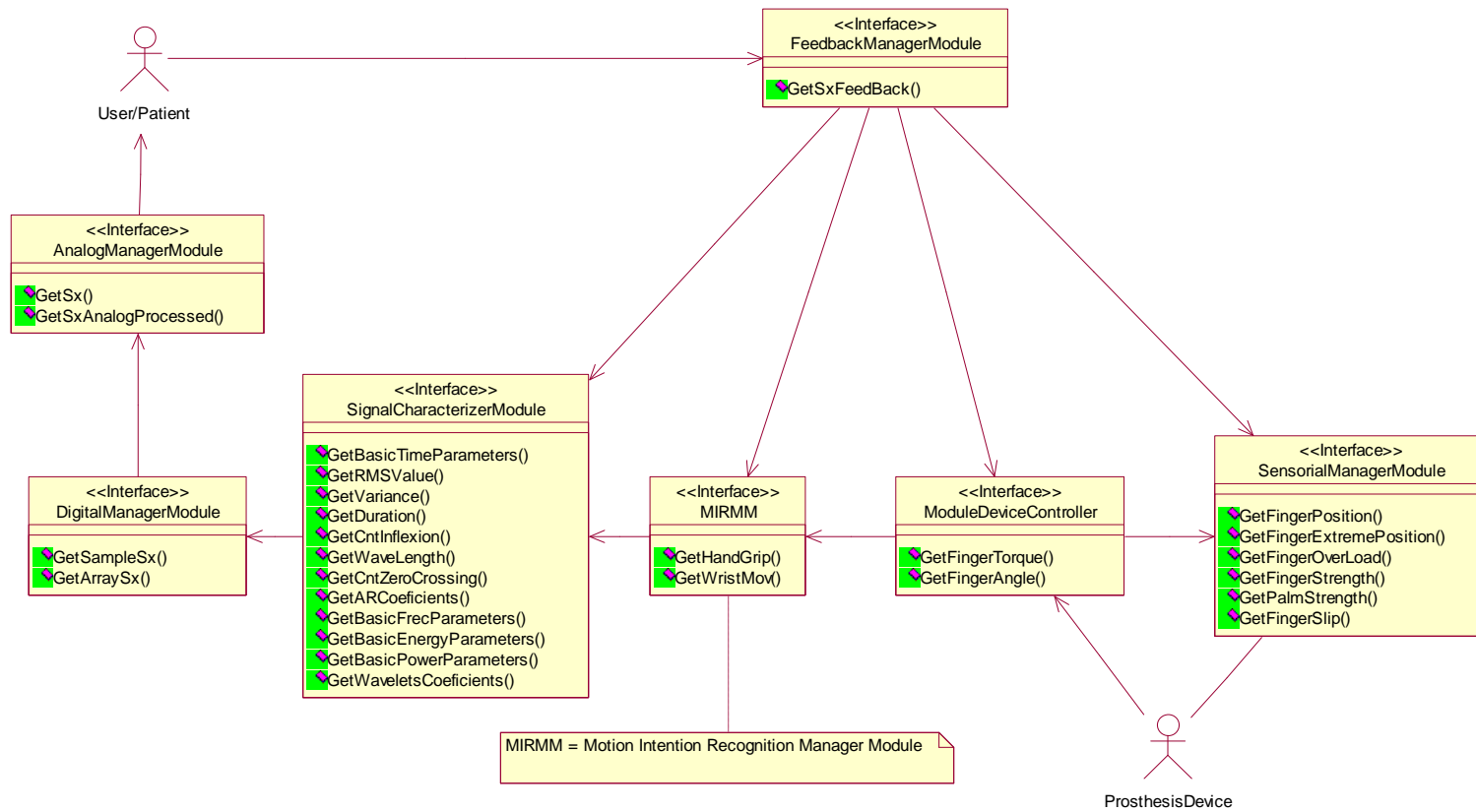


Figura 3.1 Diagrama UML del conjunto normalizado de interfaces para los componentes de un simulador virtual de prótesis de mano robótica

3.2.1 Modulo Gestor Analógico (MGA)

En este módulo se describe la etapa de transformación, amplificación y filtraje analógico de la señal mioeléctrica. A este módulo en (Delsys, 2009) se lo denomina como sensor activo dado que contiene los electrodos y la circuitería electrónica para realizar un pre-procesamiento de la señal ME.

A partir del modelo de la Figura 3.2, de izquierda a derecha inicialmente se encuentra la interfaz de sensado de la fuente de señal o *Interface_SignalSource_HW*. Dicha interfaz corresponde a los dispositivos que *transforman* la señal eléctrica proveniente de la actividad muscular a otra señal eléctrica interpretable por dispositivos electrónicos convencionales. Tales dispositivos filtran y amplifican la señal, para luego ser presentada a un siguiente módulo a través de la interfaz Gestor Analógico o *Interface_AnalogManager_HW*.

La interfaz de sensado de la fuente de señal le corresponde al conjunto de dispositivos que transforman las señales musculares eléctricas a señales interpretables por dispositivos electrónicos convencionales. Específicamente, en esta interfaz se presenta el resultado del sensado de la señal mioeléctrica tal y como se adquiere directamente de los músculos, es decir, con bajos niveles de voltaje, entre 0 y 10 mVpp y altos niveles de ruido especialmente provenientes de componentes electrónicos adyacentes e incluso del propio sistema de adquisición de señal ME. El ruido presente en el proceso tiene una frecuencia que va desde cero hasta varios cientos de Hertz.

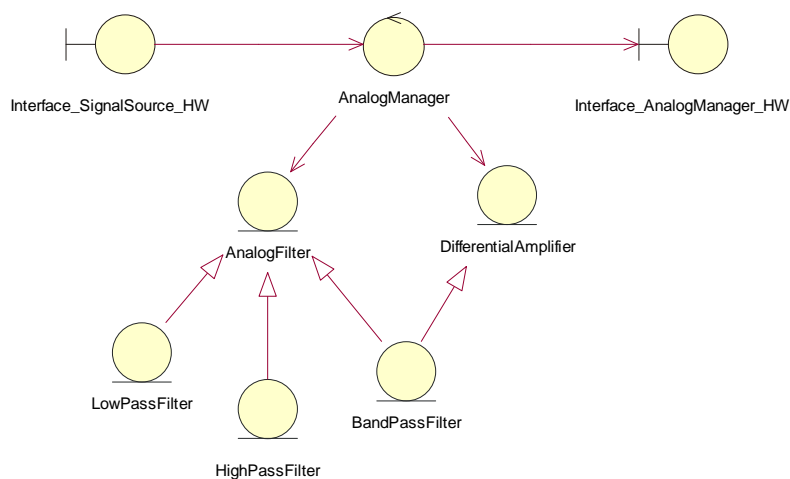


Figura 3.2 Modelo de análisis en UML del Modulo Gestor Analógico.

La interfaz *Interface_SignalSource_HW* es implementado por dispositivos tales como los electrodos de superficie, de inserción o coaxiales.

Electrodos superficiales: son pequeños conos o discos metálicos fabricados de plata o acero inoxidable que se ponen en contacto directamente con la piel. Para reducir la resistencia del contacto se utiliza pasta conductora.

Electrodos de inserción: consiste de una aguja metálica corriente en cuya longitud total, excepto en la punta, ha sido aislada. La variación del potencial se mide entre el extremo de la punta ubicada en el músculo y un electrodo superficial de referencia ubicado en la piel. A esta clase de electrodos se les conoce como *monopolar*.

Electrodos coaxiales: los cuales consisten de una aguja en cuyo interior se han insertado uno o varios conductores metálicos finos aislados entre sí. La diferencia de potencial se mide entre los conductores.

De lo expuesto anteriormente se deduce que la principal responsabilidad de la interfaz *Interface_SignalSource_HW*, es la de obtener una señal eléctrica proporcional a la actividad muscular. Visto de una manera sistemática se trata de una función que retorna una variable analógica que toma valores de acuerdo con la actividad muscular. En el presente proyecto se ha denominado a esta función como, “obtener señal mioeléctrica” o de manera compacta, *GetSxME()*. Dicha función retorna una señal mioeléctrica de tipo análoga denominada *Sx_ME*.

Por otra parte, la interfaz Gestor Analógica o *Interface_AnalogManager_HW*, tiene como objetivo registrar las responsabilidades del proceso de filtrado y amplificación analógica que se realiza sobre la señal mioeléctrica, *Sx_ME*. Por un lado su finalidad es la de amplificar los diminutos potenciales recogidos en el músculo y por otro, la de eliminar toda señal distinta a la proveniente de la actividad muscular.

A continuación se define el módulo gestor analógico haciendo uso de IDL y posteriormente, con base al estándar de desarrollo firmware especificado en (Ganssle, 2004), en las Tabla 3.2 y Tabla 3.3 se documentan las responsabilidades identificadas para el Módulo Gestor Analógico (MGA)


```

Module Analog Manager
{
  Interface SignalSource_HW
  {
    out Analog SxME GetSx( void );
  }
  Interface AnalogManager
  {
    GetSxAnalogProcessed( in Analog SxME, out Analog SxMEAnalogProcessed);
  }
};
    
```

Módulo:	Analog Manager
Interfaz:	SignalSource_HW
Responsabilidad:	out Analog GetSx(void)
Retorno:	Argumento de salida de la señal a partir de la fuente.
Arg1:	No se sugiere ningún argumento de entrada.
Tiempo de Respuesta:	Del orden de los nanosegundos. Dado que se trata de una circuitería analógica no proporciona latencias importantes para el desempeño de todo el sistema.
Descripción:	Recoge la actividad eléctrica del músculo ya sea por electrodos superficiales o de inserción.
Nota:	<ol style="list-style-type: none"> 1. Se requiere de un potencial eléctrico de referencia para la señal mioeléctrica. Se recomienda usar las articulaciones del esqueleto para tal fin. 2. Si no se cuenta con un sensor activo de señales ME, Se recomienda el uso electrodos desechables de electrocardiografía superficial.

Tabla 3.2 MGA, Sensado de Fuente de Señal, Obtener Señal.

En desarrollos como (Alonso, y otros, 2002) se separa la circuitería de amplificación de la de filtrado, sin embargo, el tipo de configuración recomendada en (Delsys, 2009) para la amplificación de señales ME es el arreglo diferencial, es decir, a partir de establecer dos puntos del músculo y adquirir de cada uno la señal ME, se restan estas señales y su resultado posteriormente es amplificado. Incluso si el rechazo del modo común es mayor a los 80dB este arreglo a su vez actúa como un filtro pasa banda para la señal vista desde la superficie de detección. Sin embargo, en (Konrad, 2005) se recomienda no usar ningún tipo de filtro hardware, excepto el amplificador pasabandas de 10 a 500Hz, el cual es necesario para evitar los efectos de *aliasing* en el proceso de muestreo de la señal.

Las responsabilidades identificadas para este módulo son claramente las de filtrar y amplificar la señal ME. Sin embargo, no es de interés para el resto del sistema tener disponibilidad de dichas funciones de manera independiente. Por lo tanto, se decide unificar estas funciones en una sola, denominándola “obtener señal procesada análogamente” o de manera compacta, *GetSxAnalogProcessed()*.

A continuación se define a través del uso del IDL las interfaces que conforman el Módulo Gestor Analógico y se especifican sus responsabilidades.

Módulo:	Analog Manager
Interfaz:	<i>AnalogManager</i>
Responsabilidad:	<i>GetSxAnalogProcessed(in Analog SxME, out Analog SxMEAnalogProcessed)</i>
Arg 1. SxME	Señal mioeléctrica de entrada.
Arg2. Sx_ME_Filt_Amp	Señal mioeléctrica filtrada y amplificada.
Tiempo de Respuesta:	Del orden de los nanosegundos. Dado que se trata de una circuitería analógica no proporciona latencias importantes para el desempeño de todo el sistema.
Descripción:	Las recomendaciones dadas en (De Luca, 2006) y (Seniam, 2007), para el pre-procesamiento de una señal ME se sugiere: <ul style="list-style-type: none"> • Amplificación de la señal ME de 1000 veces. • Filtrado con ancho de banda de 20 – 500Hz • Tasa de rechazo de modo común CMRR > 80dB • Impedancia de entrada > 100 Mega Ohms.
Nota:	En esta interfaz no se debería realizar ningún proceso diferente al de amplificar y filtrar la señal analógicamente. Por ejemplo, no se debería rectificar la señal ME. Para este proceso se destina un módulo especial que se describe más adelante. Debido a que la señal EMG es de baja amplitud respecto a otras señales del ambiente sobre la superficie de la piel es necesario y conveniente detectarla con una configuración diferencial.

Tabla 3.3 MGA, Gestor Analógico, Obtener señal pre-procesada analógicamente

3.2.2 Módulo Gestor Digital (MGD)

Este módulo digitaliza la señal mioeléctrica con el fin de utilizar las ventajas del procesamiento digital de señales, particularmente, para obtener una disminución del nivel de ruido a través de la ejecución de filtros. En la figura 3.3 se presenta el modelo de análisis del módulo de digitalización.

Entre los algoritmos de filtrado digital utilizados se encuentran los filtros *Notch* o ranura, filtros FIR del inglés *Finite Impulse Response*, filtros IIR del inglés *Infinite Impulse Response* y las dos variaciones de este tipo: AR Autoregresivo y ARMA

Autoregresivo de promedio móvil y por último, los filtros adaptativos los cuales pueden cambiar su comportamiento, es decir, se ajustan los coeficientes de acuerdo con el algoritmo de adaptación LMS del inglés, *Least-Mean-Square algorithm* o RMS *Recursive-Least-Squares algorithm*.

Con cualquiera de los anteriores algoritmos, excepto con el *Notch*, se implementan los siguientes tipos: filtros pasa bajas frecuencias, pasa altas frecuencias y las respectivas combinaciones entre ellos. Particularmente el filtro *Notch* o de ranura es utilizado para eliminar una sola frecuencia, es un filtro altamente selectivo el cual también puede ser implementado con un algoritmo de adaptación.

Para implementar un proceso de digitalización de señal se deben considerar principalmente los parámetros de: frecuencia de muestreo y resolución de conversión. Particularmente para el caso de la señal ME, la frecuencia de muestreo se calcula a partir del criterio de *Nyquist*, el cual sugiere que el muestreo de la señal debe ser mayor a doble de la frecuencia más alta contenida en la señal. Considerando que la mayor densidad espectral de potencia de la señal ME se encuentra debajo de los 500Hz, la frecuencia de muestreo debe ser de al menos de 1000 muestras por segundo, es decir, una frecuencia de 1KHz.

El Timer que aparece en la Figura 3.3 genera la frecuencia de muestreo, quien a su vez dispara el proceso de digitalización y filtrado de la señal ME. Es decir, cada que el Timer genere un flanco, ascendente o descendente pero no ambos a la vez, se adquiere una muestra de la señal ME del Módulo Gestor Analógico. Posteriormente se ejecuta un algoritmo basado en la convolución, el cual genera un dato correspondiente a una muestra de la señal filtrada.

Modelo de análisis en UML del Módulo Gestor Digital (MGD)

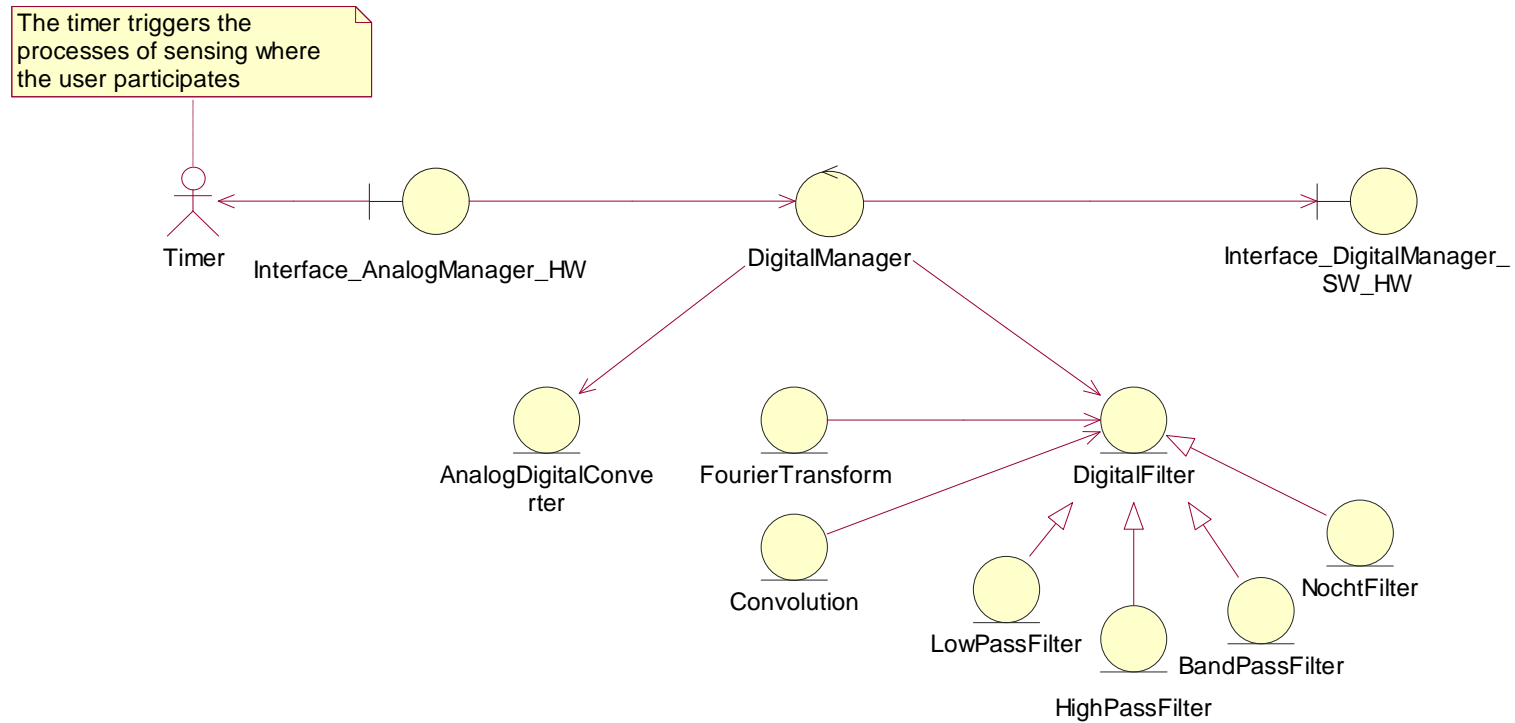


Figura 3.3. Modelo de análisis del módulo de digitalización y filtraje digital en UML

Por otro lado, a diferencia de la convolución los algoritmos basados en el uso de las transformadas en frecuencia requieren como insumo un conjunto o arreglo de muestras para su procesamiento como se explicará más adelante.

En el caso de la señal ME, la cantidad de muestras recomendadas en (Romo, y otros, 2007) es de 256 muestras a una frecuencia de muestreo 1KHz; lo que equivale a tener almacenado 256ms de información de la señal. A diferencia del modulo analógico, el cual se procesa muy rápidamente, un procesamiento digital que tarde 256 ms es relativamente lento. En (Todd R. Farrell, 2007) demuestran que los sistemas prostéticos deben responder a una intención de movimiento por parte del usuario en un tiempo no mayor a 200ms para que se consideren dispositivos funcionales.

Respecto a la resolución del conversor análogo digital las posibilidades pueden ser muy amplias. Sin embargo, comercialmente se encuentran de una manera más frecuente conversores con resoluciones de 8, 10, 12, 14 y 16 bits. Para elegir una resolución adecuada se debe tener en cuenta las características de la señal ME posterior al módulo de amplificación

La señal ME capturada a través de electrodos de plata o acero inoxidable varía entre 0 y 6mVp. Si se considera un factor de amplificación de 500 veces, la señal ME variaría entre 0 y 3Vp. Con un conversor de 10 bits, es decir de 1024 posibles niveles, se tendría una digitalización de 5.8 mV / bit. Considerando que la detección de detalles en bajas amplitudes está limitada por el nivel de ruido del sistema amplificador y que la mayoría de amplificadores tiene un nivel de ruido del orden de unos pocos mili-voltios; la resolución proporcionada por un conversor de 10 bits generará pérdida de información de la señal ME.

Por lo tanto, para evitar pérdida de información y disminuir los efectos de cuantización se recomienda usar dispositivos con una resolución de conversión mayor a 10 bits y una frecuencia de muestreo de al menos 1KHz. Una alternativa interesante son los conversores de señal de audio, tales como los de la familia SI3000 los cuales cuentan con una resolución de conversión de 16 bits, una frecuencia de muestreo entre 4KHz y 12KHz, convirtiéndolos en una excelente alternativa para el proceso de digitalización de la señal ME.

A continuación se describe, a través del uso del IDL, el Módulo de Filtrado digital. Se especifica las responsabilidades que se deben implementar para cada interfaz.

Module Digital Manager

```
{
  Interface Digital Manager SW HW
  {
    bool GetSampleSx( out integer SampleSxME );
    bool GetArraySx( out integer ArraySxME[], out integer SizeArray );
  }
};
```

Módulo:	<i>Digital Manager</i>
Interfaz:	<i>DigitalSx_filter</i>
Responsabilidad:	bool GetSampleSx(out integer SampleSxME)
Arg 1.	<i>GetSampleSxME</i> Valor entero correspondiente a la muestra digital de la señal ME.
Retornos: booleano.	Retorna verdadero si ya se ha adquirido una nueva muestra o falso aún se encuentra en proceso adquisición. Esta señal puede ser representada por una línea de salida digital hardware denominada <i>DataReady</i> .
Tiempo de Respuesta:	Máximo 1 ms. Dado que la frecuencia de muestreo mínima es de 1 KHz.
Descripción:	Realiza un proceso de digitaliza y filtrado de la señal ME e informa a través del “Data Ready” que el proceso de adquisición se ha llevado a cabo.
Nota:	Se recomienda utilizar un módulo de conversión análogo digital de resolución mayor a 10 bits y una frecuencia de muestreo de al menos 1 KHz.

