

**EVALUACIÓN DE LA USABILIDAD DE UNA PRÓTESIS VIRTUAL DE MANO
BASADA EN UN MECANISMO DE AGARRE ADAPTATIVO Y GOBERNADA
MEDIANTE SEÑALES EMGs**



Universidad
del Cauca

MARIAN SERNA IMBACHI

Tesis de grado en Automática Industrial.

Director:

CARLOS ALBERTO GAVIRIA LOPEZ
Ingeniero en electrónica y telecomunicaciones

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Electrónica, Instrumentación y Control
Popayán, 2018

EVALUACIÓN DE LA USABILIDAD DE UNA PRÓTESIS VIRTUAL DE MANO
BASADA EN UN MECANISMO DE AGARRE ADAPTATIVO Y GOBERNADA
MEDIANTE SEÑALES EMGs

MARIAN SERNA IMBACHI

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

Ingeniera en Automática Industrial.

Director:
CARLOS ALBERTO GAVIRIA LOPEZ, Ingeniero en electrónica y
telecomunicaciones

Popayán
2018

Dedicatoria < Marian Serna >

A Dios por permitirme el don de la vida, la salud, la enseñanza y paciencia de excelentes maestros, el amor de una familia y amigos, en suma, una educación integral rica en conocimientos teóricos y personales.

A mis padres quienes desde la tierra y el cielo han sido una fuente de apoyo e inspiración en los sueños y logros de mi vida. A mi segunda madre, mi tía, quién ha contribuido infinitamente en mi formación espiritual y personal. A mis hermanos por ser la inspiración en la construcción de un ser ejemplar. A mi abuelito Manuel por creer en mi, y quién desde antes de morir me expresó su sueño de ver un nieto suyo convertirse en un profesional. A mi abuelito Jeremias quién ha sido un apoyo emocional y económico muy grande.

A todos mis compañeros y amigos, especialmente a Cristian Tobar y Luz Angélica Pipicano por el gran equipo que formamos en el logro de esta meta.

A mi madre, de una manera muy especial, quién sé está desde el cielo con una gran sonrisa de orgullo y una lagrima de felicidad al ver a su hija convertida en toda una ingeniera. ¡Ya soy ingeniera mami, este logro es de las dos!

A mi pareja, quién ha sido una hermosa luz, fuente de amor, apoyo, comprensión y motivación constante.

Finalmente, a todas las enseñanzas y ejemplos de aquellas personas con las que nuestro sendero de la vida se intersectó en algún momento.

Agradecimientos

A Dios por ser mi fuente de ánimo, fuerza y amor inagotable. Por permitirme el compromiso, responsabilidad y disciplina en la culminación de esta importante etapa profesional.

A mi director de tesis, el ingeniero Carlos Alberto Gaviria por haberme brindado la oportunidad de realizar esta tesis y sobre todo por su paciente y comprometida guía llena de conocimientos en este proceso de investigación. Por usted esto es posible.

Al ingeniero en automática industrial Juan Felipe Solarte y el ingeniero mecánico Ivan Paez por sus esenciales aportes técnicos que hicieron posible la base y punto de partida de este proyecto.

A la universidad del Cauca por las herramientas tangibles e intangibles que permiten la culminación de este proyecto.

A mi padre por darme la oportunidad de tener una excelente educación y ser un apoyo económico fundamental.

Finalmente a mis amigos y familiares por creer en mi y haber hecho de mi etapa universitaria un trayecto lleno de enriquecedoras vivencias que nunca olvidaré.

Resumen

Por lo que se refiere a diseños de prótesis antropomórficas, existen numerosos aspectos que se deben mejorar para acercarse a la verdadera funcionalidad de una mano humana, tal como la capacidad natural de agarre y adaptación de objetos con diferentes formas y tamaños sin la necesidad de conocer previamente qué exactamente se va a sujetar. Además de las estrategias utilizadas para el control de las prótesis desde EMGs que requieren complejos entrenamientos por parte del usuario con amputación de miembro superior para lograr dominarla con facilidad. Al mismo tiempo es necesario que la prótesis sea ligera, buscando el menor peso posible en la implementación de motores, cableado, sensores, etc, lo cuál exige la utilización de mecanismos subactuados, autoadaptativos, simples y eficientes.

Con el fin de averiguar si las ventajas de una prótesis con mecanismos de autoadaptación podrían contribuir a mejorar la percepción de los usuarios en cuanto a la facilidad de gobierno de la misma a partir de señales EMG superficiales, se procede a la simulación dinámica multicuerpo entre una prótesis de mano autoadaptativa subactuada y objetos que puedan ser agarrados a partir de señales electromiográficas. El control del agarre de los objetos se realiza gracias a la conexión entre Matlab, Adams, una placa e-health v 2.0 y una placa de arduino Mega 2560. Matlab es el software que contiene los algoritmos de control, Adams es quién proporciona la información dinámica de la planta (prótesis de mano), y finalmente, el trabajo conjunto de las placas e-health v 2.0, arduino Mega 2560, y 3 electrodos permiten la adquisición de las señales EMGs de un grupo de estudiantes sin amputación de miembro superior.

Al realizar las pruebas con el grupo de estudiantes se concluye una fácil percepción por parte del usuario en cuanto al manejo de la prótesis virtual de mano.

Palabras clave: Protésis de mano, autoadaptabilidad, mecanismos subactuados, prótesis gobernadas por EMGs, electromiografía, mano protésica antropomórfica, control mioeléctrico.

Abstract

As far as anthropomorphic prosthesis designs are concerned, there are many aspects that must be improved in order to approach the true functionality of a human hand, such as the natural ability to grip and adapt objects with different shapes and sizes without the need to know previously what exactly is going to be fastened. In addition to the strategies used to control prostheses from EMGs that require complex training by the user with upper limb amputation to achieve mastery easily. At the same time it is necessary that the prosthesis is lightweight, looking for the lowest possible weight in the implementation of motors, wiring, sensors, etc, which requires the use of simple, efficient, self-adaptative and underactuated mechanisms.

In order to find out if the advantages of a prosthesis with self-adaptation mechanisms could contribute to improve the perception of the users regarding the ease of governing it from superficial EMG signals, we proceed to the dynamic multibody simulation between a self-adaptive and underactuated hand prosthesis and objects that can be grasped from electromyographic signals. The control of the grip of the objects is made thanks to the connection between Matlab, Adams, an e-health v 2.0 board and an Arduino Mega 2560 board. Matlab is the software that contains the control algorithms, Adams is the one who provides the information plant dynamics (hand prosthesis), and finally, the joint work of the e-health v 2.0 board, Arduino Mega 2560 board, and 3 electrodes allow the acquisition of EMG signals from a group of students without upper limb amputation.

When carrying out the tests with the group of students, an easy perception is concluded by the user regarding the handling of the virtual hand prosthesis.

Key words: prosthetic hand, self-adaptative, underactuated mechanism, prostheses governed by EMGs, electromyography, anthropomorphic prosthetic hand, myoelectric control.

Índice general

Agradecimientos	II
Resumen	III
Abstract	IV
Introducción	1
1. MÉTODOS Y MARCO TEÓRICO	3
1.1. Cinemática de mecanismos	3
1.1.1. Diagramas cinemáticos [1, 2]	3
1.1.2. Movilidad o grados de libertad [1, 2]	6
1.1.2.1. Ecuación de Gruebler	6
1.1.3. Eslabones y uniones usados comúnmente [1, 2]	7
1.1.3.1. Unión de perno en una ranura	7
1.1.4. Mecanismo de 4 barras [1, 2]	7
1.1.5. Modelo cinemático directo	8
1.1.6. Cálculo de la jacobiana	9
1.1.6.1. Modelo cinemático para cadenas cerradas	9
1.1.6.2. Método experimental	12
1.2. Sub-actuación y auto-adaptabilidad	12
1.2.1. Mecanismos sub-actuados acoplados	13
1.2.2. Mecanismos sub-actuados auto-adaptables	14
1.2.2.1. Autoadaptación con dos o más actuadores	15
1.2.2.2. Autoadaptación sistema Pull On – Off o SA pasivo	15
1.2.2.3. Autoadaptación sistema de deslizamiento	16
1.3. Control de posición	16
1.3.1. Estrategias de control	17
1.3.1.1. Sintonización de controladores PID (Para el esquema PID en espacio cartesiano)	17
2. ENTORNO VIRTUAL	20
2.1. Requerimientos	20
2.2. Elección de software	21
2.3. Entorno virtual diseñado	26
2.4. Diseño CAD	28
2.4.1. Sub-actuación y auto adaptabilidad de la prótesis	29
2.4.1.1. Mecanismo de dedo A	29

2.4.1.1.1. Diagrama cinemático	30
2.4.1.2. Mecanismo de dedo B	37
2.4.1.2.1. Diagrama cinemático	38
3. Control de posición	46
3.1. Requerimientos de software	46
3.1.1. Elección de software	49
3.2. Requerimientos para la estrategia de control	49
3.3. Virtual Spring-Damper (VSD)	50
3.3.1. Obtención de la jacobiana	51
3.3.1.1. Método analítico	51
3.3.1.2. Método experimental	53
3.3.2. Sintonía de la ley de control	58
3.4. PID en el espacio operacional	59
3.4.1. Esquema de control	60
3.4.1.1. Conclusión	62
3.5. Esquema de control cascada sintonizado por IMC	63
3.5.1. Teoría	63
3.6. Implementación	65
3.6.1. Identificación de las plantas	66
3.6.2. Esquema de control	70
3.6.2.1. Conclusión	79
4. Gobierno con EMG	80
4.1. Teoría	80
4.1.1. Señales electromiograficas (EMG)	80
4.1.1.1. Electromiografía (EMG)	80
4.1.1.2. Electroodos	81
4.1.1.2.1. Perturbación de la señal captada del sensor sEMG	82
4.1.1.3. Requisitos y recomendaciones para el estudio con electrodos EMGs	82
4.1.1.3.1. Consideraciones generales	83
4.1.1.4. Procesamiento de la señal- Normalización de amplitud	84
4.1.1.4.1. El concepto de MVC Normalización	84
4.1.1.4.2. La práctica de la Normalización MVC	84
4.1.1.4.3. Propuesta para MVC - posiciones de prueba	85
4.1.1.5. Sistema básico de adquisición de señales EMG	86
4.2. Requerimientos	87
4.2.1. E-Health Sensor Platform V2.0	87
4.3. Implementación	87
4.3.1. Hardware	87
4.3.2. Software	88
4.3.3. Estrategia umbral flexión-extensión	88
4.4. Objetivos no alcanzados	89
4.5. Conclusión	90

5. Evaluación de aceptación de usuario	91
5.1. Teoría	91
5.1.1. Usabilidad y facilidad de uso	91
5.1.1.1. Atributos de la usabilidad	92
5.1.2. Escala de medida	92
5.1.2.1. Escala tipo Likert	93
5.1.3. Escala de Usabilidad del Sistema SUS	94
5.2. Protocolo	94
5.3. Encuesta de evaluación	96
5.3.1. Encuesta diseñada	96
5.4. Resultado	98
5.4.1. Resultados de la encuesta	98
5.4.2. Resultados de la Escala de Usabilidad del Sistema - SUS	102
5.5. Conclusiones	102
Conclusiones y Trabajos Futuros	104
A. Procedimientos en SolidWorks, Adams y Matlab	105
A.1. Importación de un modelo CAD de SolidWorks en Adams	105
A.2. Comunicación de la planta en Adams con Matlab	116
B. Procedimiento software y hardware para comunicar Arduino, E-Health y sensores EMGs	119
B.1. Ensamble de Arduino, E-Health y sensores EMGs	119
B.2. Lectura de los electrodos EMGs desde Arduino	121
Bibliografía	122

Índice de figuras

1.1. Símbolos que se utilizan en los diagramas cinemáticos. Tomado de [1].	4
1.2. Símbolos que se utilizan en los diagramas cinemáticos. Tomado de [1].	5
1.3. Pinzas. Tomado de [1].	5
1.4. Diagrama cinemático de las pinzas. Tomado de [1].	6
1.5. Unión de perno en una ranura: a) Unión real de perno en una ranura y b) Modelo de perno en una ranura. Tomado de [1].	7
1.6. Mecanismo del limpiador para el cristal trasero: a) Mecanismo real y b) Diagrama cinemático. Tomado de [1].	8
1.7. Mecanismo de cuatro barras.	10
1.8. Oscilación Sostenida.	18
2.1. Software requerido para el proyecto.	21
2.2. Representación de entrada salida de una planta en Adamas.	25
2.3. Representación de un sistema de control en lazo cerrado usando el bloque de la planta generado por Adams.	25
2.4. Archivo Adams_sys que simula la dinámica de la planta.	25
2.5. Contenido en el interior del bloque de adams_sub	26
2.6. Entorno virtual de la prótesis de mano con esfera de 20 mm.	27
2.7. Entorno virtual de la prótesis de mano con esfera de 40 mm.	27
2.8. Entorno virtual de la prótesis de mano con Cubo de 57 mm.	27
2.9. Entorno virtual de la prótesis de mano con Cubo de de 47 mm.	28
2.10. Entorno virtual de la prótesis de mano agarrando esfera de 40 mm.	28
2.11. Entorno virtual de la prótesis de mano agarrando esfera de 40 mm.	28
2.12. Casos en los que entra en funcionamiento alguno de los mecanismos que componen el modelo de dedo A.	30
2.13. Diseño CAD del mecanismo de dedo A con motor lineal, dedo índice.	31
2.14. Diagrama cinemático modelo de dedo A para actuación con motor lineal.	31
2.15. Diagrama cinemático modelo de dedo A para actuación con motor rotacional.	32
2.16. Diseño CAD del mecanismo de dedo A con motor rotacional, dedo índice.	32
2.17. Mecanismos de SA y AA del modelo de dedo A para actuación con motor lineal.	33
2.18. Mecanismos de SA y AA del modelo de dedo A para actuación con motor rotacional.	33
2.19. Diagrama cinemático modelo de dedo A para actuación con motor lineal antes del contacto con un objeto.	34

2.20. Diagrama cinemático modelo de dedo A para actuación con motor rotacional antes del contacto con un objeto.	34
2.21. Diagrama cinemático modelo de dedo A para actuación con motor lineal, caso AA-FP.	35
2.22. Diagrama cinemático modelo de dedo A para actuación con motor rotacional, caso AA-FP.	36
2.23. Diagrama cinemático modelo de dedo A para actuación con motor lineal, caso AA-FP-FM.	37
2.24. Diagrama cinemático modelo de dedo A para actuación con motor rotacional, caso AA-FP-FM.	37
2.25. Casos en los que entra en funcionamiento alguno de los mecanismo que componen el modelo de dedo B.	38
2.26. Casos en los que entra en funcionamiento alguno de los mecanismo que componen el modelo de dedo B.	39
2.27. Diagrama cinemático modelo de dedo B para actuación con motor lineal.	39
2.28. Diagrama cinemático modelo de dedo B para actuación con motor rotacional.	39
2.29. Diseño CAD del mecanismo de dedo B con motor rotacional.	40
2.30. Mecanismos de SA y AA del modelo de dedo B para actuación con motor lineal.	40
2.31. Mecanismos de SA y AA del modelo de dedo B para actuación con motor rotacional.	40
2.32. Diagrama cinemático modelo de dedo B para actuación con motor lineal antes del contacto con un objeto.	41
2.33. Diagrama cinemático modelo de dedo B para actuación con motor rotacional antes del contacto con un objeto.	42
2.34. Diagrama cinemático modelo de dedo B para actuación con motor lineal, caso AA-M.	43
2.35. Diagrama cinemático modelo de dedo B para actuación con motor rotacional, caso AA-M.	44
2.36. Diagrama cinemático modelo de dedo B para actuación con motor lineal, caso AA-FP-FM.	44
2.37. Diagrama cinemático modelo de dedo B para actuación con motor rotacional, caso AA-FP-FM.	45
3.1. Estrategia de gobierno de prótesis de mano virtual.	46
3.2. Esquema general del control VSD. Tomado de [3].	51
3.3. Entradas y salidas del bloque de la planta en simulink para hallar la jacobiana.	54
3.4. Trayectoria cartesiana obtenida para el modelo de dedo A.	55
3.5. Trayectoria de la articulación actuada obtenida para el modelo de dedo A.	55
3.6. Entorno de la herramienta <i>cftool</i> para estimación de la función que representa la fila 2 de la jacobiana.	56
3.7. Comparación de la fila 1 de la jacobiana numérica y la función de la jacobiana estimada con entrada del parámetro <i>th</i>	57