Tabla 3.4 MGD, Gestor Digital SW HW, Filtro digital. Obtener muestra de señal

Módulo:	<i>Digital Manager</i>
Interfaz:	<i>GetArraySx</i>
Responsabilidad:	bool GetArraySx(out integer ArraySxME[], out integer SizeArray)
Arg 1.	<i>ArraySxME</i> Arreglo de valores enteros correspondiente al arreglo de muestras digitalizadas de la señal mioeléctrica.
Arg 2.	<i>SizeArray</i> Tamaño del arreglo de muestras de la señal ME.
Retornos: booleano.	Retorna verdadero si ya se ha adquirido un conjunto de muestras o falso si aún se encuentra en proceso adquisición. Esta señal puede ser representada por una línea de salida digital hardware denominada <i>DataReadyArray</i> .
Tiempo de Respuesta:	Máximo 256 ms. Considerando un arreglo de 256 muestras digitalizadas a una frecuencia de muestreo de 1 KHz.
Descripción:	Realiza un proceso de digitalización y filtrado de la señal ME e informa a través del “DataReadyArray” que el proceso de adquisición se ha llevado a cabo.
Nota:	Se recomienda utilizar un conversor de 16 bits y una frecuencia de muestreo mayor a 1 KHz.

Tabla 3.5. MGD, Gestor Digital SW HW, Obtener arreglo de muestras de la señal.

3.2.3 Módulo Caracterizador de Señal (MCS)

Este módulo tiene como propósito extraer características ya sean temporales o espectrales de la señal ME digitalizada con miras de reconocer y diferenciar una señal de otra.

Características temporales

En el dominio temporal, la extracción de características es basada en cálculos estadísticos los cuales a diferencia del las características espectrales se ejecutan de manera rápida. Para la extracción de características espectrales se requiere de un proceso de transformación de la señal de tiempo a frecuencia, lo cual evidentemente conlleva un mayor esfuerzo computacional.

Características espectrales

Para lograr una descripción más completa del fenómeno físico es necesario extraer información del espectro de frecuencia de la señal. Para tal fin, se realiza un proceso de transformación de la señal del dominio del tiempo al dominio de la frecuencia. Es decir, a partir del conjunto o arreglo de muestras obtenido en el proceso de digitalización en el dominio del tiempo se ejecuta un algoritmo de transformación el cual genera un arreglo de muestras con la información en el dominio de la frecuencia.

Entre las herramientas de transformación de mayor interés se encuentran la Transformada de Fourier y la Transformada Wavelet en su versión discreta. De acuerdo a la herramienta elegida para generar el arreglo de muestras en el dominio de la frecuencia es posible extraer una serie de características de la señal.

De acuerdo a lo expuesto anteriormente se propone el siguiente modelo de análisis en UML del Módulo Caracterizador de Señal MCS.

Modelo de análisis en UML del Módulo Caracterizador de Señal (MDS)

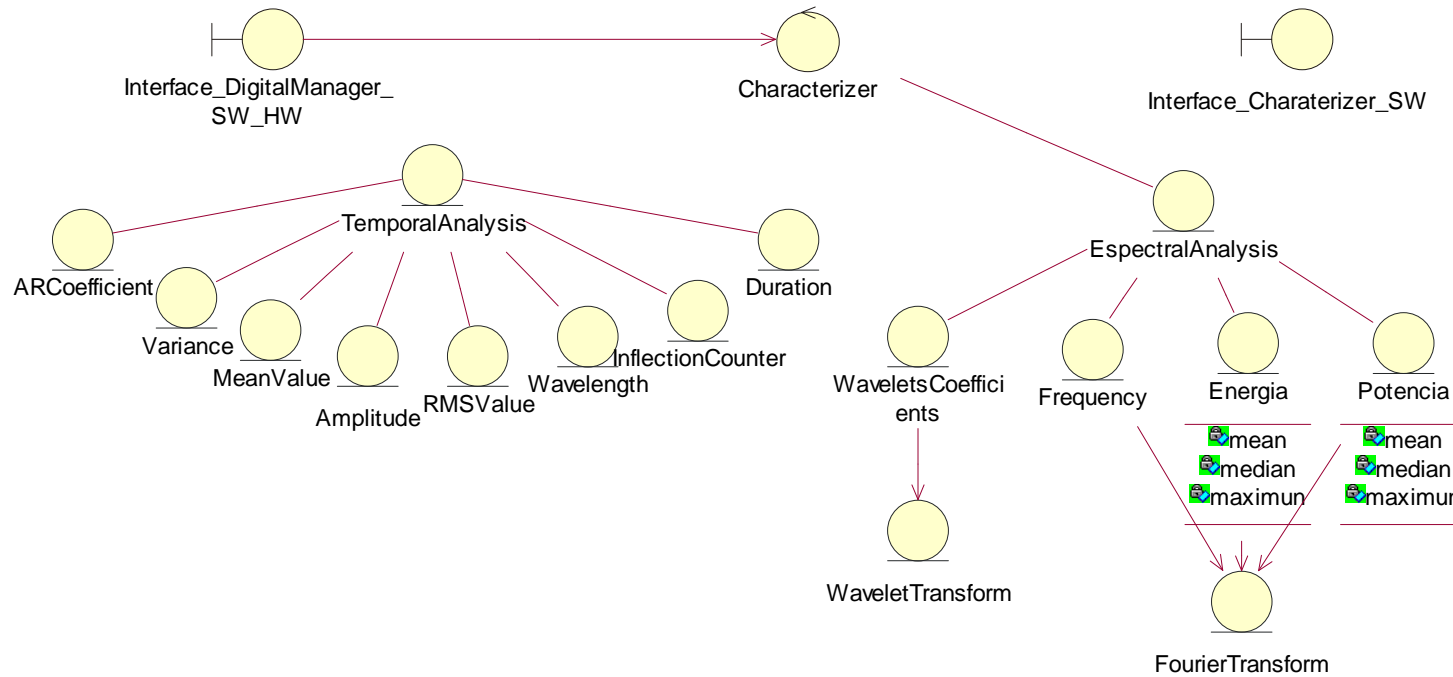


Figura 3.4. Modelo de análisis del Módulo Caracterizador de Señal.

A continuación se describe a través del uso de IDL el Módulo Caracterizador de la Señal (MCS) y se especifica las responsabilidades que se deben implementar en cada interfaz.

Es importante anotar que, del tamaño de la resolución del convertor análogo digital depende el tipo de dato para cada característica de la señal ME. Considerando la recomendación (Seniam, 2007) se prefiere un convertor de 16 bits de resolución, por lo tanto, se considera que el tipo de dato entero o *word* para esta aplicación es de 16bits.

A continuación se presenta la descripción IDL de algunas de las características de especial interés en la caracterización de señales ME, en la cual se especifica el nombre de la característica y el tipo de dato.

Module *Signal Characterizer*

```
{
  Interface Signal Characterizer SW
  {
    bool GetBasicTimeParameters( out integer Mean, out integer Median, out integer Max );
    bool GetRMSValue( out integer RMSValue );
    bool GetVariance( out integer Variance );
    bool GetDuration( out integer CntSamples, out integer SampleFrecuency );
    bool GetCntInflexion( out integer CntInflexion );
    bool GetWaveLength( out integer Wavelength );
    bool GetCntZeroCrossing( out integer CntZeroCrossing );
    bool GetARCoeficients( out float ARCoeficients[], out integer Length );
    bool GetBasicFrecParameters( out integer Mean, out integer Median, out integer Max );
    bool GetBasicEnergyParameters(out integer Mean,out integer Median, out integer Max );
    bool GetBasicPowerParameters(out integer Mean, out integer Median, out integer Max );
    bool GetWaveletsCoeficients( out float WaveletsCoeficients[], out integer Length );
  }
};
```

Módulo:		<i>Signal Characterizer</i>
Interfaz:		<i>Signal Characterizer SW</i>
Responsabilidad:		bool <i>GetBasicTimeParameters</i> (out integer <i>Mean</i> , out integer <i>Median</i> , out integer <i>Max</i>)
Arg. 1	Mean	Valor promedio del conjunto de muestras en el dominio temporal.
Arg. 2	Median	Valor que divide el área del conjunto de muestras en dos áreas iguales.
Arg. 3	Max	Registra el valor de amplitud máximo del conjunto de muestras de la señal ME durante la contracción muscular.
Retorno:	booleano	Retorna verdadero si el procesamiento es exitoso.
Tiempo de Respuesta:		Máximo 100ms. A una frecuencia de muestreo de 1 KHz.
Descripción:		Se almacena 100 ms de información de la señal ME, es decir, si la frecuencia de muestreo es de 1Khz el cálculo de los valores estadísticos: media, mediana y máxima se realizan a partir de la adquisición de 100 muestras de la señal ME. A partir de la muestra adquirida a los 100ms. el arreglo de almacenamiento de muestras se comporta como una sistema FIFO, es decir, elimina la muestra más antigua para darle paso a la más reciente; de esta manera es posible generar una respuesta de los parámetros temporales: valor medio, mediana y valor máximo cada periodo de muestreo.
Nota:		Particularmente, el valor máximo o amplitud se refiere a la intensidad de voltaje registrada durante la contracción el cual se utiliza frecuentemente como indicador de activación muscular. En (De Luca, 1997) se recomienda normalizar la amplitud con el valor máximo obtenido de una evaluación; a este valor se lo conoce como la máxima contracción voluntaria o MVC. Un cálculo particular es la amplitud de Wilson, el cual es utilizado en (Dumitru, 2002) y (Farina, y otros, 2004) para diagnosticar condiciones patológicas del músculo. Su cálculo se realiza de la siguiente manera $WAMP \equiv \sum_{k=1}^N f(x_k - x_{k+1}) \quad (3.1)$ Siendo $f(x) = 1$ si $ x_k - x_{k+1} > Umbral$ o $f(x) = 0$ en cualquier otro caso.

Tabla 3.6. MCS, Caracterizador de Señal SW, Obtener parámetros Temporales Básicos

Módulo:	<i>Signal Characterizer</i>	
Interfaz:	<i>Signal Characterizer SW</i>	
Responsabilidad:	bool <i>GetRMSValue(out integer RMSValue)</i>	
Arg 1.	<i>RMSValue</i>	Raíz cuadrada de la media de las muestras de la señal al cuadrado.
Retornos:	boolean	Retorna verdadero si se el procesamiento es exitoso.
Tiempo de Respuesta:	Máximo 100ms. Considerando una frecuencia de muestreo de 1 KHz.	
Descripción:	<p>El valor RMS viene del inglés, <i>Root Mean Square</i>, la cual es conocida como raíz cuadrática media o simplemente media cuadrática. El valor RMS es especialmente usado cuando las muestras de la señal a analizar pueden tomar valores positivos o negativos. Si los valores siempre son positivos o negativos, basta con calcular el valor medio. Matemáticamente, el valor RMS se calcula a partir de la raíz cuadrada de la media de las muestras de la señal al cuadrado, tal como se muestra a continuación:</p> $x_{rms} = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} = \sqrt{\frac{x_1^2 + x_2^2 + \dots + x_n^2}{n}}. \quad (3.2)$	
Nota:	Es considerado el valor más significativo dentro de los parámetros temporales de la señal ME, dado que este valor suministra una medida de la potencia de la señal (Delsys, 2009)	

Tabla 3.7. MCS, Caracterizador de Señal SW, Obtener valor RMS

Módulo:	<i>Signal Characterizer</i>	
Interfaz:	<i>Signal Characterizer SW</i>	
Responsabilidad:	bool <i>GetVariance(out integer Variance)</i>	
Arg 1.	<i>Variance</i>	Es una medida de la dispersión del conjunto de muestras de la señal.
Retornos:	boolean	Retorna verdadero si se el procesamiento es exitoso
Tiempo de Respuesta:	Máximo 100ms. Considerando una frecuencia de muestreo de 1 KHz.	
Descripción:	<p>La varianza es utilizada para diferenciar movimientos a partir de la información de diferentes músculos. Se define:</p> $VAR = \sigma^2 = \frac{1}{N-1} \sum_{k=1}^N (x_k - \bar{X})^2 \quad (3.3)$ <p>Donde “x_k” es cada una de las muestras adquiridas, “X” es el valor medio de las muestras y N es número total de muestras.</p>	
Nota:	En investigaciones como (Muñoz Burbano, y otros, 2004) selecciona la Varianza como parámetro característico para realizar la clasificación ya que esta refleja la fuerza muscular realizada durante una contracción.	

Tabla 3.8. MCS, Caracterizador de Señal SW, Obtener la varianza

Módulo:		<i>Signal Characterizer</i>
Interfaz:		<i>Signal Characterizer SW</i>
Responsabilidad:		bool <i>GetDuration</i> (out integer <i>CntSamples</i> , out integer <i>SampleFrequency</i>)
Arg 1.	<i>Duration</i>	Calcula el tiempo de activación muscular o presencia de señal ME.
Arg 2.	<i>SampleFrequency</i>	Frecuencia de muestreo de adquisición de la señal ME.
Retornos:	boolean	Retorna verdadero si se el procesamiento es exitoso.
Tiempo de Respuesta:		Depende directamente de la activación muscular.
Descripción:		Contabiliza el número de muestras que sobrepasan un valor umbral de amplitud con el fin de determinar el tiempo de activación muscular o presencia de señal ME. Para calcular en el tiempo de duración de la activación muscular es necesario multiplicar el contador de muestras <i>CntSamples</i> por el inverso de la frecuencia de muestreo <i>SampleFrequency</i> .
Nota:		La operación de multiplicación por el periodo de muestreo podría no ser necesaria dado que la cantidad de muestras que superan el umbral implícitamente contiene información de la duración de la activación y permite diferenciar una señales ME de otra.

Tabla 3.9. MCS, Caracterizador de Señal SW, Obtener duración

Módulo:		<i>Signal Characterizer</i>
Interfaz:		<i>Signal Characterizer SW</i>
Responsabilidad:		bool <i>GetCntInflexion</i> (out integer <i>CntInflexion</i>)
Arg 1.	<i>CntInflexion</i>	Contabiliza las veces que la pendiente de la señal cambia de signo.
Retornos:	booleano	Retorna verdadero si se el procesamiento es exitoso.
Tiempo de Respuesta:		100 ms correspondiente a la ventana de análisis.
Descripción:		Requiere de un contador que contabilice el número de veces que la pendiente de la señal cambia de signo y una base temporal o ventana de análisis en la cual se contabiliza los cambios de pendiente de la señal. El algoritmo de identificación del cambio de signo de la pendiente de la señal se reduce a comparar cada muestra x_k con sus vecinas x_{k-1} y x_{k+1} de la siguiente manera: $x_k > x_{k-1} \text{ y } x_k > x_{k+1} \quad (3.4)$ $x_k < x_{k-1} \text{ y } x_k < x_{k+1}$ Si la comparación es verdadera en cualquiera de los dos casos, se incrementa el contador de inflexiones.
Nota:		Se recomienda usar un umbral con el fin de reducir los incrementos del contador inducidos por ruido. Para ello se evalúa que la distancia entre x_k y sus vecinos sea mayor a un umbral. $ x_k - x_{k+1} \geq 0,01 \text{ V o } x_k - x_{k-1} \geq 0,01 \text{ V} \quad (3.5)$

Tabla 3.10 MCS, Caracterizador de Señal SW, Obtener contador de inflexión

Módulo:	<i>Signal Characterizer</i>	
Interfaz:	<i>Signal Characterizer SW</i>	
Responsabilidad:	bool <i>GetWaveLength(out integer Wavelength)</i>	
Arg 1.	<i>Wavelength</i>	Provee información de la longitud de onda en cada segmento
Retornos:	booleano	Retorna verdadero si se el procesamiento es exitoso
Tiempo de Respuesta:	100 ms correspondiente a la ventana de análisis de la seña ME	
Descripción:	<p>Longitud de Onda: Provee información de la longitud de onda en cada segmento. El valor resultante da una medida de la amplitud de la forma de onda, su frecuencia y duración en un solo parámetro. Su cálculo se realiza de la siguiente manera:</p> $l_o = \sum_{k=1}^N \Delta x_k \quad (3.6)$ <p>Donde $\Delta x_k = x_k - x_{k-1}$ es la diferencia ente muestras consecutivas.</p>	
Nota:		

Tabla 3.11. MCS, Caracterizador de Señal SW, Obtener longitud de onda

Módulo:	<i>Signal Characterizer</i>	
Interfaz:	<i>Signal Characterizer SW</i>	
Responsabilidad:	bool <i>GetCntZeroCrossing (out integer CntZeroCrossing)</i>	
Arg 1.	<i>CntZeroCrossing</i>	
Retornos:	<i>Booleano</i>	Retorna verdadero si se el procesamiento es exitoso
Tiempo de Respuesta:	100 ms correspondiente a la ventana de análisis de la señal ME	
Descripción:	<p>Requiere un contador el cual almacena el número de veces que la señal cruza por cero (eje de las abscisas). En (Romo, y otros, 2007) se propone su cálculo a través de la comparación con cero de dos muestras consecutivas.</p> $x_k > 0 \text{ y } x_{k+1} < 0, \text{ o } x_k < 0 \text{ y } x_{k+1} > 0 \quad (3.7)$ <p>En (Zecca, 2003), se realiza el cálculo por medio de la función signo.</p> $ZC = \sum_{i=1}^N \text{sgn}(-x_{ik}x_{i+1}) \quad (3.8)$ <p>Donde $\text{sgn}(x)$ es la función signo; vale 1 si $x=1$ y 0 para otro caso.</p>	
Nota:	<p>En cualquier caso se recomienda usar umbral con el fin de reducir los cruces por cero inducidos por el ruido.</p> $ x_k - x_{k-1} \geq 0,01 \text{ V} \quad (3.9)$	

Tabla 3.12. MCS, Caracterizador de Señal SW, Obtener contador de cruce por cero

Módulo:		<i>Signal Characterizer</i>
Interfaz:		<i>Signal Characterizer SW</i>
Responsabilidad:		bool <i>GetARCoeficients(out float ARCoeficients[], out integer Length)</i>
Arg 1.	ARCoeficients[]	Arreglo de coeficientes AR
Arg 2.	Length	Cantidad de coeficientes o tamaño del arreglo
Retornos:	Booleano	Retorna verdadero si se el procesamiento es exitoso
Tiempo de Respuesta:		La señal ME puede ser modelada como un modelo AR con una estacionariedad de 200 ms (Graupe, 1989).
Descripción:		<p>Se basan en el modelo paramétrico Auto-Regresivo (AR), El cual permite modela series de tiempo mediante la siguiente expresión:</p> $x(k) = -\sum_{i=1}^P a_i x(k-i) + \epsilon(k) \quad (3.10)$ <p>Donde a_i son los coeficientes AR, P el orden del modelo y $\epsilon(k)$ un término de error. (Graupe, y otros, 1975)</p> <p>Este modelo es igual al de un filtro FIR, por lo que tanto los coeficientes de este modelo como los de un filtro FIR del mismo orden son iguales.</p> <p>El objetivo del análisis de una serie temporal es el conocimiento de su patrón de comportamiento, para así prever su evolución futura, suponiendo que las condiciones no variarán, es decir, este parámetro tiene el inconveniente de suponer la condición de estacionariedad en la señal lo cual no se cumple para las señales ME; sin embargo, en un intervalo corto de tiempo puede considerarse como un proceso Gausiano estacionario (ver 3. EMG Pattern recognition) el cual permite ser representado a través de un modelo AR</p>
Nota:		<p>Se debe tener en cuenta que en este modelamiento paramétrico la caracterización de la señal de entrada es altamente dependiente del orden P del modelo, por lo tanto, la selección de este orden P es de especial atención.</p> <p>En (Ferguson, y otros, 2002) y (Birkendal, y otros, 2002) se recomienda que para caracterizar la señal ME a través de un modelo AR, el orden P del modelo AR sea $P = 5$ y $P = 10$. Por otro lado, en (Franco, y otros, 2004) se menciona que es suficiente para la caracterización de la señal ME con los 3 primeros coeficientes del modelo AR, ya sea de orden $P = 5$ o de orden $P = 10$, dado no se observa diferencia entre ellos.</p>

Tabla 3.13. MCS, Caracterizador de Señal SW, Obtener coeficientes AR

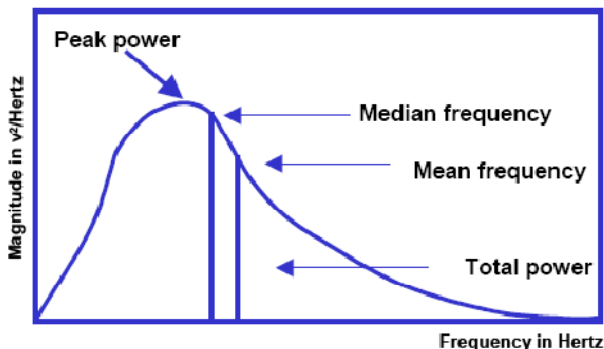
Módulo:		<i>Signal Characterizer</i>
Interfaz:		<i>Signal Characterizer SW</i>
Responsabilidad:		bool <i>GetFrecBasicParameters</i> (out integer <i>Mean</i> , out integer <i>Median</i> , out integer <i>Max</i>)
Arg 1.	Mean	Valor promedio del conjunto de muestras en el dominio de frecuencia.
Arg 2.	Median	Divide el área del espectro de frecuencia total en dos áreas iguales.
Arg 3.	Max	Frecuencia predominante o pico de frecuencia a la que oscila la señal
Retornos:	Booleano	Retorna verdadero si se el procesamiento es exitoso
Tiempo de Respuesta:		De acuerdo con (Salazar, y otros, 2006) el límite temporal es impuesto por el tiempo de grabación de la señal. Según (Romo, y otros, 2007) un tiempo adecuado es de 256 ms muestreando a una frecuencia de 1KHz.
Descripción:		<p>Frecuencia media: La frecuencia media es la frecuencia a la cual los datos están en el percentil 50. Representa la mitad de la gráfica del espectro de frecuencia</p> <p>Frecuencia mediana: El parámetro que divide el área del espectro de frecuencia total en dos áreas iguales.</p> <p>Frecuencia de mayor potencia: Es la frecuencia predominante o pico de frecuencia a la que oscila la señal. Esta característica se obtiene hallando el mayor valor entre el conjunto o arreglo de datos del espectro y encontrando su correspondencia en frecuencia.</p>  <p>Figura 3.5 Representación frecuencia media, mediana y potencia máxima</p>
Nota:		De acuerdo con (Salazar, y otros, 2006), al igual que los parámetros referenciados en esta responsabilidad es posible obtener otro tipo de parámetros basados en la estadística del conjunto de muestras de la señal ME. Numerosos autores tales como (De Luca, 2001), (Roy, y otros, 1997) y (Oddson, y otros, 1997) hacen uso de la frecuencia media como indicador de la fatiga muscular.

Tabla 3.14 MCS, Caracterizador de Señal SW, Obtener parámetros de frecuencia básicos

Módulo:		<i>Signal Characterizer</i>
Interfaz:		<i>Signal Characterizer SW</i>
Responsabilidad:		bool <i>GetBasicEnergyParameters</i> (out integer <i>Mean</i> , out integer <i>Median</i> , out integer <i>Max</i>);
Arg 1.	<i>Mean</i>	Valor promedio del conjunto de datos que conforman el espectrograma.
Arg 2.	<i>Median</i>	Valor de energía que divide el espectrograma en dos partes iguales.
Arg 3.	<i>Max</i>	Valor de energía predominante del conjunto de datos del espectrograma.
Retorno	boolean	Retorna verdadero si se el procesamiento es exitoso.
Tiempo de Respuesta:		De acuerdo con (Salazar, y otros, 2006) el límite temporal es impuesto por el tiempo de grabación de la señal. Según (Romo, y otros, 2007) un tiempo adecuado es de 256 ms muestreando a una frecuencia de 1KHz.
Descripción:		Energía Media y Energía Máxima: Específicamente para el cálculo de la Energía de la señal se realiza a partir del cuadrado del resultado de la STFT (Cáceres, 2007)
Nota:		A este cálculo también se conoce como espectrograma, el cual corresponde a la distribución conjunta <i>tiempo-frecuencia</i> de energía de la señal.

Tabla 3.15. MCS, Caracterizador de Señal SW, Obtener parámetros de energía básicos

Módulo:		<i>Signal Characterizer</i>
Interfaz:		<i>Signal Characterizer SW</i>
Responsabilidad:		bool <i>GetBasicPowerParameters</i> (out integer <i>Mean</i> , out integer <i>Median</i> , out integer <i>Max</i>);
Arg 1.	<i>Mean</i>	Valor promedio del conjunto de datos que conforman el periodograma
Arg. 2	<i>Median</i>	Valor de energía que divide el periodograma en dos partes iguales.
Arg. 3	<i>Max</i>	Valor de energía predominante del conjunto de datos del periodograma.
Retornos:	<i>Booleano</i>	Retorna verdadero si se el procesamiento es exitoso.
Tiempo de Respuesta:		De acuerdo con (Salazar, y otros, 2006) el límite temporal es impuesto por el tiempo de grabación de la señal. Según (Romo, y otros, 2007) un tiempo adecuado es de 256 ms muestreando a una frecuencia de 1KHz.
Descripción:		Potencia Media y Potencia Máxima: Se calcula a partir de la Densidad del Espectro de Potencia o PSD. Este cálculo también es conocido como periodograma el cual se obtiene como la magnitud al cuadrado de cada uno de los componentes de la Transformada Discreta de Fourier de la señal $x[n]$. El periodograma estimado del espectro de potencia de una señal X de longitud N es: $P(k) = \frac{ X(k) ^2}{N}, k = 1, \dots, N \quad (3.11)$
Nota:		Para estimar la potencia promedio y máxima se calcula el valor medio y máximo del conjunto de datos $P(K)$

Tabla 3.16. MCS, Caracterizador de Señal SW, Obtener parámetros de potencia básicos

Módulo:	<i>Signal Characterizer</i>	
Interfaz:	<i>Signal Characterizer SW</i>	
Responsabilidad:	bool <i>GetWaveletsCoeficients(out float WaveletsCoeficients[], out integer Length);</i>	
Arg 1.	<i>WaveletsCoeficients[]</i>	Arreglo de coeficientes Wavelets.
Arg 2.	<i>Length</i>	Numero de coeficientes en el arreglo <i>WaveletsCoeficients[]</i>
Retorno	<i>Booleano</i>	Retorna verdadero si se el procesamiento es exitoso.
Tiempo de Respuesta:	En (Romo, y otros, 2007) se sugiere que para el caso de la obtención de los coeficientes wavelets se debe contar con al menos 256ms de grabación de la señal y realizar el procesamiento de la obtención de los coeficientes en máximo esta ventana temporal.	
Descripción:	Se basa en la transformada wavelets discreta el cual es utilizada generalmente para la codificación de las señales y es esta característica la que permite encontrar la huella digital de las señales ME. Dada su utilidad y las diversas aplicaciones en el ámbito de las señales ME se amplía el tema en el Anexo 4.	
Nota:	(Zecca, y otros, 2002), (Ferguson, y otros, 2002) han obtenido un buen desempeño de clasificación de patrones. (Englehart, y otros, 1999) han trabajado especialmente comparando el uso de la Transformada Rápida de Fourier, los Cepstrales y el análisis por Wavelets; encontrando que la Transformada de Paquetes Wavelet (WPT) supera en efectividad el comportamiento de la Transformada Wavelet Discreta (DWT) y las representaciones tiempo-frecuencia de la transformada de Fourier, más presenta deficiencias por el prolongado periodo de cómputo requerido en comparación con las otras técnicas.	

Tabla 3.17. MCS, Caracterizador de Señal SW, Obtener coeficientes Wavelets

3.2.4 Módulo Gestor de Reconocimiento de Intención de Movimiento (MGRIM)

Este módulo tiene como objetivo reconocer la intención de movimiento del paciente, a partir de las características de la señal mioeléctrica.

Considerando el amplio grupo de características que se pueden extraer de una señal mioeléctrica y la elevada carga computacional que esto implica, se recomienda reducir dicha información al mínimo conjunto de características posibles, pero suficientes como para diferenciar entre distintas intenciones de movimiento. Algunos autores como (Ferguson, y otros, 2002) y (Zecca, y otros, 2002) nombran a este procedimiento como reducción de dimensionalidad por el hecho de agrupar las variables en un vector *n-dimensional*, donde *n* corresponde al número de características. En este caso el algoritmo de reconocimiento de intenciones de movimiento es utilizado sobre el conjunto de características reducidas y la señal medida es clasificada dentro de ciertos comandos predefinidos.

En el Capítulo 2 del presente documento se mencionan algunas técnicas de reducción de dimensionalidad.

La mano es el órgano del cuerpo humano que mejor adaptado está a la función de manipulación. La disposición de sus 27 huesos, movidos por 37 músculos permite un amplio margen de movimientos (Aguilar, 2008). Sin embargo, para efectos prácticos se recomienda limitar la cantidad de movimientos con el fin de que el dispositivo protésico los pueda efectuar. En (Mackenzie, y otros, 1994) se presentan las clasificaciones de Shelsinger, Napier y Arbib, de los movimientos de una mano humana teniendo en cuenta la forma del agarre y la fuerza para su ejecución.

De dichas clasificaciones se presenta el siguiente grupo de agarres: palmar, cilíndrico, gancho, pinza, lateral y esférico, a los cuales se les controla la proporción de apertura, al igual que la fuerza con la que se ejecuta el movimiento. Por lo tanto, se diferencian tres variables de control para la ejecución de un agarre: la forma del agarre, la proporción de cierre o apertura y la fuerza con la que se ejecuta.

De acuerdo con (Rohen, y otros, 2003) las manos en los seres humanos abarcan desde la muñeca hasta la yema de los dedos. Sin embargo, la mayoría de las prótesis no tiene movimiento de muñeca, es decir, realizan el movimiento de agarre, apertura y cierre de la mano, pero sin movimiento de muñeca (TabascoHoy.com)

La importancia de la implementación de la muñeca en los dispositivos protésicos radica en igualar los movimientos de una mano natural. Es decir que, los

movimientos realizados por el dispositivo prostético se asemejen en lo posible a los movimientos de una mano humana y se diferencien claramente de los movimientos de un robot.

Los movimientos de la muñeca se efectúan en torno a dos ejes: un eje transversal comprendido en un plano frontal, quien condiciona los movimientos de flexoextensión o movimiento de *pitch* y un eje anteroposterior comprendido en un plano sagital, quien condiciona los movimientos de aducción-abducción o movimiento de *yaw*. Otro posible movimiento es el de Pronación-Supinación el cual es equivalente al movimiento de *roll*, (Huaroto) y (Gazitúa, 2004).

A continuación se presentan una ilustración de los posibles movimientos para la muñeca.

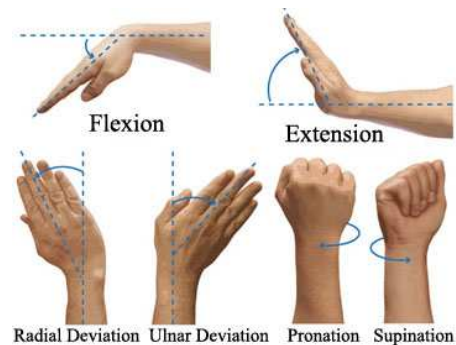


Figura 3.6 Movimientos Muñeca

Entre las prótesis que implementa un sistema para la muñeca se encuentran las propuestas por (Rodríguez, y otros, 2006) y (Zecca, 2003).

Por otro lado, dado que la finalidad del presente módulo es el reconocimiento de patrones a partir de una señal estocástica como lo es la señal mioeléctrica, a continuación se mencionan algunas de las herramientas matemáticas utilizadas para tal fin. No se profundiza en cada uno de este tema dado que sale del alcance del presente proyecto.

- Clasificador Bayesiano.
- Análisis del Discriminante Lineal.
- Redes Neuronales Artificiales.
- Lógica Difusa.
- Sistemas Neuro-Difusos.

Es importante considerar que entre más información se requiera extraer de la señal ME, tanto el tiempo de procesamiento como el número de señales ME provenientes de diferentes músculos será mayor. A su vez, desde el punto de vista de la usabilidad del paciente se incrementa la dificultad de operación, convirtiéndose en una herramienta muy compleja de controlar (O. B. H. Care).

Con el fin de simplificar el proceso de *identificación de intenciones de movimiento* se reduce la cantidad de información a extraer de la señal mioeléctrica.

En el actual modelo se propone una interfaz que puede ser implementada ya sea en hardware o software.

Con el propósito de conformar el conjunto de interfaces, a continuación se describe las interfaces para el Módulo Gestor de Reconocimiento de intención de movimiento (MGRIM).

Module *Motion Intention Recognition Manager (MIRMM)*

```
{  
  Interface Movement HW SW  
  {  
    bool GetHandGrip( out integer GripShape, out integer ApertureRatio, out integer Force );  
    bool GetWristMov( out integer WirstMov, out integer RotationRatio, out integer Force );  
  }  
}
```

Modelo de análisis en UML del Módulo Gestor de Reconocimiento de Intención de Movimiento (MGRIM)

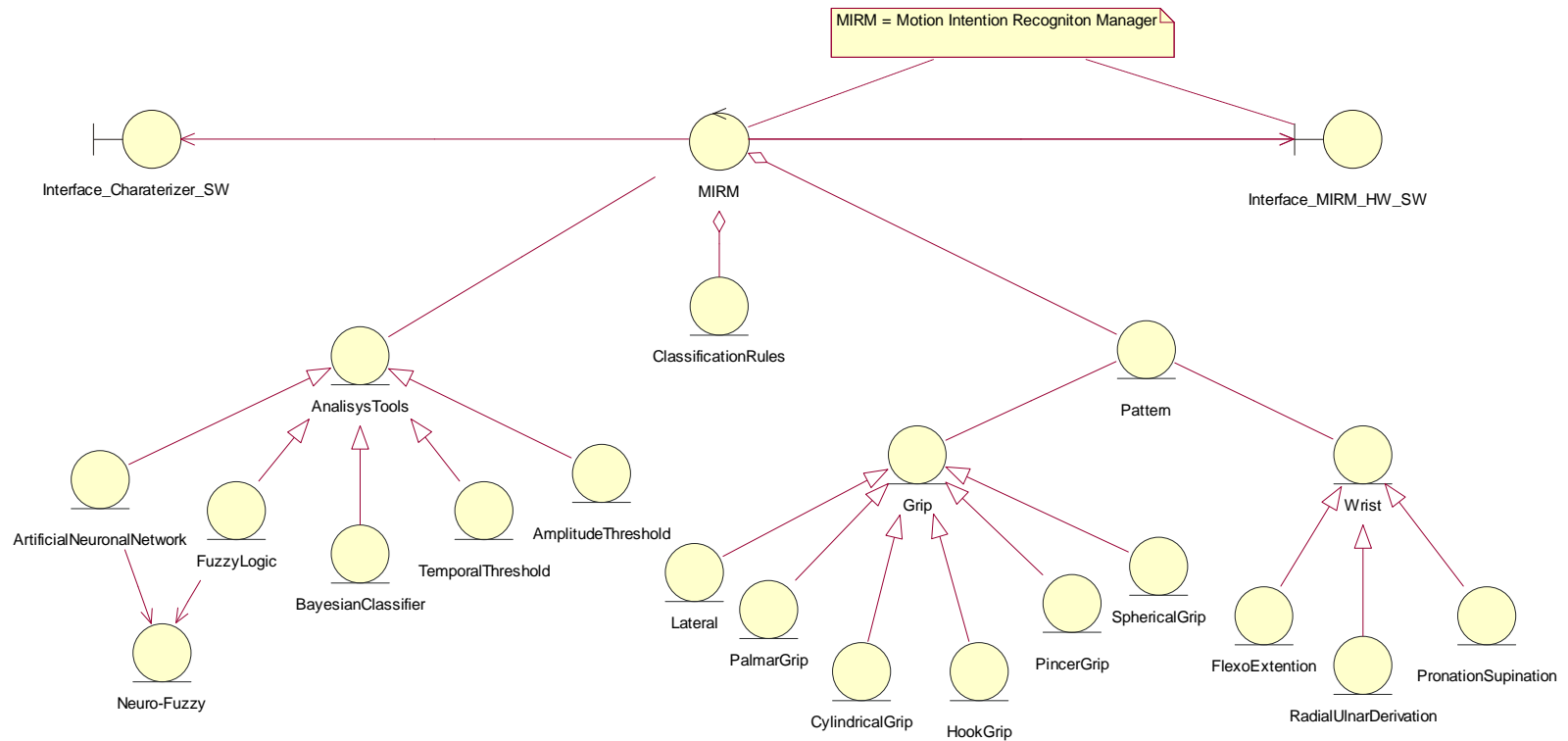


Figura 3.7 Modelo de análisis en UML del Módulo Gestor de Reconocimiento de Intención de Movimiento (MGRIM)

Módulo:		<i>Motion Intention Recognition Manager (MIRMM)</i>																		
Interfaz:		<i>Movement HW SW</i>																		
Responsabilidad:		bool <i>GetHandGrip</i> (out integer <i>GripShape</i> , out integer <i>ApertureRatio</i> , out integer <i>Force</i>);																		
Arg 1.	<i>GripShape</i>	Valor entero de un byte. A cada bit se le asigna una forma de agarre.																		
Arg 2.	<i>ApertureRatio</i>	Valor entre 0 y 180 grados.																		
Arg 3.	<i>Force</i>	Valor de fuerza con que se ejecuta el movimiento de agarre. Como la fuerza es una variable continua y por lo tanto puede adquirir infinitos valores, se propone reducir este conjunto a 256, es decir a una variable entera de un byte.																		
Retornos	Booleano	Si el proceso es exitoso retorna verdadero, de lo contrario retorna falso.																		
Tiempo de Respuesta:		Menor a 100ms. Para ejecutar el proceso reconocimiento de intención de movimiento se debe contar con el conjunto seleccionado de características de la señal ME. Dado que el periodo de actualización del conjunto de características es de 100ms, se cuenta con un periodo igual para realizar el proceso de clasificación y reconocimiento de la intención de movimiento.																		
Descripción:		<p>Esta interfaz tiene como finalidad reconocer la intención de movimiento del paciente a través de las características extraídas de la señal mioeléctrica.</p> <p>Se sugiere establecer un protocolo para la variable <i>GripShape</i>, en español formas de agarre, el cual puede ser utilizado tanto en una interfaz de comunicación hardware, serie o paralela, como en una interfaz software. El protocolo se presenta en la siguiente Tabla</p> <table border="1" data-bbox="531 1070 1372 1429"> <thead> <tr> <th>Tipo de agarre (<i>GripShape</i>)</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Palmar.</td> <td>1</td> </tr> <tr> <td>Cilíndrico.</td> <td>2</td> </tr> <tr> <td>Gancho.</td> <td>4</td> </tr> <tr> <td>Pinza</td> <td>8</td> </tr> <tr> <td>Lateral</td> <td>16</td> </tr> <tr> <td>Esférico</td> <td>32</td> </tr> <tr> <td>Reservado</td> <td>64</td> </tr> <tr> <td>Reservado</td> <td>128</td> </tr> </tbody> </table> <p>Tabla 3.18. Tipo de agarre y valor asignado.</p> <p>Por otro lado, la variable <i>ApertureRatio</i>, en Español Coeficiente de Apertura, y la variable <i>Force</i>, en Español Fuerza, complementan la información necesaria para la ejecución de un agarre dado que no es suficiente con especificar la forma o el tipo de agarre, sino también se requiere el grado de apertura y fuerza con que se ejecuta.</p>	Tipo de agarre (<i>GripShape</i>)	Valor	Palmar.	1	Cilíndrico.	2	Gancho.	4	Pinza	8	Lateral	16	Esférico	32	Reservado	64	Reservado	128
Tipo de agarre (<i>GripShape</i>)	Valor																			
Palmar.	1																			
Cilíndrico.	2																			
Gancho.	4																			
Pinza	8																			
Lateral	16																			
Esférico	32																			
Reservado	64																			
Reservado	128																			
Nota:		<p>Se recomienda que la variable <i>ApertureRatio</i> tome tan solo dos valores, uno para apertura y otro para el cierre del agarre.</p> <p>Respecto a la variable <i>Force</i>, en español fuerza, se recomienda que sea directamente el módulo de control del dispositivo a través de un arreglo de sensores quien se encargue de regular la fuerza con la que se ejecuta el agarre.</p> <p>La variable <i>Force</i> informa la cantidad de fuerza con que se debe ejecuta el agarre a partir de un proceso de normalización respecto a la mayor fuerza que puede efectuar. el dispositivo prostético.</p>																		

Tabla 3.19. MGRIM, Movimiento HW SW. Obtener tipo de agarre.

Módulo:	<i>Motion Intention Recognition Manager (MIRMM)</i>									
Interfaz:	<i>Movement HW SW</i>									
Responsabilidad:	bool GetWristMov(out integer WirstMov, out integer RotationRatio, out integer Force)									
Arg 1.	<i>WirstMov</i>	Valor entero de un byte. Informa el tipo de movimiento de la muñeca.								
Arg. 2	<i>RotationAngle</i>	Valor entero de un byte. Informa el ángulo en grados en el que el tipo de movimiento o <i>WirstMov</i> se debe ubicar.								
Arg. 3	<i>Force</i>	Valor de fuerza con que se ejecuta el movimiento de la muñeca. Como la fuerza es una variable continua y por lo tanto puede adquirir infinitos valores, se propone reducir este conjunto a 256, es decir a una variable entera de un byte.								
Retornos	<i>Booleano</i>	Si el proceso es exitoso retorna verdadero, de lo contrario retorna falso.								
Tiempo de Respuesta:	Para ejecutar el proceso reconocimiento de intención de movimiento se debe contar con el conjunto seleccionado de características de la señal ME. Dado que el periodo de actualización del conjunto de características es de 100ms, se cuenta con un periodo igual para realizar el proceso de clasificación y reconocimiento de la intención de movimiento.									
Descripción:	<p>Se tiene como finalidad reconocer la intención de movimiento respecto a la muñeca a través de las características extraídas de la señal mioeléctrica.</p> <p>A continuación se propone un protocolo para la variable <i>WirstMov</i>, en Español movimiento de muñeca, el cual se presenta en la siguiente tabla.</p> <table border="1" data-bbox="566 1104 1409 1285"> <thead> <tr> <th>Movimiento de muñeca (<i>WirstMov</i>)</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Flexo-Extensión</td> <td>1</td> </tr> <tr> <td>Aducción-Abducción</td> <td>2</td> </tr> <tr> <td>Pronación-Supinación</td> <td>3</td> </tr> </tbody> </table> <p>Tabla 3.20. Movimientos de la muñeca</p> <p>La variable <i>RotationAngle</i>, en Español ángulo de rotación, informa el ángulo en el que se debe ubicar la muñeca; ya sea referente al movimiento de <i>Flexo-Extensión (Pitch)</i>, <i>Aducción/Abducción (Yaw)</i> y <i>Pronación-Supinación (Roll)</i>.</p> <p>La variable <i>Force</i>, en Español fuerza, informa la cantidad de fuerza con que se ejecuta el movimiento a partir de un proceso de normalización. Es decir, que al valor máximo que puede adquirir la variable <i>Force</i>, en este caso 256, se le asigna la mayor fuerza que el dispositivo prostético pueda ejecutar.</p>		Movimiento de muñeca (<i>WirstMov</i>)	Valor	Flexo-Extensión	1	Aducción-Abducción	2	Pronación-Supinación	3
Movimiento de muñeca (<i>WirstMov</i>)	Valor									
Flexo-Extensión	1									
Aducción-Abducción	2									
Pronación-Supinación	3									
Nota:	De acuerdo con (Rodríguez, 2005) se considera que los paciente que usan este dispositivos prostéticos de mano, deberían recuperar tanto la forma del agarre como la precisión y la fuerza. Por lo tanto, dotar con dispositivo con los movimientos de la muñeca permitiría la manipulación de los objetos con mayor destreza en tareas tanto laborales como de la vida diaria.									