3.8. Comparación de la fila 2 de la jacobiana numérica y la función de la jacobiana estimada con entrada del parámetro th (sin información de los 500 primeros datos).	57
3.9. Comparación entre la velocidad cartesiana $\dot{X}_{simulacion}$ y $\dot{Y}_{estimada}$	58
3.10. Comparación entre la velocidad cartesiana $\dot{Y}_{simulacion}$ y $\dot{Y}_{estimada}$	58
3.11. Diagrama de bloques de un esquema de control PID en el espacio de tarea. Tomado de [4].	59
3.12. Fases y condiciones para realizar el control PID en el espacio de tarea para la prótesis de mano.	60
3.13. Implementación de un controlador en el espacio de tarea para el control de posición de la yema del modelo de dedo A.	61
3.14. Implementación de un controlador en el espacio de tarea discretizado para el control de posición de la yema del modelo de dedo A.	62
3.15. Estructura de control cascada. Tomado de [5].	63
3.16. Interfaz de usuario de toolbox de identificación Ident	66
3.17. Comparación entre modelo real y estimado por ident.	67
3.18. Diagrama de bloques para realizar la identificación del modelo de estados para el dedo pulgar e índice.	67
3.19. Generación del archivo <i>Controls_plant</i> desde Adams con las entradas y salidas de la planta.	68
3.20. Cambio de ángulo generado por una excitación de par a la entrada de la planta del dedo Pulgar.	69
3.21. Cambio de ángulo generado por una excitación de par a la entrada de la planta del dedo Índice.	69
3.22. Esquema de control PID y LQR con observador de estados para un dedo.	71
3.23. Bloque de <i>Adams_sub</i> generado desde Adams con 5 entradas y 5 salidas.	72
3.24. Trayectoria grado 5 para el dedo Índice, desde 0° hasta -30°	72
3.25. Trayectoria grado 5 para el dedo Pulgar, desde 0° hasta 10°	73
3.26. Trayectoria grado 5 seguida por el dedo Pulgar, desde 0° hasta 10°	73
3.27. Trayectoria grado 5 seguida por el dedo Índice, desde 0° hasta -30°	73
3.28. Error de seguimiento de la trayectoria grado 5 para el dedo Pulgar.	74
3.29. Error de seguimiento de la trayectoria grado 5 para el dedo Índice.	74
3.30. Prótesis de mano en <i>Adams</i> controlada en posición final de 10° para el dedo Pulgar y de -30° para el dedo Índice, Medio, Anillo y Meñique.	74
3.31. Prótesis virtual de mano agarrando una esfera de 40 mm	75
3.32. Esfuerzo de control del dedo Pulgar en el agarre de una esfera.	75
3.33. Esfuerzo de control del dedo Índice en el agarre de una esfera.	76
3.34. Esfuerzo de control del dedo Medio en el agarre de una esfera.	76
3.35. Esfuerzo de control del dedo Anillo en el agarre de una esfera.	76
3.36. Esfuerzo de control del dedo Meñique en el agarre de una esfera.	77
3.37. Error de seguimiento en trayectoria del dedo Pulgar en agarre de objeto.	77
3.38. Error de seguimiento en trayectoria del dedo Índice en agarre de objeto.	77
3.39. Error de seguimiento en trayectoria del dedo Medio en agarre de objeto.	78
3.40. Error de seguimiento en trayectoria del dedo Anillo en agarre de objeto.	78

3.41. Error de seguimiento en trayectoria del dedo Meñique en agarre de objeto.	78
4.1. Esquema representativo del mecanismo básico de control y de los componentes de la unidad motora. Tomado de [6].	81
4.2. Detección de señales mioeléctricas. Tomado de [7].	81
4.3. El concepto de normalización MVC. Antes de la prueba / ejercicios, se realiza una contracción MVC estática para cada músculo. Este nivel de inervación MVC sirve como nivel de referencia (= 100 %) para todos los ensayos futuros. Tomado de [8].	85
4.4. Propuesta para el arreglo de prueba MVC del brazo. La flecha delgada negra indica la dirección del movimiento, las flechas gruesas blancas la dirección de la resistencia. Tomado de [8].	86
4.5. Sistema básico de adquisición de señales EMG. Tomado de [7].	86
4.6. Ensamblaje de placa E-Health y Arduino.	88
4.7. Definición de trayectoria de referencia a partir de señales EMGs en Simulink.	89
5.1. Ejemplo de escala de Likert. Tomado de [9].	93
5.2. Ejemplo de una adaptación a la escala de Likert. Tomado de [9].	94
5.3. Ejemplo de una adaptación a la escala de Likert.	94
5.4. Puntuación de la Escala de Usabilidad del Sistema en termino de adjetivos.	94
5.5. Respuesta a pregunta 1 de la encuesta de validación.	98
5.6. Respuesta a pregunta 2 de la encuesta de validación.	99
5.7. Respuesta a pregunta 3 de la encuesta de validación.	99
5.8. Respuesta a pregunta 4 de la encuesta de validación.	99
5.9. Respuesta a pregunta 5 de la encuesta de validación.	100
5.10. Respuesta a pregunta 6 de la encuesta de validación.	100
5.11. Respuesta a pregunta 7 de la encuesta de validación.	100
5.12. Respuesta a pregunta 8 de la encuesta de validación.	101
5.13. Respuesta a pregunta 9 de la encuesta de validación.	101
5.14. Respuesta a pregunta 10 de la encuesta de validación.	101
5.15. Escala de usabilidad de la prótesis de mano virtual.	102
A.1. Inicio del software Adams.	106
A.2. Creación de un nuevo modelo en Adams.	106
A.3. Importación del modelo CAD.	107
A.4. Elegir el archivo Parasolid exportado desde SolidWorks.	107
A.5. Elegir el archivo Parasolid exportado desde SolidWorks.	108
A.6. Diseño CAD exportado en Adams en vista no renderizada.	108
A.7. Diseño CAD exportado en Adams en vista renderizada.	109
A.8. Configuración de las propiedades másicas para una pieza.	109
A.9. Tipos de juntas en Adams.	110
A.10. Sección de Forces para aplicar un Torque rotacional.	111
A.11. Pestaña de modificación de un torque.	112
A.12. Selección del icono de medidas en la Pestaña Modify Joint	112

A.13.Pestaña Joint Measure para crear medidas de ángulo, fuerzas, velocidades, etc.	113
A.14.Pestaña Orientation Measure para modificar los marcadores de medida de ángulo.	113
A.15.Medir ángulos cambiando los marcadores por defecto.	114
A.16.Base de datos de todos los marcadores del proyecto en Adams.	114
A.17.Pestaña de Simulation Control para iniciar la simulación.	115
A.18.Menú Element para crear una variable de estado.	115
A.19.Creando una variable de estado.	115
A.20.Asignación de las variables de estado a los componentes de medidas.	116
A.21.Plugin Adams Controls.	117
A.22.Carga de variables Adams en el Workspace de Matlab.	117
A.23.Archivo de Simulink <i>Adams_sys</i>	118
A.24.Configuración de Animación ADAMS	118
B.1. Ensamblaje de placa E-Health y Arduino.	119
B.2. Puente entre los pines ECG/EMG 2 Y 3.	120
B.3. Sensores EMGs con electrodos adhesivos.	120
B.4. Conexión de electrodos EMGs a la placa E-Health.	120
B.5. Separación del protector de los electrodos.	121
B.6. Ubicación de los elctrodos sobre la mano de un usuario.	121
B.7. Potenciómetro GAIN de la placa E-Health.	121
B.8. Código en arduino para leer sensores EMGs.	122

Índice de tablas

1.1. Valores de sintonización para un controlador P, PI y PID a partir de K_{cr} y P_{cr}	18
2.1. Software para ingeniería de diseño mecánico 3D disponible.	22
2.2. Software simulación dinámica disponible.	23
3.1. Software de programación e implementación de algoritmos de control disponible.	48

Introducción

El diseño de prótesis de mano evoluciona continuamente en busca de la mejora de algunos aspectos tales como métodos de control, ergonomía, costos, autonomía y eficiencia energética con el objetivo último de devolver la funcionalidad a los amputados de miembro superior y reducir el abandono de las prótesis. Algunas de las razones para este abandono son por ejemplo, la percepción por parte de los pacientes con amputación de miembro superior de que el gobierno de su prótesis de mano es muy complejo, incluyendo el tiempo que les toma entrenarse para gobernar sus movimientos, lo que requiere una alta carga cognitiva [10]. A lo anterior se agregan algunos otros aspectos críticos tales como las interfaces adoptadas para su control, la falta de fuerza o de retroalimentación táctil, peso del dispositivo, la destreza y capacidades de agarre de objetos, uso de información limitada (señales EMG registradas) que se utiliza para activar varios grados de libertad (DoFs) de la extremidad artificial, con lo que el control de prótesis no es intuitivo y es antinatural, o la necesidad de movimientos independientes para el pulgar, índice y muñeca [11].

En el diseño de prótesis de mano antropomórficas es muy importante buscar el mejor funcionamiento con el número mínimo de actuadores [12], lo que lleva a la necesidad de mecanismos que puedan mover varias articulaciones con una sola articulación actuada, es decir, mecanismos subactuados. Al mismo tiempo se requiere de mecanismos de autoadaptación tales que los dedos tengan la capacidad de adaptarse en el agarre de diferentes geometrías y tamaño de objetos [13, 14], sin requerir de intervención supervisada para dicha adaptación. Esta autoadaptación lleva no solo a reducir el número de articulaciones actuadas, sino también a reducir la complejidad del control aprovechando las sinergias adaptativas [15].

Las señales de electromiografía superficial (EMGs) pueden usarse para gobernar la fuerza del agarre y la postura de la mano, sin embargo, es difícil controlar estos dos aspectos simultáneamente, ya que estas señales son muy débiles, están afectadas por fatiga, sudoración, se necesitan varios canales [16], requieren de una gran habilidad y arduo entrenamiento del usuario para producir cambios discriminantes en las señales que están a cargo del gobierno de múltiples grados de libertad de la prótesis [17].

El tema a desarrollar en este trabajo de grado es la evaluación de las ventajas potenciales derivadas de implementar un mecanismo de autoadaptación para el agarre de objetos en una prótesis de mano activa virtual, en cuanto a si facilitan el gobierno de la mano desde señales de EMG. Para responder a este problema planteado se plantea el siguiente camino metodológico: 1) Construir un entorno vir-

tual que permita implementar el diseño CAD de la prótesis autoadaptativa donde se pueda evaluar su desempeño en cuanto a la realización de movimientos de flexión y extensión en presencia de objetos de diferentes formas y tamaño, considerando efectos tales como la gravedad, colisión con objetos, y medición de variables tales como posición de las yemas, ángulos y pares; 2) Construir un controlador para la flexión y extensión de la mano; 3) Proponer una estrategia para gobernar la flexión y extensión de la mano mediante señales EMG de un músculo de un usuario sin amputación, 4) evaluar la percepción del usuario en cuanto a la facilidad del gobierno de la flexión y extensión de la mano en el entorno virtual.

Al llegar a este punto conviene mencionar que el presente proyecto de grado está íntimamente ligado a dos proyectos de maestría, el primero tiene como objetivo la implementación real de la prótesis de mano virtual elegida, cuyo diseño es proporcionado por el segundo proyecto, además de establecer un control para el agarre de objetos. Con el propósito de llevar a cabo el primer proyecto es necesario desarrollar una estrategia de control sencilla que no requiera de un modelo matemático complejo, ni de una realimentación de sensores por cada articulación y tampoco de una actuación de numerosos motores, ya que para lograr la funcionalidad de una prótesis no es práctico saturarla de dispositivos y conexiones que generan un gran peso adicional.

El documento está organizado como sigue. El capítulo 1 establece el marco teórico con el propósito de presentar los conceptos necesarios para la comprensión de los procedimientos realizados en este proyecto. En el capítulo 2 se describe cómo fue construido el entorno virtual, sus virtudes y limitaciones. En el capítulo 3 se describe la estrategia de control utilizada para lograr el control de posición en movimientos de flexión y extensión de la mano. El capítulo 4 trata sobre la estrategia plantada para el gobierno de la mano utilizando señales EMGs. En el capítulo 5 se presenta la evaluación de la percepción de uso sobre una muestra de estudiantes sin amputación de miembro superior. En el capítulo final se resumen las conclusiones y trabajos futuros.

Capítulo 1

MÉTODOS Y MARCO TEÓRICO

El propósito de este capítulo es explicar algunos de los procedimientos, conceptos y métodos que se utilizan en la obtención de resultados del trabajo de grado, a fin de que el lector pueda en los capítulos siguientes, entender el origen de los resultados.

1.1. Cinemática de mecanismos

La mano considerada para este trabajo utiliza mecanismos de barras para el movimiento de las falanges de los dedos. Este tipo de estructuras llevan a la existencia de cadenas cinemáticas cerradas, lo que implica una complejidad adicional en las herramientas para análisis cinemático. Se resumen algunos de los procedimientos que se tuvieron en cuenta para este propósito.

1.1.1. Diagramas cinemáticos [1, 2]

Los eslabones son las partes individuales de un mecanismo y se consideran cuerpos rígidos que están conectados con otros eslabones para transmitir movimiento y fuerzas. Los eslabonamientos pueden ser cadenas abiertas o cerradas. Una cadena cinemática es una agrupación de varios eslabones unidos por medio de pares cinemáticos. Cuando cada eslabón de la cadena cinemática se conecta al menos con otros dos, esta forma uno o más bucles cerrados, definiéndose una (ó varias) cadena cinemática cerrada. En una cadena abierta por lo menos un eslabón está conectado únicamente a otro eslabón.

Para que una cadena cinemática se convierta en mecanismo, se necesita que un eslabón esté fijo, de forma que el movimiento de todos los demás puntos se medirá con respecto al eslabón que se considere fijo. El eslabón fijo se denomina bancada.

En el análisis del movimiento de una máquina, con frecuencia se dificulta visualizar el movimiento de los componentes en el dibujo completo de un ensamble. Una imagen completa de una maquina es muy compleja, por lo que resulta difícil concentrarse en el movimiento del mecanismo en consideración. Es más fácil representar las partes de manera esquemática, de modo que solo se muestren las dimensio-

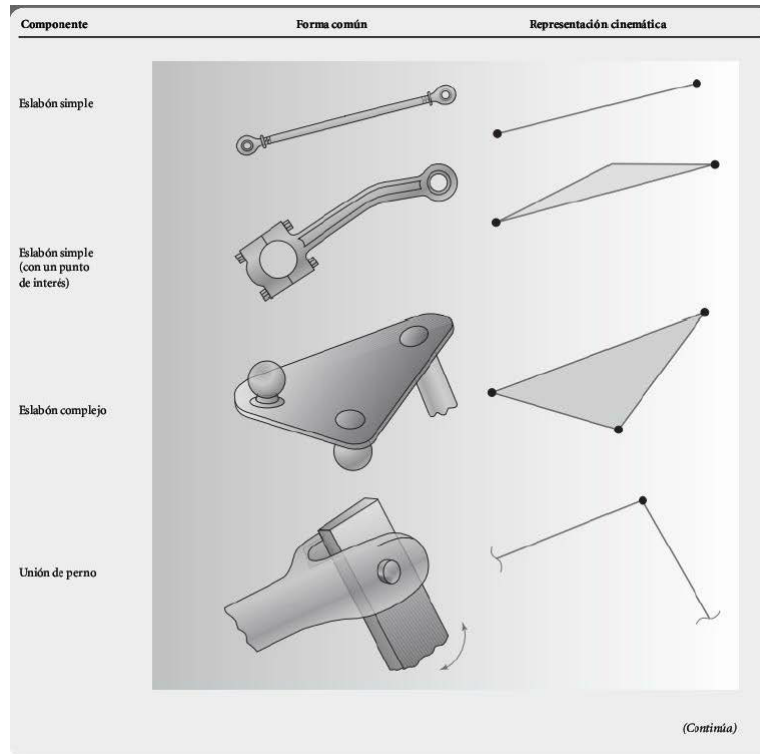


Figura 1.1: Símbolos que se utilizan en los diagramas cinemáticos. Tomado de [1].

nes que influyen en el movimiento del mecanismo. Tales diagramas “desmontados” se conocen con frecuencia como diagramas cinemáticos, cuyo propósito es similar al de los diagramas esquemáticos de los circuitos eléctricos o de los diagramas de tubería, donde se representan las variables que afectan la función principal del mecanismo.

Las figuras 1.1 y 1.2 muestran las convenciones comunes que se usan en la elaboración de los diagramas cinemáticos. Se requiere que un diagrama cinemático se dibuje a una escala proporcional con el mecanismo real. Para efectos de identificación, los eslabones se numeran, iniciando con la bancada como el eslabón número 1. Para evitar confusión, las uniones se identifican con letras.

Como ejemplo se verá el diagrama cinemático de las pinzas de la figura 1.3. El primer paso en la elaboración de un diagrama cinemático es decidir la parte que se diseñará como la bancada. El movimiento de todos los demás eslabones se determinará en relación con la bancada. En algunos casos, la selección es evidente porque la bancada está firmemente sujeta al suelo. En este problema no hay partes sujetas al suelo, por consiguiente, la selección de la bancada es arbitraria. Se designa el mango superior como bancada. El movimiento de todos los demás eslabones se determinará en relación con el mango superior. El mango superior se identifica como el eslabón 1.

El segundo paso es identificar todos los demás eslabones. Una observación cuidadosa revela otras partes que se mueven:

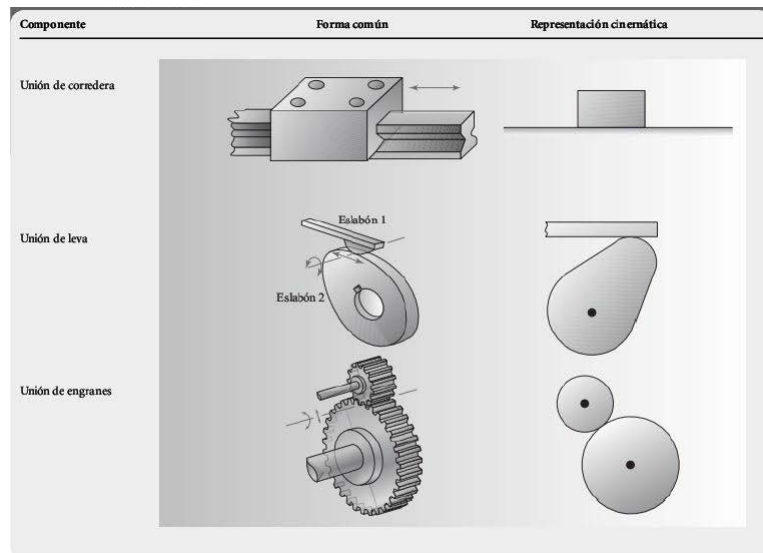


Figura 1.2: Símbolos que se utilizan en los diagramas cinemáticos. Tomado de [1].

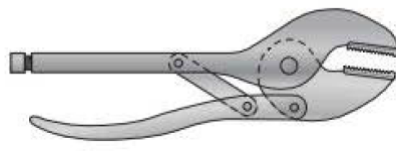


Figura 1.3: Pinzas. Tomado de [1].

- Eslabón 2: Mango inferior.
- Eslabón 3: Mordaza inferior.
- Eslabón 4: Barra que conecta el mango superior y el mango inferior.