Tabla 3.21. MGRIM. Gestor de Reconocimiento de Intención de Movimiento. Movimiento HW SW .Obtener tipo de movimiento de la muñeca.

3.2.5 Módulo Controlador del Dispositivo (MCD)

El objetivo de un dispositivo prostético es suplir en lo posible las funcionalidades del miembro a reemplazar. Para el caso de la prótesis de mano, tanto la anatomía como la fuerza y la destreza para la ejecución de un movimiento deben asemejarse a los de la mano natural. Hecho que convierte la construcción del dispositivo prostético, en un desafío tecnológico de muy alto grado de complejidad.

El sistema de control ideal para un paciente es de tal forma que logre ejecutar los movimientos de la prótesis con las mismas señales que usaba antes de la amputación (Hauschild, y otros, 2007).

Antes de proponer una estrategia de control, es necesario conocer el sistema a controlar, por lo tanto, en la Figura 3.10, se presenta una descripción general de la anatomía de la mano humana, en la cual se diferencia los 27 huesos que la conforman. El carpio o muñeca tiene 8; el metacarpo o palma tiene 5 y los 14 huesos restantes son digitales. Gracias a sus 27 huesos posee cerca de 26 grados de libertad lo cual permite hacer movimientos de una complejidad extraordinaria, todo esto basado en la maniobrabilidad y precisión que se consigue con la oposición del pulgar o de la palma a los demás dedos (Aguilar, 2008)

Es importante resaltar, que del conjunto de articulaciones que conforman la mano humana en su totalidad son de tipo rotacional, o juntas rotoides. Por lo tanto, para la ejecución de un agarre o movimiento, el sistema de control del dispositivo prostético debe generar incrementos angulares por cada articulación.

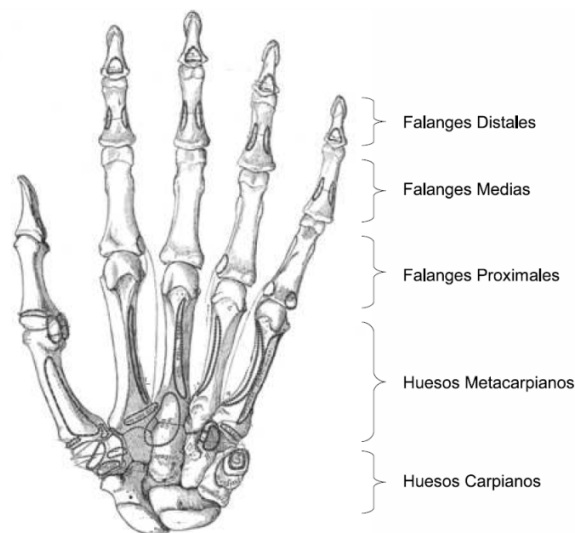


Figura 3.8. Anatomía de la mano humana

Dado el alto grado de complejidad de la mano humana, tanto ingenieros como investigadores han simplificado el diseño del dispositivo protésico. Como se evidencia en (Alonso, y otros, 2002), (Soares, y otros, 2003), (Hanafusa, y otros, 1977), (Mason, y otros), (Rodríguez, y otros, 2006), (Franco, y otros, 2004), el mecanismo de la prótesis consiste en una palma estática central o metacarpo de la que surgen entre dos y cinco dedos. Dicha palma está unida al antebrazo por una unión llamada muñeca o carpo. Aunque el pulgar no tiene falange media y por eso no es correcto considerarlo como dedo, siempre se encuentra presente en el diseño de los dispositivos protésicos.

Control Híbrido de fuerza y posición

Específicamente, el módulo de control recibe como entrada la información generada por el Módulo de Gestor de Reconocimiento de Intención de Movimiento (MGRIM) y las señales provenientes del Módulo Gestor Sensorial (MGS) para controlar tanto los movimientos como la fuerza de agarre a través de los motores instalados en el dispositivo

En el momento de implementar un agarre de un determinado objeto se deben tener en cuenta aspectos como: geometría o formar del objeto a sujetar, puntos de contacto de los dedos de la mano con el objeto, cálculo de las trayectorias de los contactos digitales con el objeto en el sistema de referencia de la mano, cálculo de la cinemática inversa que convierte trayectorias en el espacio cartesiano en incrementos angulares en las articulaciones que permiten la generación de la trayectoria planificada y finalmente, es necesario calcular la fuerza necesaria a emplear para manipular el objeto (Gaviria, y otros, 2008)

La información necesaria para la determinación de la geometría y puntos de contacto con el objeto se adquiere a través de la visión del paciente, quien es el responsable de genera las señales mioeléctricas a través de los músculos del antebrazo. A partir de las señales ME se reconoce la intención del movimiento y es aquí donde el sistema de control comienza a trabajar.

Luego de reconocer la intención de movimiento fijada por el paciente, se elije la trayectoria de pre-agarre el cual se representa como un conjunto de vectores o arreglos que contienen las secuencias angulares por cada articulación. La responsabilidad del sistema de control es cerciorarse de que dichas secuencias sean ejecutas por el dispositivo protésico de manera correcta. Por lo tanto, dependiendo

del tipo de motores instalados en la prótesis se hará necesario un sistema basado en sensores de posición angular que realimenten en cada momento el estado de las articulaciones y de esta manera, a partir del error entre estas dos señales generar una señal de control.

En el instante que el dispositivo prostético entra en contacto con el objeto (fase de agarre), empieza a tomar relevancia el sistema de control de fuerza. El cual tiene como responsabilidad mantener una aprehensión estable con el fin de que el objeto no se resbale ni se deforme o maltrate a causa de un exceso de fuerza. De la misma manera que para el control de posición, para el control de fuerza se hace necesario un sistema de realimentación sensorial que permita detectar en cualquier instante de tiempo la fuerza que imprime el dispositivo prostético sobre el objeto.

La finalidad del módulo de control radica en realizar dos acciones: ubicar los dedos en las posiciones deseadas según el tipo de agarre (consignas de posición) y posteriormente entrar en contacto con el objeto controlando la fuerza (consignas de fuerza). (Aguilar, 2008)

Nótese que tanto el control de posición como el de fuerza generan una orden de par a cada uno de los actuadores de las articulaciones para garantizar el seguimiento de las consignas. La acción simultánea de los dos controles puede generar conflicto dado que dos tipos diferentes de acción se están ordenando a los mismos actuadores. Por lo tanto, es necesario crear una estrategia de sincronización con el fin de que las acciones correctivas generadas por uno de los controles no interfieran con las señales del otro.

Por su fácil adaptación con los circuitos integrados se eligen los motores eléctricos como actuadores para la ejecución del movimiento de las articulaciones.

De acuerdo a la fuente de tensión que alimenta al motor se clasifican en: motores de corriente alterna y motores de corriente continua. Sin embargo, los motores de corriente continua a diferencia de los de corriente alterna se fabrican en pequeños tamaños y bajo peso; y a su vez permiten regular la velocidad del eje de rotación del motor. Dichas características son deseadas en la construcción de dispositivos portátiles, tales como, las prótesis robóticas de mano, en los cuales es necesario regular la velocidad de rotación para el control de los movimientos a ejecutar y a su vez, ser un dispositivo portátil y liviano para su manipulación.

En el Anexo 1 se presenta los conceptos básicos para el control del par de un motor de corriente continua considerando la velocidad y la potencia suministrada.

Modelo UML del Módulo Controlador del Dispositivo

En la Figura 3.9 se presenta un esquema habitual de un sistema de control realimentado, en donde el dispositivo prostético se encuentra representado por la planta o proceso, el sistema de realimentación sensorial por el módulo de medición y por último, la estrategia de control representada por el módulo controlador.

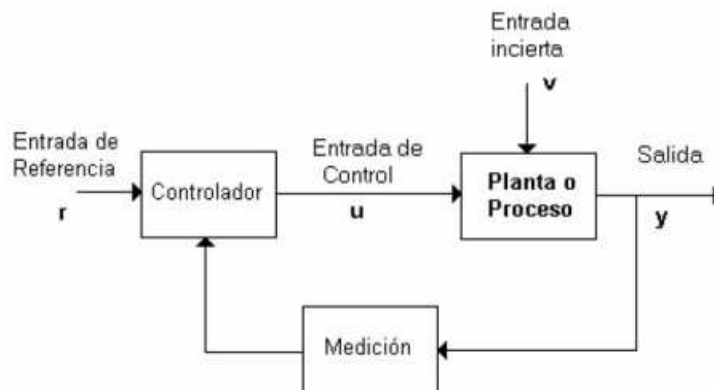


Figura 3.9. Esquema básico de control realimentado

En la Figura 3.10, el sistema de control se conforma principalmente de la clase controlador la cual tiene como responsabilidad adquirir y procesar la información proveniente de las clases entidad, tales como: la clase *trayectorias*, *control de fuerza*, *control de posición* y *reglas de control*, y posteriormente realimentar dicha información a través de las clases de interfaz.

En el modulo de control propuesto se reconocen tres clases de interfaz. La primera de ellas es la interfaz de *Intención de Movimiento* la cual es administrada por el módulo *Reconocimiento de Movimiento* y representa a la información de entrada al módulo de control. Las otras dos interfaces son: la Interfaz *Movimiento HW* y la interfaz de *Realimentación Sensorial*, las cuales se describen a continuación.

La interfaz *Movimiento HW* se utiliza para fijar los valores de par por cada una de las articulaciones que conforman el dispositivo prostético. Por otro lado, la interfaz de *Realimentación Sensorial* se utiliza para adquirir a través de sensores de posición y de fuerza los valores reales en los que se encuentre cada articulación.

Para definir las interfaces es necesario establecer las variables de entrada y salida del módulo de control. A continuación se identifican dichas variables y se describen las interfaces hardware/software que conforman el módulo.

Las variables de entrada al controlador están sujetas a las variables generadas por el *módulo de reconocimiento de intención de movimiento*, es decir, si se trata de un agarre las variables generadas son *Grip Shape*, *Aperture Ratio* y *Force* y para el movimiento de la muñeca son *WirstMov*, *Rotation Ratio* y *Strength*. La descripción detallada del comportamiento de dichas variables se presenta en el *módulo de Reconocimiento de Intención de Movimiento*.

Ahora bien, dependiendo de la resolución del sistema mecánico y del grado de precisión que se desea alcanzar para la ejecución de un movimiento se define el *tipo de dato* para las variables de la interfaz, ver Anexo 2.

Para construir un sistema de control realimentado de fuerza o posición se utilizan ya sea las herramientas del control clásico o moderno como del control inteligente. Sin embargo, no es relevante el tipo de herramienta utilizada para definir las interfaces del módulo de control. Es decir, sin importar la estrategia de control las variables de entrada y salida del módulo deben ser las mismas.

A continuación, se presenta el diagrama de análisis *Modelo, Vista, Control* para el sistema de control propuesto utilizando notación UML y su descripción a través de IDL.

Descripción del Módulo Controlador del Dispositivo a través de IDL

```
Module Device Controller
{
  Interface Device Controller
  {
    void GetFingerTorque( out integer Proximal, out integer Medial, out integer Distal );
    void GetFingerAngle( out integer Proximal, out integer Medial, out integer Distal );
  }
}
```

Modelo de análisis en UML del Módulo de Control del Dispositivo

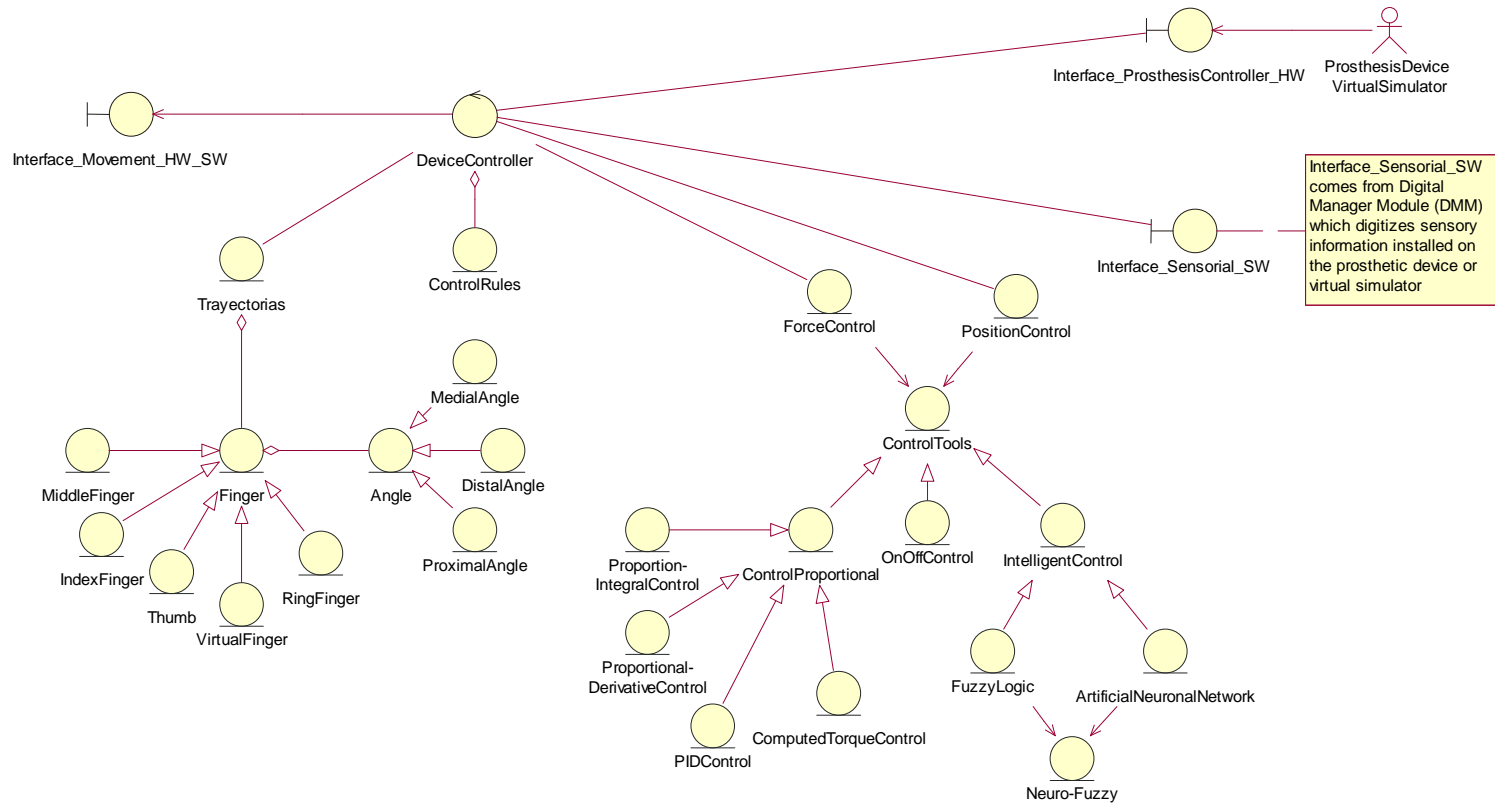


Figura 3.10 Modelo de análisis en UML del Módulo controlador del Dispositivo

Módulo:	<i>Device Controller</i>	
Interfaz:	<i>Device Controller</i>	
Responsabilidad:	Get FingerTorque (out integer <i>Proximal</i> , out integer <i>Medial</i> , out integer <i>Distal</i>)	
Arg. 2	<i>Proximal</i>	Valor de par de la articulación proximal
Arg. 3	<i>Medial</i>	Valor de par de la articulación media
Arg. 3	<i>Distal</i>	Valor de par de la articulación distal
Tiempo de Respuesta:	Se encuentra dentro de la ventana de adquisición del arreglo de muestras de la señal ME, Se consideran entre 100 y 120ms como el tiempo mínimo necesario para procesar la señal ME, reconocer una nueva intención de movimiento y ubicar al dispositivo protésico en su nueva posición.	
Descripción:	<p>A través del control de la potencia suministrada a los motores se controla tanto la posición como la fuerza de las articulaciones que conforman el dispositivo protésico de mano.</p> <p>Control de posición: El objetivo es ubicar al conjunto de dedos del dispositivo protésico de mano en una posición en particular. Para ello, el sistema de control fija un valor de par por cada una de las articulaciones que lo conforman. Dichos valores se representan por los argumentos de salida de tipo entero denominados: <i>Proximal</i>, <i>Medial</i> y <i>Distal</i>.</p> <p>De acuerdo al tipo de actuador a utilizar para generar la rotación de una articulación, las señales de control pueden ser: palabras digitales, valores analógicos o señales moduladas por ancho de pulso o PWM. Sin embargo, por su grado de simplicidad y adaptación entre los sistemas digitales y analógicos se sugiere el uso de la señal PWM.</p> <p>Dado que un dispositivo protésico de mano cuenta con al menos dos dedos, debe existir una función <i>SetFingerPosition()</i> por cada dedo que lo conforma. Es decir, si el dispositivo cuenta con el dedo pulgar y el dedo índice, debe existir una función <i>SetPulgarPosition()</i> y <i>SetIndicePosition()</i> con sus respectivos argumentos.</p> <p>Control de fuerza: Es de interés controlar la fuerza de contacto entre los dedos del dispositivo protésico y la superficie de los objetos. Para ello se hace uso de la información proporcionadas por sensores de fuerza ubicados en la superficie de los dedos, equivalentemente como lo haría una mano humana.</p> <p>En el instante en que entran en contacto los sensores de fuerza con el objeto se conoce la fuerza con la que se inicia la fase de agarre. Ahora bien, el sistema de control de fuerza impone un orden de par en los actuadores para garantizar el seguimiento de acuerdo a las consignas de fuerza preestablecidas.</p>	
Nota:	Aunque el tiempo mínimo de respuesta sea de 100ms, no siempre es deseable reaccionar en tan corto tiempo. Igualmente es importante realizar movimientos lentos y precisos que permitan ser fácilmente controlados por el usuario. Particularmente la función <i>SetPulgarPosition()</i> , el argumento <i>Medial</i> se fija a cero o simplemente no se considera.	

Tabla 3.22. MCP.Controlador de prótesis. Obtener torque de los dedos

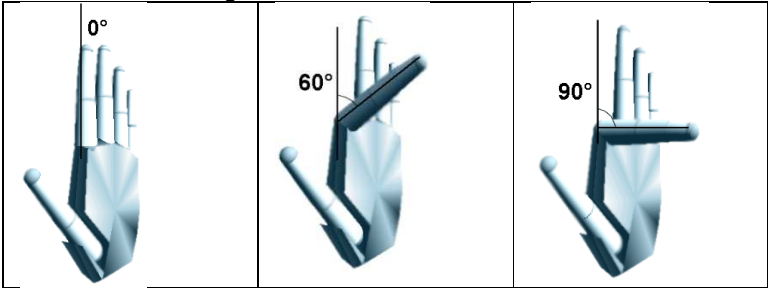
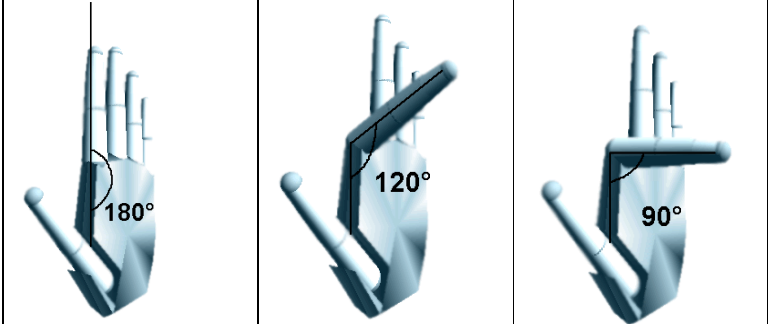
Módulo:	<i>Device Controller</i>	
Interfaz:	<i>Device Controller</i>	
Responsabilidad:	Get FingerAngle (out integer <i>Proximal</i> , out integer <i>Medial</i> , out integer <i>Distal</i>)	
Arg. 2	<i>Proximal</i>	Valor angular de la articulación proximal
Arg. 3	<i>Medial</i>	Valor angular de la articulación media
Arg. 3	<i>Distal</i>	Valor angular de la articulación distal
Tiempo de Respuesta:	Se encuentra dentro de la ventana de adquisición del arreglo de muestras de la señal ME, Se consideran entre 100 y 120ms como el tiempo mínimo necesario para procesar la señal ME, reconocer una nueva intención de movimiento y ubicar al dispositivo protésico en su nueva posición.	
Descripción:	<p>Dependiendo del tipo de aplicación, el controlador fija ya sea el valor de par del actuador o directamente el ángulo en que debe ubicarse cada una de las articulaciones en un instante determinado.</p> <p>Por ejemplo, si la aplicación es de un simulador virtual, como el objetivo del presente proyecto, el sistema de control fija directamente el valor angular de cada articulación. Aunque, si se cuenta con el modelo matemático del actuador, se podría fijar el valor de par, el cual se aproximaría más a la realidad.</p>	
Nota:	<p>Es importante definir una convención de medida angular para la rotación de las articulaciones, con el fin de que, tanto el sistema de control como el simulador virtual o los actuadores de la prótesis, interpreten de la misma manera un valor angular y logren ejecutar el movimiento deseado. A continuación se define las dos posibles referencias para medir el ángulo de rotación de una articulación, sin embargo, se recomienda la convención que cuando la mano esté extendida, las articulaciones se encuentren en cero grado de rotación.</p> <p>Mano extendida, 0 grados de rotación</p>  <p>Mano extendida, 180 grados de rotación</p> 	

Tabla 3.23. MCP. Controlador de prótesis. Obtener ángulo de los dedos

3.2.6 Módulo Gestor Sensorial (MGS)

Para que el dispositivo protésico de mano funcione de manera natural es necesario contar con un sistema sensorial inspirado en la mano humana. A pesar que existen muchas sensaciones que se han perdido en un proceso de amputación, de acuerdo con (Rodríguez, 2004) y (Duarte, y otros, 2000) las sensaciones de contacto, fuerza y deslizamiento son las más útiles para el agarre.