El tercer paso es identificar las uniones. Se utilizan cuatro pernos para conectar estos eslabones (el eslabón 1 al 2, el 2 al 3, el 3 al 4 y el 4 al 1). Estas uniones se identifican con las letras A a D.

El cuarto paso es identificar los puntos de interés.

Se desea conocer el movimiento en el extremo de la mordaza inferior, el cual se designa como el punto de interés X. Finalmente, también se busca determinar el movimiento en el extremo del mango inferior, que se designa como el punto de interés Y.

El quinto paso es elaborar el diagrama cinemático.

El diagrama cinemático se observa en la figura 1.4.

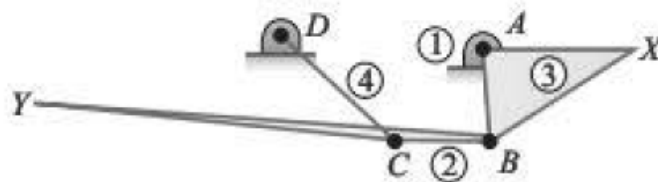


Figura 1.4: Diagrama cinemático de las pinzas. Tomado de [1].

1.1.2. Movilidad o grados de libertad [1, 2]

Una propiedad importante en el análisis de mecanismos es el número de grados de libertad del eslabonamiento. El grado de libertad es el número de entradas independientes requeridas para posicionar con exactitud todos los eslabones de un mecanismo con respecto al suelo. También se puede definir como el número de actuadores necesarios para operar el mecanismo.

El número de grados de libertad de un mecanismo también se conoce como movilidad, el cual se identifica con el símbolo M .

Si un mecanismo plano posee n eslabones, cada uno de ellos, antes de conectarse, poseerá tres grados de libertad, excepto el eslabón fijo ó bancada. Luego antes de conectarse, el número de grados de libertad será de:

$$3(n - 1) \quad (1.1)$$

1.1.2.1. Ecuación de Gruebler

A medida que se van conectando eslabones por medio de pares, se está restringiendo el movimiento relativo entre ellos, por lo tanto, una vez conectados todos los

eslabones, el número de grados de libertad para eslabonamientos planos conectados con uniones comunes se calculan con la ecuación (1.2) de Gruebler.

$$M = \text{grados de libertad} = 3 * (n - 1) - 2j_p - j_h \quad (1.2)$$

Donde:

n = número total de eslabones en el mecanismo.

j_p = número total de uniones principales (uniones de pernos o de correderas).

j_h = número total de uniones de orden superior (uniones de levas, engranes o de perno en una ranura.)

1.1.3. Eslabones y uniones usados comúnmente [1, 2]

1.1.3.1. Unión de perno en una ranura

Una conexión común entre eslabones es la unión de perno en una ranura, como la que se ilustra en la figura 1.5a. Se trata de una unión de orden superior porque permite que los dos eslabones giren y se deslicen entre sí. Para simplificar el análisis cinemático, se utilizan las uniones principales para modelar esta unión de orden superior. La unión de perno en una ranura se vuelve una combinación de unión de perno y unión de corredera, como en la figura 1.5b. Se observa que así se agrega otro eslabón al mecanismo. En ambos casos, el movimiento relativo entre los eslabones es el mismo. No obstante, el uso de un modelo cinemático con las uniones principales facilita el análisis.

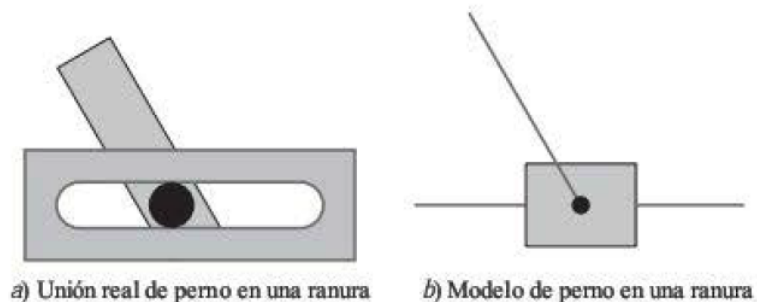


Figura 1.5: Unión de perno en una ranura: a) Unión real de perno en una ranura y b) Modelo de perno en una ranura. Tomado de [1].

1.1.4. Mecanismo de 4 barras [1, 2]

El eslabonamiento más simple y más común es el eslabonamiento de cuatro barras. Es una combinación de cuatro eslabones, uno designado como la bancada y conectado por cuatro uniones de perno.

La figura 1.6a muestra el mecanismo de un sistema de limpiador para el cristal trasero de un automóvil. El diagrama cinemático se presenta en la figura 1.6b. Se observa que es un mecanismo de cuatro barras, ya que se integra con cuatro eslabones conectados por cuatro uniones de perno y un eslabón está impedido para

moverse.

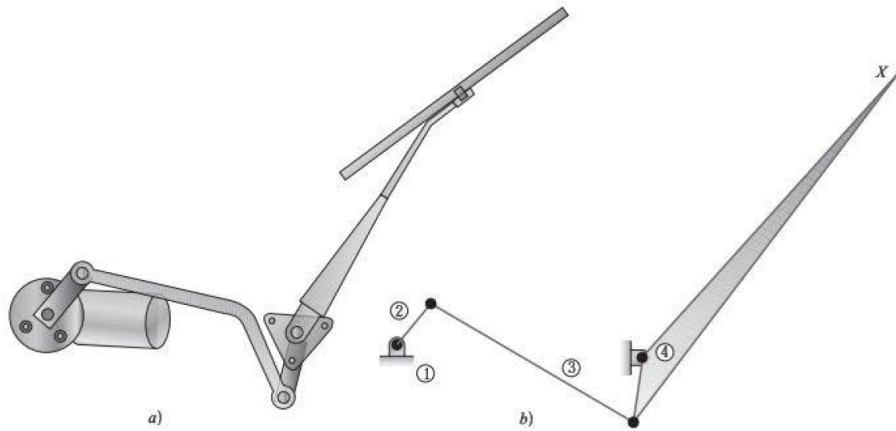


Figura 1.6: Mecanismo del limpiador para el cristal trasero: a) Mecanismo real y b) Diagrama cinemático. Tomado de [1].

La movilidad de un mecanismo de cuatro barras es como sigue.

$$n = 4, j_p = 4 \text{ pernos}, j_h = 0$$

Y

$$M = 3(n - 1) - 2j_p - j_h = 3(4 - 1) - 2(4) - 0 = 1$$

Como el mecanismo de cuatro barras tiene un grado de libertad, está restringido a un solo actuador o es totalmente operado por este. El sistema del limpiador de la figura 1.6 es activado por un motor eléctrico de corriente continua.

Por supuesto, el eslabón impedido para moverse se elige como la bancada. Por lo general, el eslabón pivote conectado al impulsor o a la fuente de potencia se conoce como eslabón de entrada. El otro eslabón pivote, sujeto a la bancada, se designa como el eslabón de salida o seguidor. El acoplador o brazo conector “acopla” el movimiento del eslabón de entrada con el del eslabón de salida.

1.1.5. Modelo cinemático directo

En robótica, el modelo cinemático directo describe las velocidades de las articulaciones del robot en el espacio operacional (donde se encuentra el organo terminal o funcional del robot) en función de las velocidades de estas articulaciones expresadas en el espacio articular. El modelo cinemático directo tiene por expresión:

$$\dot{X} = J(q)\dot{q} \quad (1.3)$$

En la ecuación 1.3, $J(q)$ es la llamada matriz Jacobiana, definida como la derivada parcial entre las posiciones cartesianas del efector final (\dot{X}) y articulares (\dot{q}):

$(\partial X/\partial q)$. La matriz Jacobiana tiene un interés particular en el diseño mecánico, en el análisis de singularidades, y en el diseño de controladores en el espacio operacional.

La matriz Jacobiana tiene una dimensión de $(6 \times n)$, donde n representa el número de grados de libertad del robot. La expresión completa del modelo cinemático directo será:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ w_x \\ w_y \\ w_z \end{pmatrix} = \begin{pmatrix} J \\ (6 \times n) \end{pmatrix} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \cdot \\ \cdot \\ \dot{q}_n \end{pmatrix} \quad (1.4)$$

Obsérvese que, en la ecuación 1.4, el vector \dot{X} se descompone en las velocidades lineales \dot{x} , \dot{y} , \dot{z} , y en las velocidades rotacionales w_x , w_y , w_z . Dado que la Jacobiana tiene una dimensión de $(6 \times n)$, ella será cuadrada solamente para robots de seis grados de libertad, indicándose con esto que dicho robot puede posicionarse perfectamente en un espacio de tres dimensiones, y puede realizar tres rotaciones en su órgano terminal.

Para hallar la jacobiana es necesario partir de los parámetros geométricos del robot y de los elementos de la matriz de transformación, resultados obtenidos dentro del cálculo del modelo geométrico directo.

Para este trabajo, la matriz jacobiana ha sido de interés en la exploración de técnicas de control para el posicionamiento de la yema de los dedos en los movimientos de flexión y extensión, ya que la fuerza que se produce en el efector final como consecuencia del par aplicado en la articulación activa se puede establecer mediante la relación de la ecuación 1.5.

$$\tau = J^T(q) F \quad (1.5)$$

siendo τ el par aplicado a la articulación activa y F la fuerza transmitida al efector final.

1.1.6. Cálculo de la jacobiana

La matriz Jacobiana es dependiente de los ángulos de las articulaciones. Para poder calcularla de forma analítica, es necesario poder calcular todos los ángulos de las articulaciones no actuadas (pasivas) como función de los ángulos de las articulaciones actuadas (activas) que se suponen conocidos.

1.1.6.1. Modelo cinemático para cadenas cerradas

En un lazo cinemático cerrado existe una condición de cierre que se origina dado que los eslabones de una cadena cerrada no pueden abrirse cuando se mueven. Al expresar matemáticamente esa condición, se originan ecuaciones de restricción.

Para dar solución a esas ecuaciones de restricción en mecanismos de 4 barras, se deben establecer relaciones geométricas para cada lazo cerrado tal que existan dos ángulos conocidos y dos desconocidos, con dos ecuaciones linealmente independientes. Partiendo del ángulo conocido, por medio de la ecuación de Freudenstein se hallan los otros dos ángulos desconocidos. Este procedimiento será extraído por claridad del documento [3] y mostrado a continuación para un mecanismo de 4 barras.

Se utilizará a manera de ejemplo ilustrativo simple, el mecanismo planar de cuatro barras de la Figura 1.7, el cual posee un DOF. El hecho de que estos cuatro eslabones siempre formen una cadena cerrada puede ser expresado matemáticamente por medio de las siguientes relaciones:

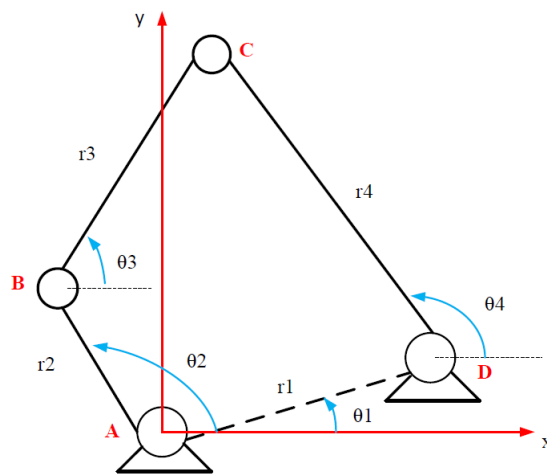


Figura 1.7: Mecanismo de cuatro barras.

$$\vec{AB} + \vec{BC} + \vec{CD} + \vec{DA} = 0 \quad (1.6)$$

$$\begin{aligned} r_1 \cos \theta_1 + r_2 \cos \theta_2 + r_3 \cos \theta_3 + r_4 \cos \theta_4 &= 0 \\ r_1 \sin \theta_1 + r_2 \sin \theta_2 + r_3 \sin \theta_3 + r_4 \sin \theta_4 &= 0 \\ \theta_1 + \theta_2 + \theta_3 + \theta_4 - 2\pi &= 0 \end{aligned} \quad (1.7)$$

Estas ecuaciones escalares se obtienen viendo el mecanismo de cuatro barras como una cadena cinemática serial con cuatro juntas de revolución. Cada conjunto de ecuaciones de este tipo recibe el nombre de ecuaciones de lazo cerrado. Tales restricciones son holonómicas, las cuales pueden escribirse como una ecuación algebraica que involucra únicamente variables de posición pero no de velocidad.

Para dar solución a estas ecuaciones se utilizó el método descrito en [3]; se debe tener en cuenta que el ángulo de entrada θ_2 es conocido ya que está asociado a la articulación actuada y es manipulado por medio de un actuador, θ_1 se conoce de la geometría del mecanismo ya que es un ángulo constante perteneciente al

eslabón fijo (tierra). El sistema de dos ecuaciones no lineales con dos incógnitas de la ecuación (1.7), se conoce como la ecuación de Freudenstein y es usada en la síntesis de mecanismos. Su solución establece la relación cinemática entre la posición del eslabón de salida, con la posición del eslabón de entrada θ_2 .

Considerando la siguiente expresión resultante de ordenar la ecuación 1.7:

$$\begin{aligned} r_3 \cos \theta_3 &= r_1 \cos \theta_1 + r_4 \cos \theta_4 - r_2 \cos \theta_2 \\ r_3 \sin \theta_3 &= r_1 \sin \theta_1 + r_4 \sin \theta_4 - r_2 \sin \theta_2 \end{aligned} \quad (1.8)$$

se elimina θ_3 elevando ambos lados de la ecuación (1.8) al cuadrado y sumando, llegando a:

$$\begin{aligned} [r_3 \cos \theta_3]^2 + [r_3 \sin \theta_3]^2 &= [r_1 \cos \theta_1 + r_4 \cos \theta_4 - r_2 \cos \theta_2]^2 \\ &+ [r_1 \sin \theta_1 + r_4 \sin \theta_4 - r_2 \sin \theta_2]^2 \end{aligned} \quad (1.9)$$

Después de operar y organizar se obtiene:

$$\begin{aligned} (2r_1 r_4 \cos \theta_1 - 2r_2 r_4 \cos \theta_2) \cos \theta_4 + (2r_1 r_4 \sin \theta_1 - 2r_2 r_4 \sin \theta_2) \sin \theta_4 \\ - 2r_1 r_2 (\cos \theta_1 \cos \theta_2 + \sin \theta_1 \sin \theta_2) + r_1^2 + r_2^2 + r_4^2 - r_3^2 = 0 \end{aligned} \quad (1.10)$$

En la ecuación (1.10) sobrevive una incógnita, a saber, θ_4 . Dicha ecuación es de la forma de la ecuación (1.11), conocida como ecuación de Freudenstein que tiene solución analítica conocida, donde α es la incógnita a resolver.

$$A \cos \alpha + B \sin \alpha + C = 0 \quad (1.11)$$

Una vez se tienen todos los parámetros geométricos conocidos, se procede a utilizar el método descrito en [3], consistente en los siguientes pasos:

- Obtener la matriz Jacobiana del mecanismo (J_m), para lo cual se utiliza el método de Khalil donde una cadena cinemática cerrada se abre y descompone en dos cadenas cinemáticas seriales que se restringen a estar unidas en el punto de apertura.
- De las restricciones resultantes, se obtiene la jacobiana de las restricciones J_c .

Teniendo en cuenta el vector de los ángulos de articulaciones pasivas (θ_p), y el vector de los ángulos de las articulaciones activas θ_a , las anteriores matrices jacobianas se particionan en sus componentes de articulaciones activas J_{ma} , J_{cp} y pasivas, J_{mp} , J_{mc} .

Finalmente, la ecuación que conduce a conocer la jacobiana buscada que permite describir las velocidades del efector final como función de las velocidades articulares de las articulaciones activas es la de la ecuación (1.12), tal que la velocidad del efector final v se puede calcular mediante la ecuación (1.13).

$$J = J_{ma} - J_{mp} J_{cp}^{-1} J_{ca} \quad (1.12)$$

$$v = (J_{ma} - J_{mp}J_{cp}^{-1}J_{ca})\dot{\theta}_a = J\dot{\theta}_a \quad (1.13)$$

En la ecuación (1.13), v está compuesto por las velocidades lineales \dot{x} y angulares w .

1.1.6.2. Método experimental

Del modelo cinemático directo se obtienen las velocidades cartesianas del efector final a partir de la Jacobiana y las velocidades angulares.

De la ecuación (1.13) se concluye que $J(\theta) = v/\dot{\theta}_a$. No obstante, v y θ_a son vectores, mientras que $J(\theta)$ es una matriz, por lo cual esta operación no se puede calcular directamente supuesto que se conocen tanto las velocidades de las articulaciones activas y las velocidades del efector final. Para robots seriales existen métodos experimentales donde se inmovilizan mecánicamente todas las articulaciones, excepto una, tal que la matriz se puede ir calculando componente a componente, haciendo que cada término corresponda a una ecuación escalar $J(\theta)_{i,j} = v_i/\dot{\theta}_{aj}$, siendo i una componente del vector de velocidad y j una componente del vector de velocidad angular de las articulaciones activas $\dot{\theta}_a$. Como del modelo geométrico es posible determinar todo el vector θ como función de los ángulos de las articulaciones activas, sin perder generalidad se podría obtener la relación útil para un cálculo experimental de la ecuación escalar (1.14).

$$J(\theta_a)_{i,j} = v_i/\dot{\theta}_{aj} \quad (1.14)$$

Donde el experimento debe variar la articulación activa a analizar, en todo el rango de valores del espacio de trabajo del robot, por lo cual para cada componente de la matriz $J(\theta_a)$ el resultado será una curva de valores como función de las variaciones de velocidad angular de la articulación activa utilizada.

En los métodos experimentales con robots reales, es necesario considerar el efecto de la gravedad en el movimiento, lo cual hace que el método sea un poco más difícil de implementar que el expuesto hasta el momento. Sin embargo, en el entorno virtual de la prótesis de mano de este proyecto esta dificultad se obvia fácilmente eliminando en el experimento la fuerza de la gravedad.