Por otro lado, para el correcto funcionamiento del control del dispositivo protésico es necesario que el controlador conozca la posición angular de cada articulación, así como, la información sensorial que permita proteger al dispositivo de sobre carga o del mal uso.

Dado que un solo tipo sensor no es suficiente para detectar distintas sensaciones se propone la conformación de un sistema integrado de sensores. Por ejemplo, (Jockusch, y otros, 1997) emplea un sensor simple FSR para mediar la fuerza ejercida y un sensor de tipo piezoeléctrico PZT para detectar el efecto del deslizamiento.

Por otro lado, es de igual importancia la disposición de dichos sensores en el dispositivo protésico. En aplicaciones de agarre, tales como las realizadas por (Speeter, 1990) se conforma una matriz de sensores táctiles que cubren la superficie de la mano robótica Utah / MIT Dextrous. En (Rodríguez, 2005) se determinan algunos puntos para la ubicación de sensores de fuerza y deslizamiento sobre la mano PROTECI basados en las observaciones del agarre de la mano humana.

A continuación se define las interfaces y el modelo de análisis en UML del Módulo Gestor Sensorial.

Module Sensorial Manager

```
{
  Interface Sensorial SW
  {
    void GetFingerPosition( out Analog Proximal, out Analog Medial, out Analog Distal);
    void GetFingerExtremePosition( out integer Proximal, out integer Medial, out integer Distal );
    void GetFingerOverLoad( in Analog Proximal, in Analog Medial, Analog Distal );
    void GetFingerForce( in Analog Proximal, in Analog Medial, in Analog Distal );
    void GetPalmForce( in Analog Palm);
    void GetFingerSlip( in Analog Proximal, in Analog Medial, in Analog Distal );
  }
}
```

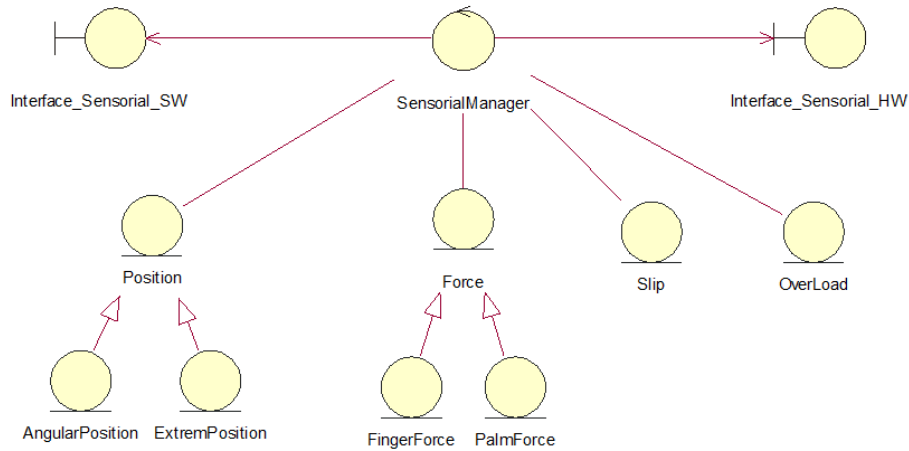



Figura 3.11. Modelo de análisis en UML del Módulo Gestor Sensorial

Módulo:	<i>Sensorial Manager</i>	
Interfaz:	<i>Sensorial SW</i>	
Responsabilidad:	bool <i>GetFingerPosition</i> (in Analog Proximal , in Analog Medial , in Analog Distal)	
Arg. 2	<i>Proximal</i>	Posición angular de la articulación proximal
Arg. 3	<i>Medial</i>	Posición angular de la articulación media
Arg. 3	<i>Distal</i>	Posición angular de la articulación distal
Tiempo de Respuesta:	Este tiempo debe ser al menos el tiempo de adquisición de la señal ME. Si se adquiere la señal cada 1ms, entonces, la realimentación sensorial debe realizarse en un tiempo menor a este. Lo anterior tiene como finalidad que el módulo de control del dispositivo MCD logre realizar cualquier tipo de corrección ya sea en la trayectoria del agarre o en la fuerza de ejecución.	
Descripción:	La presente responsabilidad tiene como finalidad conocer la posición angular de las articulaciones que conforman el dispositivo protésico de mano en cada instante tiempo. Los valores de posición angular serán realimentados al sistema de control quien es el encargado de generar una acción correctiva de acuerdo a las consignas de posición preestablecidas según el tipo de agarre.	
Nota:	Para tal fin, se recomienda el uso de sensores de flexión, los cuales, al ser doblados la resistencia en sus terminales disminuye en relación con el radio de curvatura. Dichos sensores actualmente son comercializados por <i>Images Scientific Instruments</i> . Se recomienda para su búsqueda las palabras claves: <i>Flex Bend Sensor</i> . De acuerdo con (Duarte, y otros, 2000) otro tipo de sensores igualmente utilizados para la medición de posición de las articulaciones de dispositivos protésicos son los sensores de Efecto Hall. Para su funcionamiento se instala una fuente de campo magnético en una de piezas acopadas en la articulación y en la otra pieza se instala el sensor. En el momento de la rotación de la articulación varía la distancia entre la fuente magnética y el sensor causando que la intensidad de campo magnético detectado por el sensor cambie proporcionalmente a la distancia de la fuente.	

Tabla 3.24. MGS. Gestor sensorial HW. Realimentación Sensorial HW. Obtener posición de los dedos

Módulo:	<i>Sensorial Manager</i>	
Interfaz:	<i>Sensorial SW</i>	
Responsabilidad:	<i>GetFingerExtremePosition(in integer Proximal, in integer Medial, in integer Distal)</i>	
Arg. 2	Proximal	Estado sensores de posición extrema, articulación proximal.
Arg. 3	Medial	Estado sensores de posición extrema, articulación media
Arg. 3	Distal	Estado sensores de posición extrema, articulación distal.
Tiempo de Respuesta:	Este tiempo debe ser al menos el tiempo de adquisición de la señal ME. Si se adquiere la señal cada 1ms, entonces, la realimentación sensorial debe realizarse en un tiempo menor a este. Lo anterior tiene con finalidad que el módulo de control del dispositivo MCD logre realizar cualquier tipo de corrección ya sea en la trayectoria del agarre o en la fuerza de ejecución.	
Descripción:	<p>Si el controlador de posición ejecuta algún tipo de agarre y no existe ningún objeto entre los dedos, la ejecución de dicho movimiento finalizará al llegar a uno de los sensores de posición extrema. Por lo tanto, esta responsabilidad se utiliza como sistema de protección en caso de fallar los sensores de posición.</p> <p>Idealmente, se debería instalar un par de sensores por cada articulación, uno en cada extremo del movimiento. Por lo tanto, el tipo de dato recomendable es el entero de 2 bits. Un bit por cada sensor.</p> <p>Por otro lado, los sensores de posición extrema son útiles en el proceso de auto-calibración. Gracias a su ubicación en las articulaciones permiten conocer un valor de posición fija el cual servirá de referencia para la calibración de los sensores de posición.</p>	
Nota:	<p>Como sistema de detección de posición extrema se recomiendan tanto los sensores de efecto hall como los sensores ópticos. Cualquiera de estos sistemas se debe ubicar de tal forma que no interfieran con el funcionamiento de la mano.</p> <p>Los sensores de efecto hall necesitan de un imán el cual actúa como fuente de campo magnético. Sin embargo, es posible que un campo magnético externo se ubique cerca de la mano introduciendo ruido al controlador. Por esta razón se recomienda acompañar a los sensores magnéticos con un sistema de filtrado que permita detectar exclusivamente la señal proveniente del imán.</p> <p>Los sensores ópticos están compuestos por un par dispositivos, un emisor y un receptor. De igual forma, se debe considerar que los sensores ópticos pueden ser activados de manera equivocada por señales de luz externas al sistema. Por lo tanto, al igual que el sistema basado en efecto hall se necesita de un sistema de filtrado que permita eliminar el ruido de la señal.</p>	

Tabla 3.25. MGS. Gestor sensorial HW. Realimentación Sensorial HW. Obtener posición extrema de los dedos

Módulo:		Gestor Sensorial
Interfaz:		<i>Realimentación Sensorial HW</i>
Responsabilidad:		GetFingerOverLoad (in integer <i>Proximal</i> , in integer <i>Medial</i> , in integer <i>Distal</i>)
Arg. 2	Proximal	Valor proporcional a la sobrecarga de la articulación proximal
Arg. 3	Medial	Valor proporcional a la sobrecarga de la articulación Medial
Arg. 3	Distal	Valor proporcional a la sobrecarga de la articulación Distal
Tiempo de Respuesta:		Este tiempo debe ser al menos el tiempo de adquisición de la señal ME. Si se adquiere la señal cada 1ms, entonces, la realimentación sensorial debe realizarse en un tiempo menor a este. Lo anterior tiene con finalidad que el módulo de control del dispositivo MCD logre realizar cualquier tipo de corrección ya sea en la trayectoria del agarre o en la fuerza de ejecución.
Descripción:		<p>El sistema de control de posición tiene como finalidad ubicar a las articulaciones en una posición determinada. Si en algún momento dado, ocurre un bloqueo mecánico de la articulación es posible que el controlador fuerce la articulación con el fin de alcanzar la posición deseada, incluso hasta el punto de dañar el sistema mecánico que la conforma.</p> <p>Por tal motivo, se recomienda instalar un sistema de detección de sobrecarga con el objetivo de proteger al sistema mecánico de un posible daño, quizás irreversible.</p>
Nota:		Se recomienda instalar una resistencia Shunt en cada uno de los actuadores con el fin de conocer en cada instante el valor de corriente suministrada al actuador. De esta manera es posible fijar un umbral de operación, el cual si es superado, el controlador deberá suspender el suministro de corriente y ubicar a la articulación en una posición de operación segura.

Tabla 3.26. MGS. Gestor sensorial. . Realimentación Sensorial HW. Obtener sobrecarga de los dedos

Módulo:	Gestor Sensorial	
Interfaz:	<i>Realimentación Sensorial HW</i>	
Responsabilidad:	bool Get <u>Finger</u> Force (in Analog <i>Proximal</i> , in Analog <i>Medial</i> , in Analog <i>Distal</i>)	
Arg. 2	<i>Proximal</i>	Valor de fuerza ejercida por la articulación proximal
Arg. 3	<i>Medial</i>	Valor fuerza ejercida por la articulación media
Arg. 3	<i>Distal</i>	Valor fuerza ejercida por la articulación distal
Retornos:	Booleano	Si el proceso es exitoso retorna verdadero, de lo contrario retorna falso.
Tiempo de Respuesta:	Este tiempo debe ser al menos el tiempo de adquisición de la señal ME. Si se adquiere la señal cada 1ms, entonces, la realimentación sensorial debe realizarse en un tiempo menor a este. Lo anterior tiene con finalidad que el módulo de control del dispositivo MCD logre realizar cualquier tipo de corrección ya sea en la trayectoria del agarre o en la fuerza de ejecución.	
Descripción:	<p>A partir del contacto con el objeto es necesario conocer el valor de fuerza ejercido por las articulaciones del dispositivo prostético Dicho valor será procesado por el sistema de control el cual generará una acción correctiva de acuerdo a la estrategia preestablecida.</p> <p>De acuerdo al dispositivo prostético se define el tipo y ubicación de los sensores de fuerza. Sin embargo, a través de esta responsabilidad se permite adaptar un sensor o arreglo de sensores por cada una de las falanges que conforman un dedo.</p> <p>Por lo tanto, el sistema de control conocerá el par ejercido por cada articulación y a partir de esta información generará una acción correctiva.</p>	
Nota:	<p>Entre las restricciones para la escogencia del sensor de fuerza se encuentran las dimensiones, la cuales deben ser reducidas y tener un diseño antropomórfico adaptable a la estructura de los dedos.</p> <p>De acuerdo a lo anterior, se recomienda el uso de sensores de fuerza resistivos o FSR, los cuales se utilizan en un elevado número de aplicaciones que requieren un control de fuerza cuando el requisito de la precisión no es indispensable. Por ejemplo, en (Dario, 1988), (Speeter, 1990), (Kyberd, 1995) (Rodriguez, 2005) hacen uso de FSR para medir la fuerza ejercida por los dedos del dispositivo prostético.</p>	

Tabla 3.27. MGS. Gestor sensorial. Realimentación Sensorial HW. Obtener fuerza ejercida por los dedos

Módulo:	Gestor Sensorial	
Interfaz:	<i>Realimentación Sensorial HW</i>	
Responsabilidad:	GetFingerSlip (out integer Proximal, out integer Medial, out integer Distal)	
Arg. 2	<i>Proximal</i>	Equivalente al valor de deslizamiento en la articulación proximal
Arg. 3	<i>Medial</i>	Equivalente al valor de deslizamiento en la articulación medial
Arg. 3	<i>Distal</i>	Equivalente al valor de deslizamiento en la articulación distal
Tiempo de Respuesta:	Este tiempo debe ser al menos el tiempo de adquisición de la señal ME. Si se adquiere la señal cada 1ms, entonces, la realimentación sensorial debe realizarse en un tiempo menor a este. Lo anterior tiene con finalidad que el módulo de control del dispositivo MCD logre realizar cualquier tipo de corrección ya sea en la trayectoria del agarre o en la fuerza de ejecución	
Descripción:	El objetivo de esta responsabilidad es detectar en el agarre un posible deslizamiento del objeto y realimentar al sistema de control con el fin de genera una acción correctiva a través del ajuste del par producido por el actuador que rige a la articulación. De igual forma que en una mano humana, la ubicación de los sensores de deslizamiento se realiza a lo largo de los dedos del dispositivo protésico, por lo tanto, se dispone de una variable por cada sensor o arreglo de sensores que valor de des	
Nota:	Se sugiere un sistema de detección basado en un arreglo sensores piezoeléctricos PZT, los cuales podrían ser ubicados en cada una de las falanges que conforma un dedo del dispositivo protésico La vibración en los sensores PZT ocurre cuando son sometidos a una fuerza externa. Por esta razón se debe seleccionar el sentido de la barra y de esta manera concuerde con la dirección del probable deslizamiento de los objetos. Es muy importante definir el sentido de deslizamiento de los objetos agarrados por la prótesis con el fin de generar la acción correctiva más adecuada.	

Tabla 3.28. MGS. Gestor sensorial HW. Realimentación Sensorial HW. Obtener el valor del deslizamiento de los dedos

Módulo:	Gestor Sensorial	
Interfaz:	<i>Realimentación Sensorial HW</i>	
Responsabilidad:	GetPalmForce(in Analog Palm)	
Arg. 1	<i>Palm</i>	Valor de fuerza ejercida sobre la palma de la prótesis de mano.
Tiempo de Respuesta:	Este tiempo debe ser al menos el tiempo de adquisición de la señal ME. Si se adquiere la señal cada 1ms, entonces, la realimentación sensorial debe realizarse en un tiempo menor a este. Lo anterior tiene con finalidad que el módulo de control del dispositivo MCD logre realizar cualquier tipo de corrección ya sea en la trayectoria del agarre o en la fuerza de ejecución	
Descripción:	Considerando que uno de los puntos más importantes en la mayoría de los agarres es la palma de la mano se dispone de una responsabilidad que permite conocer el valor de fuerza que ejercen los dedos en el momento de la oposición hacia ella.	
Nota:	Para la detección de fuerza ejercida sobre la palma se recomienda de igual manera que para los dedos, el uso de sensores de fuerza resistivos o FSR.	

Tabla 3.29. MGS. Gestor sensorial. Realimentación Sensorial HW. Obtener fuerza ejercida por la palma de la prótesis

3.2.7 Módulo Gestor de Realimentación (MGR)

El principal interés de este módulo es proporcionarle al usuario algunas sensaciones con el fin de conocer el estado del dispositivo protésico, especialmente en la percepción del agarre y la generación de señales ME.

Aunque este módulo sea considerado de baja acogida entre los pacientes debido a que las técnicas hasta el momento utilizadas no se asemejen a una sensación natural, es de gran importancia no perderlo de vista para futuros avances en el área.

Las sensaciones se transmiten al usuario a través de señales visuales, estímulos táctiles, sonidos, temperatura y especialmente a través de la estimulación eléctrica funcional (FES) la cual se aplica ya sea en el músculo o directamente en los hilos nerviosos presentes en el miembro amputado.

De acuerdo con las necesidades del paciente y las especificaciones del dispositivo protésico se determina el tipo de información de realimentación al usuario. Por este motivo, el MGR se asocia a los demás módulos con el fin de conocer la información generada por dichos módulos y realimentarla al usuario.

El MGR tiene como finalidad ajustar la intensidad de la señal transmitida al usuario de acuerdo a la información adquirida de los demás módulos. Por ejemplo, la información a realimentar puede estar relacionada con la señal mioeléctrica, la fuerza transmitida, el instante en que el dispositivo entra en contacto con el objeto o incluso el nivel de batería. Por lo tanto, de acuerdo al estado de la anterior información el MGR ajusta la señal de realimentación.

A continuación se presenta la descripción de las interfaces y responsabilidades que conforman el Módulo Gestor de Realimentación.

Descripción IDL

```
Module FeedbackManager
{
  Interface Feedback
  {
    void GetSxFeedBack( in integer SxType, out Analog Value );
  }
}
```

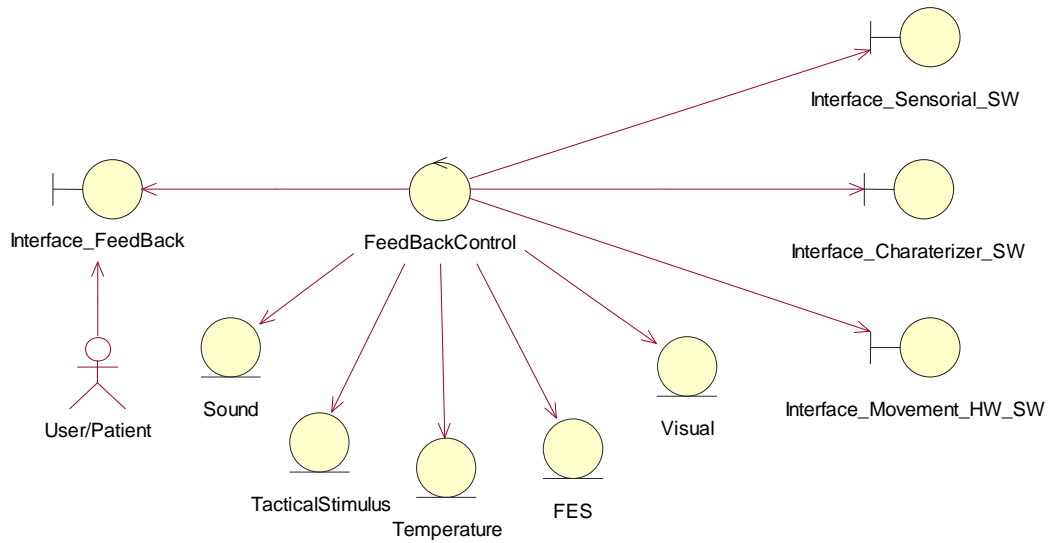


Figura 3.12. Modelo de análisis en UML del Módulo Gestor de Realimentación (MGR)

Módulo:	<i>Feedback Manager</i>	
Interfaz:	<i>Feedback</i>	
Responsabilidad:	<i>GetSxFeedBack(in integer SxType, out integer Value)</i>	
Arg. 1	SxType	Tipo de señal de realimentación.
Arg. 2	Value	Valor analógico que adquiere la señal de realimentación.
Tiempo de Respuesta:	Para este módulo, no habrían restricciones estrictas de tiempo, sin embargo, se recomienda que el tiempo de respuesta sea menor a 100ms, el cual corresponde a la ventana de análisis de la señal ME.	
Descripción:	<p>La presente responsabilidad tiene como finalidad obtener el tipo y el valor de la señal de realimentación al usuario. Tales señales pueden ser:</p> <ul style="list-style-type: none"> • Estimulación Eléctrica Funcional (FES) • Estimulación Táctil. • Señales visuales • Sonido. • Temperatura <p>La información a realimentar puede estar relacionada con cualquiera de los estados del dispositivo prostético, por ejemplo: señal mioeléctrica, señal de fuerza, señal de vibración o deslizamiento, entre otras.</p>	
Nota:	A pesar que existen muchas sensaciones que se han perdido en un proceso de amputación se considera en (Rodríguez, 2005) que las sensaciones de contacto, fuerza de agarre y deslizamiento son las más útiles para este tipo de dispositivo.	

Tabla 3.30. MGR. Gestor de Realimentación, realimentación, Obtener señal de realimentación

Capítulo 4

4 Verificación del Conjunto de Interfaces

4.1 Responsabilidades del conjunto normalizado de interfaces

A continuación se define una de las posibles implementaciones de un simulador virtual de prótesis de mano robótica haciendo uso de las interfaces propuestas.

Dado que el objetivo de este capítulo se enfoca en la utilización del conjunto de interfaces y no en la implementación de un simulador virtual en particular, se limitará el alcance de tal manera que se evidencie el uso de las interfaces y no la implementación de diversas estrategias para el desarrollo de prótesis de mano robótica, aunque es necesario hacer uso de una de ellas como insumo para el método de verificación.

Una de las estrategias utilizada en dispositivos prostéticos, tales como, Proteci y Ottobock se basa en la detección de la duración de la actividad muscular a través de la señal ME. De acuerdo a dicha duración se reconoce la intención de movimiento a ejecutar, por ejemplo, abrir o cerrar la mano. Con esta información, el módulo controlador del dispositivo fija las trayectorias para la ejecución del movimiento y a

través de un conjunto de sensores se realimenta tanto de la posición de cada articulación como de la fuerza que le imprime el dispositivo protésico al objeto.

Considerando la estrategia expuesta anteriormente se define las responsabilidades de los módulos que integran al simulador virtual de prótesis de mano robótica.