En este trabajo se utilizó la curva resultante para estimar una función polinómica no lineal de la componente de la jacobiana buscada, como función de las velocidades de las articulaciones activas. Para ello se utilizaron los algoritmos de ajuste paramétrico de curvas disponibles en Matlab. Es de anotar que cuando la velocidad del efector final es cero, los resultados son indeterminados, por lo cual tales valores deben ser extrapolados.

1.2. Sub-actuación y auto-adaptabilidad

Por lo que se refiere al diseño de prótesis de mano antropomórficas, es muy importante buscar el mejor funcionamiento con el número mínimo de actuadores, lo

que lleva a la necesidad de mecanismos que puedan mover varias articulaciones con una sola articulación actuada. Así mismo, ha surgido la necesidad de mecanismos en dedos de prótesis que tengan la capacidad de adaptarse en el agarre de diferentes geometrías y tamaño de objetos sin requerir de intervención supervisada para dicha adaptación. Esta adaptabilidad se logra si el dispositivo permite varias posiciones posibles del punto de contacto con la misma postura común (sinergia) de la prótesis de mano, lo que lleva no solo a reducir el número de articulaciones actuadas, sino también a reducir la complejidad del control.

Desde una perspectiva mecánica, es muy difícil integrar un gran número de DoFs articulados con sus actuadores correspondientes para manipular una prótesis de mano antropomórfica con alta dimensionalidad. Además, el aumento de los DoFs actuados en las prótesis da como resultado una disminución de la fuerza de agarre en general y sistemas inestables. A raíz de lo anterior, surgen las prótesis de manos sub-actuadas (UA), que tienen menor número de actuadores en sus dedos que grados de libertad (DOFs). Los mecanismos de dedo tradicionales sub-actuados (UA) se dividen principalmente en dos: dedo UA acoplado y dedo UA autoadaptable. En los dedos UA acoplados, si la articulación proximal gira, la articulación distal girará en ángulo con una proporción fija de la articulación proximal. En los dedos UA auto-adaptables, un agarre adaptativo fiable se refiere a dos cuestiones básicas: ajustarse a la forma de los objetos de diferentes geometrías y tamaños, además de aplicar suficiente fuerza de agarre sobre los objetos sujetados, lo cual genera que el rango de agarre se agrande y sea más estable. Los mecanismos sub-actuados conducen a la auto-adaptabilidad en manos robóticas o protésicas a través de la conformación automática a la forma del objeto agarrado sin complejas estrategias de control [18].

1.2.1. Mecanismos sub-actuados acoplados

Un mecanismo sub-actuado es aquel mecanismo que posee menos actuadores que grados de libertad disponibles. Los movimientos para este tipo de mecanismos pueden ser generados por la acción de actuadores, conocida como actuación directa, o bien por otros elementos a los que se encuentren interconectados, llamada actuación indirecta.

Respecto a los mecanismos de barras, han sido utilizados para la actuación de dedos en prótesis de mano robóticas ya que permiten gobernar el movimiento de las articulaciones de los dedos utilizando un solo motor por dedo como los trabajos citados en [19], que poseen ventajas sobre otros mecanismos tales como tendones y poleas en cuanto a su robustez mecánica y transmisión de par.

Entre las ventajas de mecanismos de barras sub-actuados se tienen:

- Generan un movimiento que se percibe más natural respecto a las totalmente actuadas.
- Permiten el ahorro de energía, reducción de peso y tamaño además del dinero en la fabricación por medio de prototipado rápido como las impresoras 3D.

- Poseen ventajas relativas respecto a sus contrapartes basadas en tendones, en cuanto a robustez mecánica al desgaste de los materiales, el número de piezas, el par que se transmite a las articulaciones distales, y la menor complejidad del modelado cinemático y dinámico.

Por lo que se refiere a las desventajas, una de las principales es la posibilidad de bloqueo de las falanges medial o distal cuando el movimiento de la articulación actuada, la proximal, queda bloqueada por el contacto con objetos rígidos, lo cual impide un agarre estable del objeto. Además estos mecanismos sub-actuados conducen a modelos cinemáticos y dinámicos complejos matemáticamente, los cuales hacen un poco difícil su análisis e implementación en simulación.

Puede agregarse que los mecanismos de subactuación generalmente imponen restricciones cinemáticas que reducen el espacio de trabajo alcanzable por la mano, donde el mecanismo solo puede cumplir la trayectoria inicialmente diseñada después de fabricado. En esta dificultad particular, la subactuación presenta un reto para poder recrear todas o la mayoría de las posiciones que pueden alcanzar la yema de los dedos de la mano humana, lo que lleva al uso de complejas técnicas de control o al uso de elementos o componentes muy específicos como es el caso de los mecanismos tipo tendones, poleas o barras, o de una serie de conjunto de engranajes en serie y paralelo. El uso de estos mecanismos logra movimientos que reciben el nombre de acoplados, en el sentido que el movimiento de una articulación va acoplado al movimiento de una articulación precedente. Suele percibirse el movimiento de dedos con este tipo de mecanismos, como uno más natural que en el caso de manos totalmente actuadas.

La principal desventaja de estos mecanismos es consecuencia de su movimiento acoplado: si un objeto bloquea el movimiento de la articulación actuada, el movimiento de las articulaciones acopladas también queda bloqueado.

Frente a los mecanismos de subactuación, desde la literatura se pueden clasificar en poleas, tendones, barras, engranajes, cámaras con fluido, materiales con memoria y materiales eléctricos y térmicos; sin embargo, a pesar de tener excelentes resultados recreando el espacio de trabajo de un dedo, la desventaja del bloqueo del movimiento de las falanges por el contacto con objetos ha abierto el estudio a una solución conocida como la autoadaptación (SA, Self-Adaptive).

1.2.2. Mecanismos sub-actuados auto-adaptables

La auto adaptación en prótesis de mano robóticas, en el contexto de actuación por mecanismos de barras, se refiere a que los mecanismos permitan la adaptación mecánica del movimiento de los dedos de la mano, en el agarre de objetos desconocidos por el sistema.

Más específicamente, la auto-adaptación en el diseño de prótesis nace a partir de la necesidad de que el movimiento de las articulaciones medial o distal continúe a pesar del bloqueo de la articulación actuada, lo cual permite a la mano adaptarse por sí misma a la forma y tamaño de objetos desconocidos.

Además, es clave que los dedos de la mano puedan alcanzar el mayor número posible de puntos de contacto con el objeto de que se desea agarrar, a fin de poder lograr un agarre estable.

Solo hasta el año 2016 se han encontrado reportes de trabajos en la construcción de dedos para prótesis actuados por mecanismos de barras que incluyan algún mecanismo para superar el problema de bloqueo en el movimiento acoplado. Esta línea de trabajos, caen dentro del concepto que se ha denominado de forma genérica bajo el nombre de agarre autoadaptativo (self-adaptive), por la capacidad que tiene el mecanismo de adaptarse por sí mismo (sin intervención de complejas leyes de control) a objetos de formas y tamaños desconocidos por el sistema [19].

Otra de las definiciones para autoadaptación (SA, self-adaptive), en el contexto de prótesis de mano, es que es un método de solución que permite que la articulación actuada continúe generando y transmitiendo su movimiento a las otras articulaciones acopladas a ella a pesar de tener un objeto que la esté bloqueando.

En la revisión de literatura realizada en [19], como parte de su trabajo de maestría, clasificó tres métodos aplicados para la auto-adaptación: Sistema de dos o más de un actuador, sistema Pull On – Off o SA pasiva, y sistema de deslizamiento. En cada uno de estos métodos los mecanismos usados son principalmente tendones, poleas, engranajes, barras, cámara de fluido y, con un elemento común en todos, los resortes o materiales elásticos.

1.2.2.1. Autoadaptación con dos o más actuadores

Se entiende como modelo de autoadaptación con dos o más actuadores a aquella solución en la cual los mecanismos acoplados al actuador principal tienen una conexión paralela a otro u otros actuadores, de tal forma que cuando se presenta el bloqueo del actuador principal entra a funcionar el actuador o los actuadores de respaldo, para esto se utilizan diferentes técnicas de control y actuación. Una de las desventajas es que implica un costo energético adicional por el actuador en paralelo.

1.2.2.2. Autoadaptación sistema Pull On – Off o SA pasivo

El sistema Pull On – Off o SA pasivo, es similar a un botón de prendido y apagado, tal que cuando se presenta el bloqueo del actuador y mecanismo principal, el movimiento es transmitido a los siguientes mecanismos acoplados después de presionar o activar otro mecanismo pasivo que genera un nuevo movimiento, diferente al del actuador principal, para continuar con el agarre total del objeto. Una de las dificultades en este tipo de autoadaptación es la carga o par alcanzado, puesto que depende exclusivamente del generado por el mecanismo pasivo.