Para la definición de responsabilidades de los módulos es importante referenciarse al conjunto normalizado de interfaces para los componentes de un simulador virtual de prótesis de mano robótica descrito en el capítulo 3; el cual presenta un modelo normalizado de los módulos que conforman el dispositivo protésico. Cabe resaltar que, en este punto el objetivo es definir una arquitectura general y no especificar la manera en que se implementa los módulos.

Ahora bien, partiendo del modelo normalizado se selecciona por cada uno de los módulos las responsabilidades que aportan directamente a la implementación de la estrategia.

Inicialmente se consideran los módulos gestor analógico (MGA) y gestor digital (MGD), de los cual se implementarán respectivamente las responsabilidades de adquirir la señal analógica pre-procesada y adquirir un arreglo de muestras digitalizadas de dicha señal.

Del módulo caracterizador de la señal (MCS) se implementará la responsabilidad *GetDuration*, en español obtener duración, con el fin de detectar el tiempo de la actividad muscular; a partir de este valor el módulo de reconocimiento de intención de movimiento (MGRIM) se encarga de identificar el tipo de movimiento a ejecutar. Conociendo el movimiento deseado por el paciente, el módulo de control del dispositivo (MCD) se encarga de escoger el conjunto de trayectorias relacionadas con dicho movimiento y ejecutarlas ya sea fijando los ángulos o el valor de torque correspondiente a cada uno de los motores que controlan a las articulaciones del dispositivo protésico.

El módulo gestor sensorial (MGS) tendrá la responsabilidad de conocer en cada instante la posición de cada una de las articulaciones del dispositivo protésico virtual, por lo tanto, tendrá a disposición la función *GetFingerPosition* o en español obtener la posición de cada dedo. Dicha función será accedida por el Módulo Controlador del Dispositivo (MCD) con el fin de realizar las correcciones pertinentes. Por lo tanto, para lograr este objetivo, se requiere que el dispositivo protésico virtual informe de la posición angular en que se encuentra cada una de las articulaciones que conforman al dispositivo protésico.

Por último, el módulo de realimentación (MGR) debe generar una señal sonora con el fin de realimentar al usuario acerca de la terminación de la ejecución del movimiento por parte del dispositivo protésico. Esta función se implementará a través de la función *GetSoundLevel* o en español obtener nivel de sonido.

A partir de la anterior descripción y del conjunto normalizado de interfaces se obtiene la arquitectura presentada en la figura 4.1.

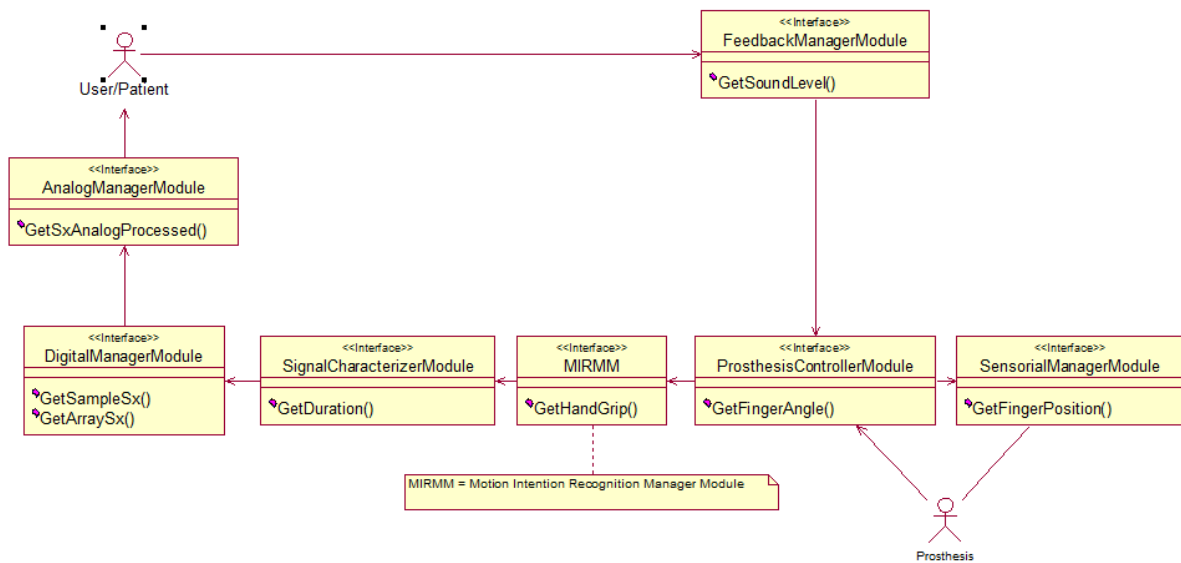


Figura 4.1. Arquitectura del simulador virtual

4.2 Descripción del simulador virtual

De acuerdo a la arquitectura presentada en la figura 4.1 se describirá las características más relevantes de cada uno de los módulos que la conforma. Para contextualizar las interfaces propuestas a una solución en particular, inicialmente se define las herramientas hardware y software.

Dado que el simulador virtual es un paso previo a la utilización de una prótesis real, se decide realizar la implementación de la arquitectura directamente en un microcontrolador, de esta manera, el presente desarrollo servirá como herramienta inicial para la creación de un sistema protésico real

Por su bajo costo y las amplias prestaciones matemáticas tanto para operaciones de punto fijo como de punto flotante se elige el microcontrolador

MCF51CN128 de *Freescale*, el cual a su vez cuenta con módulo de conversión análoga digital de 12 bits utilizado para digitalizar la señal ME y un módulo de comunicación serie asíncrona por el que se transmite las señales de control al dispositivo protésico virtual

Como lenguaje de programación se elige el lenguaje C, gracias a que es un lenguaje orientado a la implementación de sistemas operativos, es apreciado por la eficiencia del código que produce y es altamente difundido entre los programadores de sistemas embebidos.

Con el fin de garantizar una adecuada respuesta al usuario en el tiempo requerido y facilitar la administración de las tareas que integran al dispositivo protésico se decide utilizar un sistema operativo de tiempo real. Para la selección del sistema operativo a utilizar se tuvo en cuenta criterios tales como, lenguaje de implementación, portabilidad entre procesadores de diferentes marcas, soporte y costo, entre otros. En la Tabla 4.1 se presenta una comparación entre algunos de los sistemas operativos reconocidos en el mercado.

Item	MQX	uC/OS	FreeRTOS
Expulsivo	SI	SI	SI
Cooperativo	NO	NO	SI
ROM Mínimo	12Kbytes	6Kbytes	2Kbytes
RAM Mínima	2,5Kbytes	-	-
Tareas, ISR	SI	SI	SI
Prioridades	SI	SI	SI
Semaforos	SI	SI	SI
Mutex	SI	SI	SI
Colas	SI	SI	SI
Listas	NO	NO	SI
Lenguaje	C	C	C
Portabilidad	Solamente Coldfire	Diversas plataformas	Diversas plataformas
Soporte	US\$ 12.000 / año	Incluido en licencia	Comunidad libre
Código abierto	NO	SI	SI
Múltiples protocolos	USB TCP/IP	USB TCP/IP	USB TCP/IP
Costo	US\$ 0	US\$ 6.000	US\$ 0

Tabla 4.1 Comparación entre algunos de los sistemas operativos reconocidos en el mercado

A partir de la Tabla 4.1 se deduce que FreeRTOS cumple con los criterios de selección relevantes para el desarrollo del presente proyecto.

Para este proyecto el actor que se muestra en la conjunto normalizado de interfaces nombrado como *ProsthesisDevice* o en Español dispositivo protésico es una mano de apariencia humana conformada por una estructura ósea que permite simular los movimientos de una mano real. Según la experiencia de profesionales del área de fisioterapia del Centro Integral de Rehabilitación de Colombia, CIREC, para lograr una exitosa adaptación a un dispositivo protésico real, es recomendable que el paciente se ejercite previamente controlando un video juego con el fin de estimular la generación de señales ME.

Por las razones expuestas se decide utilizar el *Framework XNA Games* desarrollado por *Microsoft* para la creación de juegos de la consola *XBOX 360*, en el cual se implementará el dispositivo protésico de mano virtual y se dejará abierta la posibilidad de controlar un juego a partir de las señales ME. Otra razón que viabiliza la utilización de esta herramienta es que se trata de una herramienta gratuita de libre distribución, incluso para fines comerciales.

4.2.1 Módulo Gestor Analógico (MGA)

Se conforma principalmente de un circuito de amplificación diferencial basado en el integrado AD620 fabricado por Analog Device. El diseño del circuito se encuentra en la nota de aplicación (Analog Device, 1999).

A continuación se describe de una manera más detalla cada una de las responsabilidades que conforma la arquitectura el dispositivo protésico.

De acuerdo con el conjunto normalizado de interfaces propuesto, el módulo MGA debe implementar las responsabilidades *GetSX* y *GetSxAnalogProcessed*. Por lo tanto, para la implementación de la interfaz *GetSX* se utilizan electrodos superficiales desechables, tales como los que se presenta en la figura 4.2



Figura 4.2. Electrodo superficiales

Respecto a la interfaz *GetSxAnalogProcessed* se implementó el circuito presentado en la figura 4.2, el cual recibe como parámetro de entrada a la señal ME adquirida directamente de los electrodos y retorna la señal ME procesada de manera analógica.

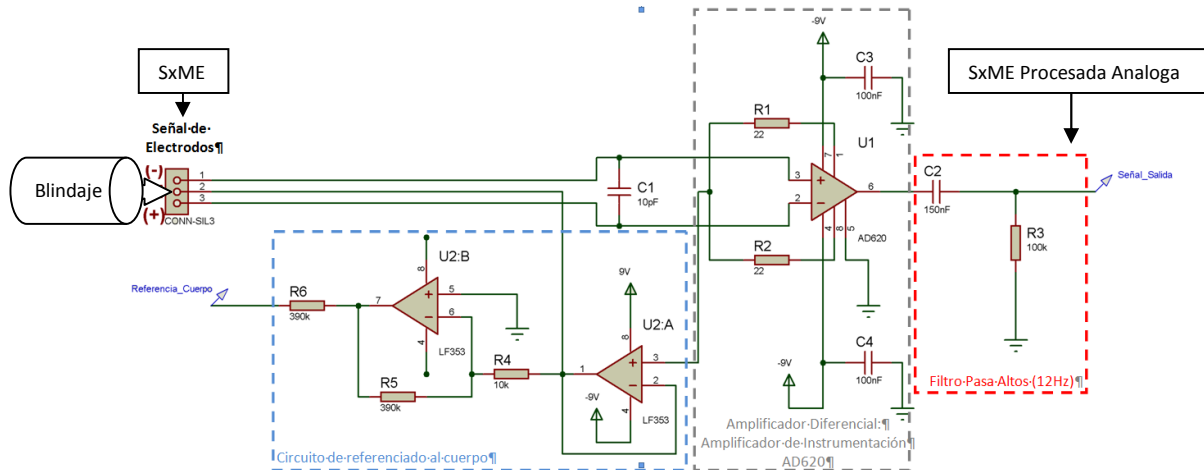


Figura 4.3 Alternativa de diseño para implementar la interfaz *GetSxAnalogProcessed*

Es importante resaltar que el circuito presentado en la figura 4.3 es uno de los posibles diseños a implementar por la interfaz *GetSxAnalogProcessed*. Otra posible implementación es la suministrada en las notas de aplicación de la hoja técnica del amplificador de instrumentación INA2128 (Burr, 2009)

4.2.2 Módulo Gestor Digital (MGD)

Este módulo ejecuta una tarea periódica con periodo de 1ms la cual se encarga de adquirir una muestra de la señal ME procesada analógicamente a través del módulo de conversión analógica digital y posteriormente someterla a un proceso de filtrado. Dadas las exigencias temporales que impone el procesamiento digital de señales, a esta tarea se le asigna la mayor prioridad entre las tareas que controla el sistema operativo.

Para este proyecto se implementó un filtro de promedio móvil en donde la muestra filtrada es almacenada en un arreglo de 200 muestras con el fin de tener disponible los últimos 200ms de información.

Para conocer el resultado de este proceso se realiza a través de las responsabilidades *GetSampleSx* y *GetArraySx* definidas previamente en el conjunto

de interfaces propuesto. Cada que se adquiere una muestra de la señal se cede un semáforo a través de la función *xSemaphoreGive* con el fin de Informarle al Módulo Caracterizador de la Señal (MCS) que se dispone de una nueva muestra de la señal.

4.2.3 Módulo Caracterizador de la Señal (MCS)

Este módulo ejecuta una tarea que se encarga de detectar el lapso en que las muestras superan un umbral determinado por el 50% de la máxima contracción voluntaria de los músculos en cuestión.

Esta tarea espera a que el Módulo Gestor Digital (MGD) le envíe una señal informado que cuenta con una nueva muestra de la señal ME para ser procesada. De igual manera como el MDG le envía una señal al módulo MCS, este envía una señal con el fin de informarle al Módulo Gestor de Reconocimiento de Intención de Movimiento (MGRIM) que se cuenta con un nuevo dato de activación muscular.

Para acceder a la información procesada por este módulo se realiza a través del la interfaz *GetDuration* la cual retorna el tiempo en milisegundo que duró dicha activación.

4.2.4 Módulo Gestor de Reconocimiento de Intención de Movimiento (MGRIM)

Este módulo ejecuta una tarea que se encarga de determinar la intención de movimiento del paciente a partir de las características generadas por el MCS.

Esta tarea espera a que el Módulo Caracterizador de la Señal le envíe una señal informando que se cuenta con una nueva característica para ser procesada. En este punto se relacionan las características de la señal ME a una intención de movimiento en particular.

Como implementación del Módulo de Reconocimiento de Intención de Movimiento (MGRIM), para el presente proyecto se determina que si la duración de la actividad muscular se encuentra entre 200ms y 500ms la intención del paciente es cerrar la mano y si la actividad muscular está entre 500ms y 1s la intención es abrirla. Estos valores podrían ser ajustados de acuerdo al criterio de usabilidad de cada paciente.

Otra posible relación entre el tiempo de actividad muscular y la intención de movimiento es detectar pulsos de corta duración que superen el 50% de la Máxima Contracción Voluntaria MCV. Por ejemplo, si el usuario genera dos pulsos ME continuos separados entre 100 y 200ms, la intención de movimiento es cerrar la mano; y ya sea con la detección de uno o tres pulsos la intención de movimiento podría ser abrirla.

En el instante en que el MGRIM detecta una nueva intención de movimiento este envía una señal al Módulo Controlador del Dispositivo (MCD) con el fin de informarle que se cuenta con una nueva orden para ser ejecutada por el dispositivo protésico.

La información de la intención de movimiento se retorna a través de la interfaz *GetHandGrip* la cual es definida por el conjunto normalizado de interfaces propuesto en este proyecto.

4.2.5 Módulo Controlador del Dispositivo (MCD)

Este módulo corre una tarea que se encarga de controlar a través de la rotación de las articulaciones los movimientos ejecutados por el dispositivo protésico.

Al recibir una señal por parte del MGRIM informando de una nueva intención de movimiento, el MCD procede a ejecutarlo. Para ello, inicialmente se selecciona la trayectoria a ejecutar y se conoce la posición actual de cada articulación a través del Módulo Gestor Sensorial (MGS) y su interfaz *GetFingerPosition*.

A partir de esta información se procesa un algoritmo de control el cual genera una serie de órdenes que ubican a las articulaciones en la posición predefinida por las trayectorias. Dichas órdenes se retornan a través de la interfaces *GetFingerAngle* o *GetFingerTorque* propuestas en el conjunto normalizado de interfaces.

Es importante mencionar que para el presente proyecto se fijarán los movimientos directamente a través de ángulos, los cuales son referenciados con la convención de “mano extendida, cero grados de rotación”.

4.2.6 Módulo Gestor Sensorial (MGS)

Este módulo ejecuta una tarea encargada de mantener la información actualizada de la posición angular de cada una de las articulaciones. En este caso, el dispositivo protésico virtual será quien suministre dicha información la cual será accedida por el MGS a través del puerto de comunicación asíncrona y retornada a través de la interfaz *GetFingerPosition*.

4.2.7 Módulo Gestor de Realimentación (MGR)

Este módulo ejecuta una tarea encargada de realimentar al usuario determinada información relacionada con el estado del dispositivo protésico. Particularmente en este proyecto se define que la información a realimentar es el momento en que el dispositivo protésico termina la ejecución de un movimiento definido por el usuario. Dicha información es retornada a través de la interfaz *GetSoundLevel*.

4.2.8 Dispositivo protésico Virtual

Se trata de una mano de tres dimensiones con apariencia humana la cual es controlada a través del puerto serial por el Módulo Controlador del Dispositivo (MCD), el cual envía la posición angular para cada una de las articulaciones.

Para este proyecto el puerto serial es configurado a 115200 baudios, con un bit de stop, sin paridad y sin control de flujo. Esta configuración no necesariamente debe ser de esta manera, por lo tanto, es importante especificarla con el fin de unificar tanto la configuración del puerto del MCD como del computador en dónde se ejecuta el dispositivo protésico virtual.

En la Figura 4.4 se presenta una ilustración de un movimiento de apertura y cierre de mano.

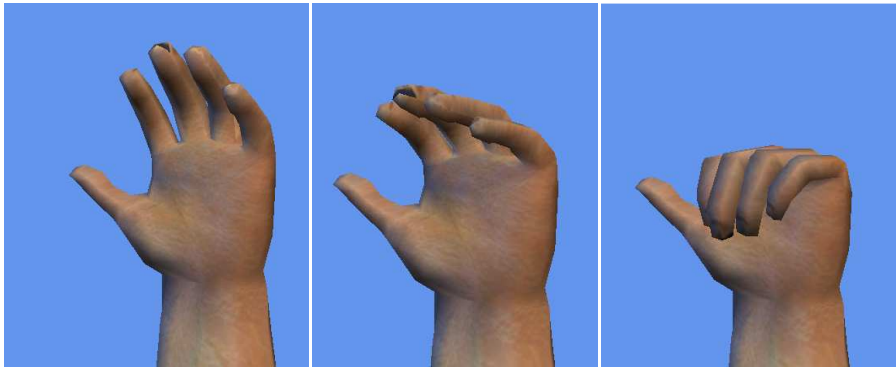


Figura 4.4 Ilustración de movimiento de apertura y cierre de mano

La mano se construye en 3D Max Studio basada en un modelo humanoide predefinido por la herramienta, la cual conserva las proporciones de una mano humana, permitiendo modificar las dimensiones de cada uno de los huesos e incluso cambiar la piel con el fin de acondicionar su aspecto .En la Figura 4.5 se presenta el modelo esquelético del dispositivo prostético de mano virtual desarrollado en el presente proyecto.

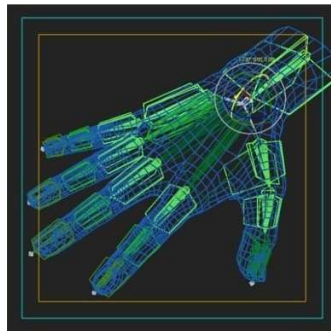


Figura 4.5 Modelo esquelético del dispositivo prostético virtual

En el lenguaje C, las interfaces de comunicación entre módulos se representan a través del archivo de cabecera de extensión `.h`. Este archivo a su vez tiene asociado un archivo de extensión `.c` en donde se implementan las interfaces.

A continuación se presenta un ejemplo de cómo se documenta las interfaces al interior del código de programación. Sin embargo, para acceder al proyecto completo, incluyendo el dispositivo virtual se sugiere acceder al siguiente repositorio a través de la herramienta de control de versiones Mercurial.

<http://code.google.com/p/standard-interfaces-for-robotic-hand-prosthesis/>

```

/*****
 * Filename: DMM.h
 * Developer(s): Juan Sebastian Cabrera
 * Language: ANSI C
 * Abstract: This module is used to acquire myoelectric signals digitally. Use
 *           the analogic digital converter installed in the microcontroler
 *           working with a resolution of 12 bits
 *****/
#ifndef __DMM_H
#define __DMM_H

/* ANSI C Headers */
#include <stdio.h>

/* Operation System Header */
#include "FreeRTOS.h"
#include "task.h"
#include "queue.h"
#include "semphr.h"

/*****
 * GLOBAL FUNCTION DECLARATION
 *****/
/*****
 * Function name      : DMM_GetSampleSxME
 * returns            : boolean data. Returns true if the Module have a new
 *                    sample or false if the sample is still in acquisition
 *                    process
 * arg1               : puiSampleSxME, which return a pointer to the last
 *                    sample
 * Created by         : Juan Sebastian Cabrera
 * Date created       : 27 may 09
 * Description        : This responsability return the last mioelectric sample
 * Note               : The ADC resolution is 12 bits
 *****/
bool DMM_GetSampleSxME( uint16_t * puiSampleSxME );

#endif /* __DMM_H */
/*****
 * END OF FILE
 *****/

```

Figura 4.6. Código ejemplo para la descripción de interfaces en lenguaje C.

Capítulo 5

5 Conclusiones, Aportes y Trabajos Futuros

Este trabajo de grado ha propuesto un conjunto normalizado de interfaces para la creación de dispositivos prostéticos y especialmente para el entrenamiento de pacientes con amputación de mano, los cuales serían candidatos a una prótesis controlada por señales mioeléctricas.

En este capítulo se presentan las conclusiones obtenidas por el proyecto y los aspectos más importantes relacionados con su desarrollo. Por último realiza una serie de propuestas para trabajos futuros.

5.1 Conclusiones

Entre los aspectos más importantes de contar con un conjunto normalizado de interfaces es que permitió al equipo de desarrollado comprender de manera lógica y temporal el comportamiento global del sistema. Logrando así mejorar la intercomunicación entre los integrantes y generar propuestas para el desarrollo del simulador virtual más acordes a las necesidades.

Dada la baja interdependencia entre los módulos propuesto en el conjunto normalizado de interfaces y al uso del sistema operativo de tiempo real se logró construir un diseño simple, segmentado, flexible y mantenible. Consecuentemente se mejora la sincronización entre los integrantes del grupo de desarrollo dado que cada uno de ellos puede trabajar de manera independiente, implementado su propia estrategia y acogiéndose a la definición del conjunto de interfaces propuesto en este proyecto.

Al igual que la famosa frase de Napoleón, divide y vencerás, los grupos de desarrollo dividen los componentes del proyecto entre sus integrantes con el fin de lograr soluciones modulares y en menor tiempo, Sin embargo los trabajos previos identificados en la revisión bibliográfica se enfocan en realizar un diagrama arquitectónico de bloques, sin efectuar una definición adecuada entre las interfaces de los componentes hardware/software, convirtiendo así, al conjunto normalizado de interfaces propuesto en una valiosa herramienta que facilita la organización y coordinación del desarrollo.

Las interfaces planteadas permiten probar en un módulo en particular distintos algoritmos o estrategias de control, sin necesidad de modificar el resto de los módulos, de esta manera, conseguir el mejor ajuste o puesta a punto para el dispositivo protético real. De igual manera le brindan al sistema flexibilidad para intercambiar los componentes que integran el dispositivo protético. Por ejemplo, si en lugar de utilizar como fuente de control las señales ME se utilizan comandos de voz, es suficiente con cambiar la interfaz de adquisición de la señal y acoplarse al protocolo propuesto. De esta manera se conservan los demás componentes del sistema y la prótesis ejecuta la función para la cual fue prevista. Esta característica de modularidad le permite al dispositivo ser escalable en el tiempo de acuerdo con los avances científicos que se vayan presentando.

Las especificaciones de la interfaz suministrarán información a los desarrolladores del componente para que conozcan qué servicios están disponibles en otros componentes y cómo pueden acceder a ellos. Las especificaciones claras y no ambiguas de la interfaz del componente reducen las oportunidades de malas interpretaciones entre componentes que suministran un servicio y los que podrían utilizar ese servicio.

Respondiendo a la sugerencia de profesionales del área de fisioterapia del Centro Integral de Rehabilitación de Colombia (CIREC) y a las propuestas de trabajos futuros realizadas por en (Ortiz, y otros, 2009) se propuso un simulador

basado *XNA* el cual se conforma por un dispositivo protésico virtual con apariencia humana y con la posibilidad de adaptarle un video juego controlado a partir de las señales ME generadas por el paciente. En este punto es importante resaltar que existe una gran cantidad de video juegos desarrollados para *XNA* que son de libre distribución, incluso de código abierto, los cuales podrían ser fácilmente controlados a partir de las señales ME a través del puerto serial, es decir, en este caso el dispositivo protésico virtual sería reemplazado por el control de un personaje o un vehículo de acuerdo con el tema del video juego.

El haber implementado las interfaces del simulador virtual propuesto directamente en un microcontrolador permite acercarse al comportamiento que se obtendría si se dispusiese del dispositivo protésico real y de esta manera realizar una evaluación más asertiva de las posibles y variadas estrategias que se implementan en dispositivos de esta índole. Así, al contar con el dispositivo protésico real, tanto para los desarrolladores como para el paciente la migración del dispositivo virtual al real contendrá menos traumatismo y se habrá logrado el objetivo del simulador el cual es entrenar a los pacientes para el control del dispositivo protésico.

Generar el modelo de análisis haciendo uso del Lenguaje Unificado de Modelado UML, permite abstraer en un solo plano de una manera detallada los componentes que integran una prótesis robótica de mano. Luego de una previa socialización del modelo de análisis con alguno de los integrantes del Grupo de Investigación en Automática Industrial se evidenció el ágil entendimiento de los subsistemas que integran un desarrollo de este tipo. Tener en un mismo plano gráfico interrelacionadas la mayoría de soluciones en el área de prótesis de mano robótica, permite a los integrantes del proyecto realizar aportes en miras al mejoramiento de la solución propuesta.

5.2 Aportes del proyecto

Este proyecto se enfocó en lograr una solución asequible a poblaciones de escasos recursos. La implementación del simulador de prótesis de mano robótica incluyendo el dispositivo protésico virtual se hizo utilizando software completamente

gratuito y con su respectiva licencia la cual está al alcance y disposición de cualquier persona. Igualmente se escogió un microcontrolador de 32bits de altas prestaciones que a la fecha no supera un costo de US\$ 3.5.

Desde la perspectiva tecnológica, se realizó un aporte en el campo de las arquitecturas de integración de componentes hardware/software al proponer un conjunto normalizado de interfaces para los componentes de una prótesis de mano y especificar a través del presente documento los principales criterios para su diseño.

Bajo el marco de desarrollo de la asignatura “Laboratorio IV” correspondiente al semestre VIII de la carrera de Ingeniería en Electrónica y Telecomunicaciones de la Universidad del Cauca, como docente encargado de dicha asignatura se desarrollaron los siguientes proyectos:

Exploración, modelado e implementación de una prótesis robótica de mano.

Adquisición de señales electromiográficas y detección de los movimientos flexión y extensión.

Los resultados de los proyectos se encuentran disponibles en la Web en (Cabrera, y otros, 2008) y (Rojas, y otros, 2008) respectivamente.

Esta tesis de maestría, partiendo de la arquitectura definida en (Grupo de Investigación en Automática Industrial, 2008), realizó una adecuada definición de las interfaces, especificando las responsabilidades de cada componente, el tipo y formato de datos en que se transmiten los mensajes y las restricciones temporales para su transmisión y respuesta, de forma que facilita su distribución y reuso.

El conjunto de interfaces para los componentes de una prótesis robótica permitirá en esta área desarrollos sostenibles y comprensibles, facilitando el análisis, integración y depuración de los componentes hardware/software, realizando un significativo aporte a la manera de realizar documentación de proyectos relacionados con electrónica y fomentando la adopción de buenas prácticas de análisis y diseño de sistemas electrónicos modulares.

5.3 Trabajos Futuros

La continua evolución científica hace que las posibilidades para conformar un dispositivo protésico sean prácticamente ilimitadas. Por lo tanto, el conjunto normalizado de interfaces debe evolucionar conforme la ciencia genere nuevas propuestas.

Al igual que en el proyecto (Seniam, 2007) tiene como objetivo realizar una integración básica y aplicada de las investigaciones acerca de señales EMG superficiales a nivel Europeo se propone la posibilidad de crear un grupo a nivel nacional encargado de centralizar los desarrollos y estandarizar las interfaces de los componentes que conforman a los dispositivos protésicos.

Dado que el presente trabajo se limitó al desarrollo de dispositivos protésicos para pacientes amputados de mano se sugiere ampliar el conjunto normalizado de interfaces a cualquier otra prótesis, tales como, prótesis de pierna o brazo.

Referente al desarrollo del entrenador virtual se deja abierta la posibilidad de acondicionar un video juego con el fin de estimular en el paciente la generación de señales mioeléctricas y lograr mejores resultados en el proceso de adaptación al dispositivo protésico real.

A partir del conjunto de interfaces propuesto facilita la posibilidad de experimentar diversas estrategias de implementación para cada uno de los módulos que conforma al dispositivo protésico, por ejemplo, las interfaces propuestas fueron concebidas en miras a la implementación de un simulador para dispositivos protésicos de mano, sin embargo, se podrían verificar en la implementación de prótesis reales.

Particularmente al interior de la Universidad del Cauca al aprovechar las ventajas propuestas por la ingeniería de software se permite que en futuros proyectos de gran magnitud donde su desarrollo se plantee de manera modular, por componentes, se cuente con un referente que permita la organización y estructuración del trabajo por medio de un acuerdo formal en el establecimiento de responsabilidades, garantizando su comprensión, evolución y mantenimiento.

Referencias Bibliográficas

Aguilar Eliana Modelo y control de posición y de fuerza de una prótesis robótica de mano [Informe] : Examen de valoración / Universidad del Cauca. - Popayán : [s.n.], 2008.

Alonso A [y otros] Myoelectric prostheses trainer for hand and arm amputees [Libro] / ed. Medicina Mapfre. - 2002. - Vol. 3 : 1.

Analog Device Low cost, low power: Instrumentation amplifier. [En línea]. - 1999. - Noviembre de 2008. - Disponible en: <http://users.ece.utexas.edu/~valvano/Datasheets/AD620.pdf>.

Azor J. R. La transformada de gabor. [En línea]. - Noviembre de 2008. - Disponible en: <http://www.um.edu.ar/math/wavelets/gabor.htm>.

B&Engineering Active surface emg electrode [En línea]. - 2009. - Noviembre de 2009. - Disponible en : <http://www.bleng.com/electrod.htm>.

Betancourt F. J. y Giraldo E Reconocimiento de patrones de movimiento [Publicación periódica] // Scientia et Técnica. - 2004. - Vol. 26. - págs. 53-58.

Bing S y Ikeuchi K A framework for Recognizing Grasps [Libro]. - Pennsylvania : [s.n.], 1991.

Bionics T The i-limb hand [En línea]. - Noviembre de 2008. - Disponible en: <http://www.touchbionics.com>.

Birkedal L [y otros] Pattern recognition of upper-body electromyography for control of lower-limb prostheses [En línea] // Sitio Web del Departamento de Ciencias de la Salud y la Tecnología de la Universidad de Aalborg. - Noviembre de 2008. - Disponible en : <http://www.miba.auc.dk/>.

Birkedal L [y otros] Patern Rocognition of upper-body electromyography for control of lower limb prostheses [Informe] / Institute of Electronic Systems ; Aalborg University. - 2002.

Brittain H [y otros] Sensory-feedback system compatible with myoelectric control, [Publicación periódica] // Med. & Biol.Eng. & Comput. - 1980. - Vol. 18. - págs. 65 - 69.

Bueche Frederick J Física para estudiantes de ciencias e ingeniería [Libro]. - Mexico : Mc Graw Hill, 1998. - 3.

Burr Brown www.datasheetcatalog.com [En línea] // INA2128. - 2009. - 12 de 10 de 2009. - Disponible en <http://www.datasheetcatalog.com/>.

Cabrera Juan [y otros] Adquisición de señales electromiográficas y detección [Informe]. - Popayán : Universidad del Cauca, 2008.

Cáceres Juan Pablo [En línea]. - Agosto de 2007. - Agosto de 2009. - Disponible en: http://ccrma.stanford.edu/workshops/cm2007/topics/clases/PDFs/05fourier_handout.pdf.

Care, M. H. Biomedical electrodes. [En línea]. - 2009. - Noviembre de 2009. - Disponible en : <http://www.3m.com/>.

Center for Innovation in Product Development Center for innovation in product development, Designin with D.C [En línea]. - 1999. - Disponible en : <http://lancet.mit.edu/motors/>.

ConMed Adul ecg electrodes. [En línea]. - 2009. - Noviembre de 2009. - Disponible en: http://www.conmed.com/products_ECGelectrod_Back1.php.

Cortez J. A., Medina F. A. y Chaves J. A. Del análisis de fourier a las wavelet : Análisis de fourier [Publicación periódica] // Scientia Et Technica. - 2007. - 34 : Vol. 13. - págs. 151-156.

Dario P Multiple Sensing for Dexterous End Effectors [Informe] / Scuola Superiore S. Anna and Centro "E. Piaggio" ; University of Pisa. - 1988. - págs. 02/03-25/03.

De Luca , J.C The use of Surface Electromyography in Biomechanics [Conferencia] // Journal of Applied Biomechanics. - págs. 135-163.

De Luca Carlo J. Encyclopedia of Medical Devices and Instrumentation [Libro]. - Boston : Wiley, 2006. - 2 : Vol. 6 : págs. 98-109.

De Luca Carlo J. Fundamental Concepts in EMG Signal Acquisition [En línea] // Delsys. - 2001. - 2009. - <http://www.delsys.com/library/tutorials.htm>.

De Luca Carlo J. Sensor Related Factors and Fidelity of the sEMG Signal [En línea]. - 2007. - Noviembre de 2008. - Disponible en : http://nmrc.bu.edu/tutorials/sensor_facts.pdf.

De Luca Carlo J. The use of Surface Electromyography in Biomechanics [Informe]. - Boston : Delsys, 1997. - págs. 135-163.

Delsys EMG Signal Analysis [En línea]. - 2009. - [http://www.delsys.com/Attachments_pdf/TN10 - EMG Signal Analysis.pdf](http://www.delsys.com/Attachments_pdf/TN10-EMG_Signal_Analysis.pdf).

Delsys Surface emg sensor [En línea]. - 2009. - Noviembre de 2009. - Disponible en: <http://www.delsys.com/>.

Delsys The use of surface electromyography in biomechanics [En línea]. - 2009. - Noviembre de 2009. - Disponible en: www.delsys.com.

Device A Low cost, low power: Instrumentation amplifier. [En línea]. - 1999. - Noviembre de 2008. - Disponible en: <http://users.ece.utexas.edu/~valvano/Datasheets/AD620.pdf>.

Devices Analog Datasheet catalog [En línea] // AD620. - 2009. - 12 de 10 de 2009. - Disponible en <http://www.datasheetcatalog.com/>.

Duarte A [y otros] Aplicación del efecto Hall a la medición de la posición en manos artificiales [Conferencia] // Jornadas de Automática. - 2000.

Duarte A [y otros] Sensores de contacto, fuerza y deslizamiento para manos protésicas [Conferencia] // Iberdiscap. - Madrid : [s.n.], 2000.

Dumitru Daniel Electrodiagnostic Medicine [Libro]. - New York : Hanley & Belfus, 2002. - Vol. 2.

Englehart K [y otros] Classification of the myoelectric signal using time-frequency based representations [Publicación periódica] // Special Issue of Medical Engineering and Physics on Intelligent Data Analysis in Electromyography and Electroneurography. - 1999.

Farina Dario, Merletti Roberto y Enoka Roger Journal of Applied physiology [Conferencia] // The extraction of neural strategies from the surface EMG. - Torino : [s.n.], 2004. - Vol. 96.

Farrell T. R. y Weir R. F. The optimal controller delay for myoelectric prostheses [Publicación periódica] // Neural Systems and Rehabilitation Engineering, IEEE. - 2007. - 1 : Vol. 15.

Ferguson y Dunlop Grasp recognition from myoelectric signals [En línea] // Australasian Conference on Robotics and Automation. - 2002. - Disponible en: <http://www.araa.asn.au/acra/acra2002/Papers/Ferguson-Dunlop.pdf>.

Ferguson, Simon y Dunlop Reg. Grasp Recognition From Myoelectric Signals [Conferencia] // Australasian Conference on Robotics and Automation. - Auckland : [s.n.], 2002. - págs. 27-29.

Franco J. F., Giraldo E y Orozco A Reconocimiento de patrones de movimiento del sistema fisiológico brazo-antebrazo, a partir de señales electromiográficas [Informe] / Universidad Tecnológica de Pereira. - Pereira : [s.n.], 2004.

Ganssle Jack Embedded System: World Class Desings [Libro]. - Burlington : newnespress, 2007. - Vol. 1.

Ganssle Jack The Ganssle Group [En línea] // 80% of all embedded systems are delivered late. - Enero de 2004. - 13 de Agosto de 2009. - [Disponible en: <http://www.ganssle.com/fsm.pdf>].

Gaviria C, Diaz J y Mosquera V Agarre estable de objetos con una prótesis de mano robótica: Informe final [Informe] : Reporte de resultados de investigación, Proyecto Prótesis Robótica de Mano / Universidad del Cauca. - Popayán : [s.n.], 2008.

Gazitúa Ricardo Manual de Semiología [En línea]. - 2004. - Agosto de 2009. - Disponible en: <http://escuela.med.puc.cl/Publ/ManualSemiologia/Indice.html>.

Gomez J.I., Mendoza J.A. y Salazar A Biorealimentación en el análisis y rehabilitación muscular [Publicación periódica] // Scientia Et Technica. - 2007.

Graupe D EMG pattern analysis for patient responsive control of FES in paraplegics for walker-supported walking. [Publicación periódica] // Fellow, IEEE trans. on biomedical engineering. - 1989. - 7 : Vol. 36.

Graupe D y Cline W Functional separation of EMG signals via ARMA identification methods for prosthesis control purposes [Publicación periódica] // IEEE Trans. Systems Man and Cybernetics. - 1975. - 252259 : Vol. 5.

Grupo de Investigación en Automática Industrial Prótesis robótica para amputados de mano, Convocatoria de proyectos de investigación número 452/08 de Colciencias [Informe]. - Bogota : Universidad del Cauca, 2008.

Hanafusa H y Asadas H A robotic hand with elastic fingers and its application to assembly process [Conferencia] // IFAC Symposium on Information and Control Problems in Manufacturing Technology. - Tokyo : [s.n.], 1977.

Hannaford B y Lehman S Short Time Fourier Analysis of the Electromyogram: Fast Movements and Constant Contraction. [Publicación periódica] // IEEE Transactions on Biomedical Engineering. - Diciembre de 1986. - Vols. BME-33. - págs. 1173-1181.

Hargrove L, Englehart K y Hudgins B A comparison of surface and intramuscular myoelectric signal classification [Conferencia]. - 2005. - págs. 5009-5012.

Hauschild M, Davoodi R y Loeb G. E. A virtual reality environment for designing and fitting neural prosthetic limb [En línea] // IEEE Transactions on Neural Systems and Rehabilitation Engineering. - 2007. - Disponible en : <http://bme.usc.edu/assets/005/57578.pdf>.

Hauschild M, Davoodi R y Loeb G.E. A Virtual Reality Environment for Designing and Fitting Neural Prosthetic Limbs [En línea] // IEEE Transactions on Neural Systems and Rehabilitation Engineering. - 2007. - Noviembre de 2008. - Disponible en: <http://bme.usc.edu/assets/005/57578.pdf>.

Huaroto Luis Julio Cirugía ortopédica y traumatología [En línea] // Semiología de antebrazo, muñeca y mano. - Agosto de 2009. - Disponible en: http://sisbib.unmsm.edu.pe/BibVirtual/libros/Medicina/cirugia/Tomo_II/semiologia_amm.htm.

Hugh D University Physics [Libro]. - 1992.

Iberall T [y otros] On the development of emg control for a prosthetic hand [Informe] / Departamento de ingeniería biomédica ; University of Southern California. - 2008.

Informática Asería Asesoría Informática [En línea] // Optimizando la tecnología de la información. - Julio de 2009. - 13 de Agosto de 2009. - Disponible en: http://www.asesoriainformatica.com/definiciones_i.htm.

Jockusch J, Walter J y Ritter H A Tactile Sensor System for a Three-Fingered Robot Manipulation [Conferencia] // Proc. Int. Conf. on Robotics and Automation (ICRA) IEEE. - 1997.

Konrad P The ABC of EMG: A Practical Introduction to Kinesiological Electromyography [En línea]. - 2005. - Septiembre de 2008. - Disponible en : <http://www.noraxon.com/>.

Kyberd P.J. MARCUS: A Two Degree of Freedom Hand Prosthesis with Hierarchical Grip Control [Publicación periódica] // IEEE Transaction on Rehabilitation Engineering. - 1995. - 1 : Vol. 3. - págs. 70-76.

La transformada de wavelet. [En línea] // Personal Internet site, Facultad de Ingeniería de la Universidad de Mendoza.. - Noviembre de 2008. - <http://www.um.edu.ar/math/wavelets/index.html>.

Mackenzie C.L. y Iberall T The grasping hand [Libro]. - 1994.

Mason M y Salisbury J.K Robot Hands and the Mechanics of Manipulation [Sección del libro] / aut. libro 1985. - Cambridge : [s.n.].

Mexico. D. M. de La prótesis i-limb, de cinco dígitos mecánicos se lleva el premio más importante de ingeniería en gb [En línea]. - Julio de 2008. - Disponible en: <http://www.medicinadigital.com>.

Montilva J, Arapé N y Colmenares J Desarrollo de software basado en componentes [Conferencia] // Actas del IV. Congreso de Automatización y Control. - 2003.

Mooney Jim Software Interface Standards [En línea]. - Dept. of Statistics and Computer Science. - Agosto de 2009. - Disponible en : <http://www.csee.wvu.edu/~jdm/research/portability/standards/stdlist.html>.

Muñoz Burbano Edgar, Paruma Oscar H. y Florez Juan Fernando Congreso de Tecnologías Aplicadas a la Enseñanza de la Electrónica [Conferencia] // APLICACIONES DE LAS SEÑALES MIOELECTRICAS PARA EL CONTROL DE INTERFACES HOMBRE-MAQUINA. - Valencia : [s.n.], 2004.

O. B. H. Care Prótesis mioeléctricas de brazo [En línea]. - Noviembre de 2008. - <http://www.ottobock.es>.

Oddson Lards [y otros] Journal of Rehabilitation Research and Development [Conferencia] // Development of new protocols and analysis procedures for the assessment of LBP by surface EMG techniques.. - Boston : [s.n.], 1997. - Vol. 34. - 4.

Ortiz Julian y Vallejos David ENTORNO GRÁFICO DE UN ENTRENADOR VIRTUAL DE PRÓTESIS DE MANO [Informe]. - Popayán : Universidad del Cauca, 2009.

Rodriguez L.E. Capitulo IV [Informe]. - Barcelona : [s.n.], 2005.

Rodriguez L.E. y Casals A Sensing and control of a prosthetic hand with myoelectric feedback [Publicación periódica] // IEEE Conference Biomedical Robotic And Biomechatronic. - 2006. - págs. 607 - 612.

Rohen J.W., Yokochi C y Lütjen-Drecoll E Atlas de Anatomía Humana. Estudio Fotográfico del Cuerpo Humano. [Libro]. - Madrid : Elsevier España S.A., 2003. - 5.

- Rojas Luis, Suarez Luis y Cabrera Juan** Exploración, modelado e implementación de una prótesis robótica de mano [Informe]. - Popayán : Universidad del Cauca, 2008.
- Romo H, Realpe J y Jojoa P** Aplicación de señales emg superficiales y su aplicación en control de prótesis de mano [Libro]. - 2007. - Vol. 4 : págs. 127-136.
- Roy Serge [y otros]** Journal of Rehabilitation Reserach and Development [Conferencia] // Clasification of back muscle impairment based on the surface electromyographic signal. - Boston : [s.n.], 1997. - Vol. 34. - 4.
- Salazar A y Ramírez J. C.** Diseño e implementación de un sistema de control que utilice señales electromiográficas para activar una prótesis mioeléctrica. - 2006.
- Sánchez C. M., Loaiza J. L. y Cortés C. J.** Diseño y construcción de prototipo de pinza y rotador (funciones de pronación y supinación) para prótesis mioeléctrica de mano [Publicación periódica] // Ingeniería e Investigación. - Septiembre de 2006. - 3 : Vol. 26.
- Sebelius F, Erikson L y Balkenius C** Neural control of a virtual prosthesis [Libro]. - 1998.
- Seniam** Surface ElectroMyoGraphy for the Non-Invasive Assessment of Muscles [En línea]. - 2009. - <http://www.seniam.org/>.
- Soares A [y otros]** The Development of a Virtual Myoelectric Prosthesis Controlled by an EMG Pattern Recognition System Based on Neural Networks [Conferencia] // Journal of Intelligent Information Systems. - 2003. - Vol. 21. - págs. 127 - 141.
- Speeter Thomas H.** A Tactile Sensing System for Robotic Manipulation, The International Journal of Robotics Research [Conferencia]. - 1990. - Vol. 9. - págs. 25-36. - 6.

Stegeman D y Hermens H Standards for surface electromyography: the european project "surface emg for non-invasive assessment of muscles (seniam). [En línea]. - 2007. - Noviembre de 2009. - Disponible en: <http://www.med.uni-jena.de/motorik/pdf/stegeman.pdf>.

TabascoHoy.com Desarrollan en IPN brazo robot para personas amputadas [En línea]// Tabasco HOy.com. - Agosto de 2009. - Disponible en: http://www.tabascohoy.com.mx/nota.php?id_nota=124000.

The use of surface electromyography in biomechanics. [En línea]. - Noviembre de 2008. - Disponible en : www.delsys.com.

The use of surface electromyography in biomechanics. [En línea]. - Noviembre de 2008. - Disponible en : www.delsys.com.

Todd R. Farrell Richard F. Weir The Optimal Controller Delay for Myoelectric Prostheses [Publicación periódica]// IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING. - [s.l.] : IEEE, 2007. - 1 : Vol. 15. - págs. 111-118.

Zecca M [y otros] Control of Multifunctional Prosthetic Hands by Processing the Electromyographic Signal [Publicación periódica] // Critical Review in Biomedical Engineering. - 2002. - Vol. 30. - págs. 459-485.

Zecca M On the development of a cybernetic prosthetic hand [Libro]. - Pisa : Scuola Superiore Sant'Anna, 2003.

Anexo 1

A.1 Control del par del motor D.C.

El control de movimiento de un motor se realiza a través del *par* del motor o también conocido como *momento* (Center for Innovation in Product Development, 1999) De acuerdo con (Hugh, 1992) El *par* es “Una fuerza que produce o tiende a producir rotación o torsión”.

De acuerdo con la primera ley de Newton: "Todo objeto en reposo o moviéndose con la misma rapidez permanece en ese mismo estado, siempre que no se le aplique una fuerza resultante no equilibrada." el torque del motor y el torque de la carga deberán estar balanceadas para que el motor pueda girar. Si el torque de la carga es mayor a la suministrada por el motor entonces este último no gira o lo hace en sentido contrario empujado por la carga.

Por ejemplo, si el dispositivo prostético de mano tiene sujetado un objeto de masa M , la cual ejerce un mayor par en sentido opuesto al que en ese instante suministra el motor de la articulación, entonces, probablemente la mano se venza y el objeto se caiga.

Ahora bien, si el objeto de masa M no ejerce una fuerza mayor a la que suministra el motor, la articulación pudiese incluso mantenerse en una posición fija si se le suministra el valor adecuado de corriente, o con su efecto análogo, usando PWM una frecuencia y ciclo útil dado.

Para que el sistema se encuentre en equilibrio se debe compensar el par ejercido por la carga con el par suministrado por el motor. La forma de controlar el par del motor DC es a través del suministro de corriente a las bobinas del rotor, con lo que a su vez se controla su velocidad. Particularmente, si el control del motor es a

través de una señal PWM, entonces es el ciclo útil de la señal el que se debe modificar, lo cual haría algo equivalente al control de suministro de corriente.

Es importante notar que la corriente máxima que soporta el motor hace que éste gire a su velocidad tope, que es cuando entrega el mayor par a la carga. Por lo tanto, un motor DC no puede mantener baja las RPM o revoluciones por minuto y a su vez entregar un par máximo o elevado a la carga como si girara a su máxima velocidad.

En conclusión, el módulo de control se asemeja a un sistema de "PWM inteligente", el cual varía automáticamente la velocidad del motor para modificar el par sobre la carga cuando esta varíe por algún motivo.

Por otro lado, para incrementar el par suministrado por un motor se acondiciona un sistema mecánico basado en engranajes conocidas como *caja de conversión mecánica*, en el cual, el par de salida es una relación entre el par de entrada generado por el motor y la relación de radio y número de dientes de cada uno de los engranajes que conforman el sistema de transmisión (Center for Innovation in Product Development, 1999).

La caja de conversión mecánica funciona como un sistema de amplificación y transmisión de fuerza. En este sistema, el motor se desempeña holgadamente transfiriendo su máximo par a la caja, la cual reduce las RPM y aumenta el par entregado a la carga que se desea mover, (Bueche, 1998).

Otra ventaja de contar con un sistema de engranajes es el incremento de la resistencia a la inercia. Por lo tanto, aun en ausencia de energía eléctrica de polarización de los motores, el sistema se detiene y conserva la posición de los dedos.

Para el control de par de un motor en función de la carga, potencia y velocidad angular se considera la siguiente ecuación:

$$T = \frac{p}{w}$$

Donde:

p = Potencia entregada a la carga.

T = Par o Torque.

w = Velocidad angular.

Para que el dispositivo protésico de mano ejerza una aprehensión estable sobre un objeto, el módulo de control de fuerza debe mantener el par suministrado

en cada una de las articulaciones de manera estable. Para ello, De acuerdo con la anterior ecuación, T (par), depende directamente proporcional a los valores de p (potencia) e inversamente a los de w (velocidad angular). Por ejemplo, si se incrementa la carga, el motor disminuye su giro y por lo tanto, el sistema de control deberá aumentar la potencia suministrada para así mantener constante el valor de T .

Por otro lado, si el dispositivo prostético se encuentra dotado de un conjunto de sensores de medición de fuerza, entonces, para controlar la fuerza ejercida sobre un objeto empuñado, tan solo es necesario corregir la potencia suministrada a los motores de cada articulación.

En síntesis, para ejercer control ya sea de posición o de fuerza de la prótesis es necesario corregir el valor del par suministrado a los actuadores con el fin de disminuir el error entre las consignas deseadas y los valores reales adquiridos a través del sistema de realimentación sensorial.

Anexo 2

A.2 Tipo de Datos en Sistemas Embebidos.

Si la resolución del sistema mecánico es de un grado por paso y teniendo en cuenta que el ángulo de rotación de las articulaciones de la mano humana en el casos más extremo es de 120° (Huaroto), el tipo de dato adecuado para las variables de control es el entero de 8 bits, el cual permite fijar valores entre 0 y 255. Sin embargo, este tipo de dato limita a los dispositivos prostéticos que mecánicamente permiten alcanzar una mejor resolución, es decir, menor a un grado por paso. Lo anterior conlleva a considerar el uso de variables de punto fijo o punto flotante las cuales permiten fijar valores fraccionales.

Es importante tener en cuenta que los procesadores digitales hardware pueden solamente almacenar y procesar *bits*, es decir, todos los números deben ser representados como una colección de bits. Cada bit representado ya sea por un *cero* o un *uno*, de aquí que el sistema numérico naturalmente usado en los sistemas embebido sea el sistema binario.

Dado que el dispositivo prostético es finalmente un dispositivo hardware, el cual es gobernado por un algoritmo de control empotrado en un sistema de recursos computacionales limitados, se recomienda que el tipo de dato a manejar sea de punto fijo.

El número de punto fijo es esencialmente un número entero escalado por un cierto factor. Por ejemplo, el número real 1.23 puede ser representado por $123/100$, donde el factor de escala es 100. Consecuentemente las operaciones matemáticas entre dos números de punto fijo son procesadas de manera más rápida, el cual es un gran beneficio para la construcción de un sistema prostético con respuesta en tiempo real.

Respecto al tipo de dato punto flotante se propone el uso del estándar IEEE 754 el cual se desarrolló para facilitar la portabilidad de los programas de un procesador a otro y alentar el desarrollo de programas numéricos sofisticados. Este estándar ha sido ampliamente adoptado y se utiliza prácticamente en todos los procesadores y coprocesadores aritméticos actuales. El estándar del IEEE define el formato para precisión simple de 32 bits y para precisión doble de 64 bits.

En la actualidad existe una gran tendencia a la fabricación de sistemas embebidos con procesadores de 32 bits, sin embargo, es necesario que además cuenten con unidad de punto flotante. Esto con el fin de que las operaciones aritméticas entre datos de este tipo se realicen directamente, sin necesidad de adecuar los datos a una unidad aritmética lógica convencional, lo cual significaría que por cada operación aritmética a realizar se ejecutarían líneas de código adicionales que retardarían la obtención del resultado y consecuentemente la respuesta del sistema.

De acuerdo al procesador a utilizar en la implementación del dispositivo prostético, el desarrollador determinará el tipo de dato para el manejo de números fraccionales. Si el procesador cuenta con una unidad de punto flotante se recomienda que el tipo de dato sea de punto flotante. Por otro lado, si el procesador no cuenta con esta unidad, el tipo de dato adecuado es el punto fijo, el cual, respecto al dato de punto flotante proporcionan un buen grado de precisión. Ver anexo, "Aritmética de punto fijo".

Anexo 3

A.3 Lenguaje de Definición de Interfaces IDL

Primero que todo, es importante especificar que IDL no es un lenguaje de programación, es un lenguaje de descripción y por lo tanto, no es posible implementar funciones o rutinas utilizando su nomenclatura.

Para realizar una especificación por medio de IDL es necesario conocer principalmente los conceptos de módulo, interface y operación:

Un módulo permite agrupar varias definiciones de interfaces que comparten un mismo propósito. De una manera más específica, un módulo es un espacio de nombre que actúa como un conector para las interfaces y declaraciones asociadas. La declaración de un modulo se presenta a continuación:

```
Module NombreModulo
{
    //líneas para definición de interfaces y atributos.
};
```

Una interfaz describe el conjunto de operaciones disponibles de un componente, pero no las implementa. Una interfaz se define dentro de un módulo como se presenta a continuación:

```
Module nombreModulo
{
    Interface nombreInterfaz1
    {
        //líneas para definición de operaciones
    }
    Interface nombreInterfaz2
    {
        //líneas para definición de operaciones
    }
};
```

Una operación es una entidad que denota un servicio que se encuentra disponible, la cual podría tomar parámetros, devolver un resultado, o lanzar excepciones. Por lo tanto, requiere un tipo de retorno, un identificador o nombre de la operación y la definición de parámetros. En la definición de parámetros se especifica el tipo de identificador y el modo de direccionamiento: *in*, *out* e *inout*.

Para comprender el modo de direccionamiento de un parámetro se hará una analogía con el lenguaje C. El modo de direccionamiento equivale a la manera en que se pasan los argumentos; si el paso es por valor el modo de direccionamiento en IDL es “in” o si es por referencia o dirección el modo es “*inout*” y “*out*” si el valor no es inicializado.

La definición de una operación se especifica a continuación.

```
Module nombreModulo
{
    Interface nombreInterfaz1
    {
        <tipoRetorno>nombreOperacion( <arg1>, <arg2>, ... );
    }
}
```

Anexo 4

A.4 Características Espectrales.

A.4.1 Transformada de Fourier

Particularmente, la Transformada de Fourier en su forma discreta calcula el espectro de potencia de una señal $x[n]$ tal como se presenta a continuación:

$$X(k) = \sum_{n=1}^N x[n] e^{-j2\pi(k-1)\frac{n-1}{N}}, k = 1, \dots, N \quad (\text{A.4.1})$$

Donde $X(k)$ es el espectro obtenido a partir de la señal $x[n]$. En el actual proyecto, $x[n]$ es un arreglo o conjunto de muestras adquiridas de la señal mioeléctrica, este conjunto de muestras se obtiene como resultado del *módulo de digitalización y filtraje digital*. Para estimar la frecuencia correspondiente a cada componente $X(k)$, es decir f_k , se realiza a partir de:

$$f_k = \frac{k f_s}{N} \quad (\text{A.4.2})$$

Donde f_s es la frecuencia de muestreo.

Para optimizar el tiempo de procesamiento de la transformada discreta de Fourier se recomienda hacer uso del algoritmo *FFT*, del inglés *Fast Fourier Transform*, el cual reduce considerablemente el número de operaciones necesarias para obtener $X(k)$.

La Transformada de Fourier actúa como un detector de aporte de energía de cada frecuencia que compone la señal. Por lo tanto es posible calcular el valor de potencia como la cantidad de energía por unidad de tiempo.

Para el cálculo de energía y potencia de una señal, diversos autores como (Cáceres, 2007) hacen uso de la transformada de Fourier de Tiempo Corto o *STFT*, del inglés Short Time Fourier Transform, también conocida como la Transformada de Fourier con ventana deslizante o Transformada Gabor debido al nombre del autor que la propuso.

El uso de la *STFT* se debe a que la Transformada Discreta de Fourier clásica requiere que la señal sea estacionaria, característica que no cumple en la señal mioeléctrica. La *STFT* aunque asume estacionariedad, lo hace en cortos periodos de tiempo a través de una ventana deslizante en el tiempo (Hannaford, y otros, 1986). Por lo tanto la *STFT* considera las variaciones de frecuencia a lo largo del tiempo.

La *STFT* muestra la información de frecuencia contenida en la señal y su variación en el dominio del tiempo. El procedimiento usado es llamado *Windowing*, del inglés, ventaneando y consiste en aplicar la transformada de Fourier clásica a segmentos cortos de la señal, es decir que cada segmento de la señal se considera estacionaria.

La *STFT* puede ser expresada usando una ventana más corta que la longitud total de la señal y puede ser posicionada arbitrariamente a lo largo de la secuencia de tiempo, así:

$$S(f, t) = \sum_{n=0}^{N-1} x[n]w[n-t]e^{-j2\pi ft} \quad (\text{A.4.3})$$

Donde $x[n]$ es la señal y $w[n]$ es la ventana.

Otra metodología aplicada para extraer características de la señal EMG, es hacer uso de los Cepstrales del cual se pueden obtener una serie de coeficientes.

A.4.2 Coeficientes Cepstrales

El Cepstral de una señal está definido como la transformada de Fourier Inversa del logaritmo de la magnitud del espectro de potencia. Al manipularse matemáticamente la información, se obtiene valores característicos llamados coeficiente cepstrales los cuales representan las propiedades de la señal.

A.4.3 Transformada Wavelets

Por otro lado, una interesante alternativa de análisis de señales no estacionarios, como las mioeléctricas, es la Transformada Wavelet; la cual es una representación de la señal en tiempo-frecuencia obtenida a partir del uso de filtros. A su vez está relacionada con el análisis de los armónicos de la señal.

A diferencia de la Transformada de Fourier, en el análisis con Wavelets no se asume que la señal analizada sea periódica; por lo tanto, es posible considerar señales que exhiben cambios bruscos e incluso discontinuidades utilizando muchas menos funciones Wavelets base que las que se necesitarían si se utilizara funciones seno y coseno para alcanzar una aproximación adecuada a partir de la Transformada de Fourier.

Con la Transformada de Fourier de tiempo corto (STFT) se logra una mejor localización de la aparición de una singularidad en una señal. Pero solo se conocerá en qué intervalo de tiempo se produce la singularidad debido a que la localización depende del ancho elegido para la función ventana. Además, dichos eventos no podrán ser resueltos si aparecen muy cerca unos de otros ya que no será posible distinguir diferentes comportamientos dentro de una misma amplitud de ventana, tal como aparece en (Birkedal, y otros).

Por otro lado, la Transformada Wavelet permite resolver este tipo de problemas dado que es capaz de concentrarse en fenómenos transitorios y de alta frecuencia mejor que la STFT. En la Transformada de Fourier de tiempo corto, una vez que el tamaño de la ventana es elegido todas las frecuencias son analizadas con las

mismas resoluciones de tiempo y frecuencia, distinto de lo que sucede en la Transformada Wavelet que tiene un tamaño de ventana adaptado a las frecuencias.

Wavelets, significa ondoletas o ondas pequeñas la cuales se utilizan para representar otras funciones tal y como se hace con las funciones seno y coseno de la Transformada de Fourier. Cada una de estas ondas estará representada por un coeficiente el cual será utilizado para caracterizar la señal. Dichos coeficientes son los que se utilizan para caracterizar la señal ME, tal como se presenta a continuación.

A.4.4 Coeficiente Wavelets

A nivel práctico y computacional se prefiere la Transformada Wavelet Discreta. Cuando se usan Wavelets discretas para transformar una señal continua, el resultado es una serie de coeficientes que se conocen como la descomposición en series wavelet.

El desarrollo de algoritmos rápidos para evaluar la Transformada Wavelet Discreta llevó a implementar los llamados bancos de filtros, el cual es un conjunto de filtros pasabanda FIR que separa la señal de entrada en varios componentes, cada uno compuesto de una subbanda de frecuencia.

La Transformada Wavelet Discreta (DWT) de una señal “x” se calcula operando la señal con una serie de filtros; inicialmente con un filtro pasa bajas, caracterizado con la respuesta al impulso “g”, que remueve las frecuencias por encima de la mitad de la frecuencia máxima. La operación entre “x” y “g” se realiza a través de la convolución y contiene los denominados *coeficientes aproximados* de la Transformada Wavelet:

$$y[n] = (x * g)[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot g[n - k] \quad (\text{A.4.4})$$

Simultáneamente la señal “x” se opera con un filtro pasa altas caracterizado con la respuesta al impulso “h”, de esta operación se obtiene los denominados coeficientes detallados de la Transformada Wavelet. El proceso anterior de operar la señal “x” con el filtro pasa bajas “g” y el pasa altas “h” con el fin de obtener respectivamente los coeficientes aproximados y detallados se muestra gráficamente como:

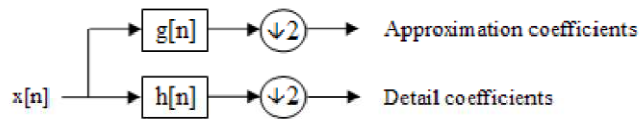


Figura A.4.4. Obtención de los coeficientes wavelet

Este proceso de filtrado puede iterarse y así obtener una descomposición a diferentes niveles. Basta hacer pasar los *coeficientes de aproximación* obtenidos del proceso de filtraje anterior por un par idéntico de filtros y así obtener los coeficientes del siguiente nivel. Gráficamente se representa como:

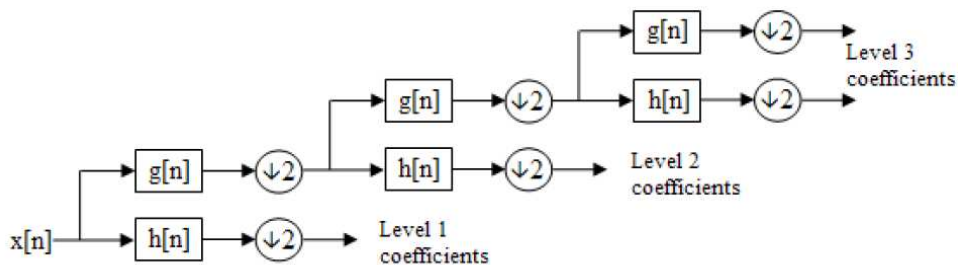


Figura A.4.1. Proceso de filtrado

Los filtros pasa bajas y pasa altas están relacionados entre sí como un filtro espejo quadrature, el cual divide la señal de entrada en dos bandas que a su vez submuestrean por dos. Estos filtros de media banda forman bases ortogonales, es decir que, la información capturada por el filtro pasa bajas es completamente independiente de la información capturada por el filtro pasa altas, son mutuamente excluyentes; de la misma manera permite que se dé la reconstrucción de la señal a partir de la transformada utilizando nuevamente *filtros de síntesis* específicos.

Otro algoritmo ampliamente utilizado es la Transformada de Paquetes Wavelet. (WPT). A diferencia de la Transformada Wavelet Discreta, en la WPT tanto los *coeficientes aproximados* como los *coeficientes detallados* se someten al proceso de filtraje, obteniendo así mayor información de la señal.

Anexo 5

A.5 Actor, Entidad Control, Interfaz.

A.5.1 Actor.

Se refiere a los usuarios, dispositivos u otros sistemas que interactúan con el sistema en evaluación. Por ejemplo, para el caso del modelo de la prótesis los actores son los pacientes que utilizan el dispositivo protésico los cuales generan la información de entrada al sistema (Señales ME). Otro actor es el sistema mecánico o sistema virtual en el que se ejecuta las ordenes de movimiento. Dichas órdenes son la información de salida del sistema en evaluación.

A.5.2 Entidad

Se refiere a la información persistente, es decir a los conceptos que se pueden reutilizar en otros sistemas hardware/software, y que no son desarrollados exclusivamente para una aplicación en particular. Por ejemplo, la Transformada Wavelet es una herramienta genérica para el análisis de señales; no es una herramienta exclusiva para el procesamiento de señales ME

A.5.3 Control

A diferencia de la clase Entidad, la clase Control se refiere a los algoritmos computacionales desarrollados exclusivamente para una determinada aplicación. Esta clase se encarga de obtener la información de otros componentes a través de la clase Interfaz y seleccionar la herramienta o regla de control adecuada entre las clases Entidad directamente relacionadas a la clase Control. De esta manera la clase Control genera una respuesta que es presentada a través de una Interfaz la cual encapsula las salidas del sistema.

A.5.4 Interfaz

Representa la información necesaria para la interconexión entre los actores del sistema y la clase control. Al tener una completa definición de la interfaz se facilitará la reutilización del sistema o componente al cual pertenece.