1.2.2.3. Autoadaptación sistema de deslizamiento

En estas soluciones se busca tener un solo actuador principal que, a pesar de existir el bloqueo, cuenta con un mecanismo en paralelo al principal el cual se acopla al actuador principal por deslizamiento en el momento de presentarse el bloqueo. El mecanismo más usado es el de poleas con correa, puesto que al bloquearse la correa principal existe una secundaria que se desliza y acopla al actuador principal y así se transmite el movimiento de este a los demás mecanismos. Esta solución es la más favorable para prótesis, pero las dificultades que se tienen es el tamaño de las piezas, y la estética, entre otras. También existe el método de deslizamiento con barras para realizar la transmisión del movimiento al deslizarse una barra secundaria y acoplarse al actuador principal. Algunos autores afirman que las barras son mejores para transmitir o multiplicar el par, dimensionalmente son estables y generan una mayor rigidez entre otras características.

1.3. Control de posición

En este trabajo se requiere que la mano virtual pueda realizar movimientos de flexión y extensión para poder agarrar objetos, dejando a los mecanismos de autoadaptación la tarea de que este movimiento lleve a la adaptación de los dedos a la forma del objeto.

La estrategia utilizada para este fin se deriva del gran beneficio obtenido de la estructura cinemática a utilizar basada en barras, donde cada dedo se convierte en un mecanismo de un solo grado de libertad efectivo y una sola articulación activa, por lo que el espacio de trabajo (cuando el mecanismo de auto adaptación no está en efecto) es único, tal que existe una relación uno a uno entre un ángulo de la articulación activa con una posición del efector final que para este caso se asume en la yema del dedo.

Entonces, resultan dos posibilidades. La primera, es mediante simulación o mediante el modelo geométrico, obtener la trayectoria de posición cartesiana del efector final para una trayectoria suave que barra toda la excursión viable físicamente de la articulación activa, y diseñar un controlador en el espacio cartesiano para seguimiento de esa trayectoria obtenida.

La segunda posibilidad, que como se verá es la más simple, es generar una trayectoria suave para barrer toda la excursión de movimiento de la articulación activa, y diseñar un controlador en el espacio articular para el seguimiento de esa trayectoria.

Así, la flexión y extensión de la mano se logra con un problema de seguimiento de una trayectoria que se barre ya sea en una dirección o en su sentido contrario.

1.3.1. Estrategias de control

Existen varias estrategias que pueden utilizarse para el seguimiento de una trayectoria de posición de un robot, éstas pueden clasificarse de la siguiente manera.

- No basadas en el modelo dinámico inverso del robot: control PID, control difuso (fuzzy), etc.
- Basadas en el modelo dinámico inverso del robot: control PID, por par calculado (CTC), control pasivo, control predictivo, control robusto, control adaptativo, control por modos deslizantes, etc.

En este trabajo se exploraron varias estrategias de control no basadas en el modelo. La razón de esta intención fue en dos vías: por una parte, como se verá mas adelante, la estructura de cadenas cerradas de los dedos del prototipo bajo estudio es bastante compleja, lo que lleva a una gran complejidad en la obtención de un modelo dinámico de forma analítica. Pero aún si tal modelo se hubiese podido calcular, la identificación de sus parámetros en una aplicación real sería una tarea realmente compleja. Las estrategias que se probaron en este trabajo fueron:

- Virtual Spring-Damper (VSD).
- PID en el espacio operacional.
- Control cascada sintonizado por modelo interno.

1.3.1.1. Sintonización de controladores PID (Para el esquema PID en espacio cartesiano)

En esta sección se describe el método de sintonía utilizado para el esquema de control PID en el espacio cartesiano. La intención de este apartado es que el lector comprenda el procedimiento seguido para esa sintonía a fin de justificar adecuadamente la no suficiencia de un método no basado en el modelo.

Ziegler y Nichols propusieron una serie de reglas para afinar controladores PID (ver ecuación 1.15) con base a una respuesta experimental.

$$u(t) = K_p \left(e(t) + \frac{1}{\tau_i} \int_{-\infty}^t e(t) dt + \tau_d \frac{de(t)}{dt} \right) \quad (1.15)$$

Donde $e(t)$ es la señal de error que entra al controlador y $u(t)$ es la acción de control que sale del controlador.

El método a utilizar es el basado en la ganancia crítica y periodo crítico, usado para sistemas que pueden tener oscilaciones sostenidas. Primero se eliminan los efectos de la parte integral y derivativa. Después, utilizando solo la ganancia K_p , se hace que el sistema tenga oscilaciones sostenidas (ver figura 1.8). El valor de ganancia con que se logre esto se llama ganancia crítica K_{cr} , que corresponde a un periodo crítico P_{cr} .

Con los valores de K_{cr} y P_{cr} se calculan los valores de los parámetros del control PID utilizando la tabla 1.1.

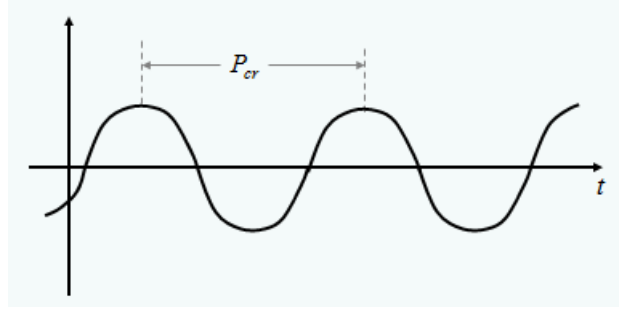


Figura 1.8: Oscilación Sostenida.

Tipo de controlador	K_p	τ_i	τ_d
P	$0,5K_{cr}$	∞	0
PI	$0,45K_{cr}$	$\frac{1}{2}P_{cr}$	0
PID	$0,6K_{cr}$	$0,5P_{cr}$	$0,125P_{cr}$

Tabla 1.1: Valores de sintonización para un controlador P, PI y PID a partir de K_{cr} y P_{cr} .

El controlador en tiempo continuo, se discretizó con un determinado tiempo de muestreo T y se implementa en un computador.

La ley de control más común es el del PID de la ecuación 1.16.

$$u(t) = K_p \left(e(t) + \frac{1}{\tau_i} \int_0^t e(\tau) d\tau + \tau_d \frac{de(t)}{dt} \right) \quad (1.16)$$

Se discretizará con la aproximación bilineal (trapezoidal o Tustin), en la cuál la derivada y la integral de la ecuación 1.16 se calculan mediante la ecuación 1.17 y 1.18 respectivamente.

$$\frac{de(t)}{dt} = \frac{e_k - e_{k-1}}{T} \quad (1.17)$$

$$\int_0^t e(\tau) d\tau = \sum_{i=1}^k \left(e(i-1) + \frac{e(i) - e(i-1)}{2} T \right) = \sum_{i=1}^k T \left(\frac{e_i + e_{i-1}}{2} \right) \quad (1.18)$$

La expresión del PID de la ecuación 1.16 al reemplazar las ecuaciones 1.17 y 1.18 es:

$$u_k = K_p \left(e_k + \frac{1}{\tau_i} \frac{T}{2} \sum_{i=1}^k (e_i + e_{i-1}) + \frac{\tau_d}{T} (e_k - e_{k-1}) \right) \quad (1.19)$$

Retrasando la expresión de la aproximación del PID de la ecuación 1.19:

$$u_{k-1} = K_p \left(e_{k-1} + \frac{1}{\tau_i} \frac{T}{2} \sum_{i=1}^{k-1} (e_i + e_{i-1}) + \frac{\tau_d}{T} (e_{k-1} - e_{k-2}) \right) \quad (1.20)$$

y restándosela a la original se obtiene:

$$\begin{aligned}
u_k - u_{k-1} &= q_0 e_k + q_1 e_{k-1} + q_2 e_{k-2} \\
q_0 &= K_p \left(1 + \frac{T}{2\tau_i} + \frac{\tau_d}{T} \right) \\
q_1 &= K_p \left(\frac{T}{2\tau_i} - 2\frac{\tau_d}{T} - 1 \right) \\
q_2 &= K_p \frac{\tau_d}{T}
\end{aligned}
\tag{1.21}$$

Aplicando la transformada Z se obtiene la función de transferencia del controlador:

$$C(z) = \frac{U(z)}{E(z)} = \frac{q_0 + z^{-1}q_1 + z^{-2}q_2}{1 - z^{-1}} \tag{1.22}$$

Finalmente se multiplica a la derecha de la ecuación 1.22 por $\frac{z^2}{z^2}$ y se obtiene:

$$\begin{aligned}
C(z) &= \frac{U(z)}{E(z)} = \frac{q_0 + q_1 z^{-1} + q_2 z^{-2}}{1 - z^{-1}} \frac{z^2}{z^2} \\
C(z) &= \frac{U(z)}{E(z)} = \frac{q_0 z^2 + q_1 z + q_2}{z^2 - z}
\end{aligned}
\tag{1.23}$$

Finalmente, en este capítulo se introdujo sobre algunos conceptos generales derivados de la teoría de mecanismos, robótica y control que permiten entender el desarrollo e implementación de la simulación de una prótesis virtual de mano en el gobierno de la flexión y extensión de sus dedos para el agarre de objetos.

Capítulo 2

ENTORNO VIRTUAL

El objetivo de este capítulo es entender los requerimientos y la implementación de un entorno virtual que permite la simulación del control de una prótesis de mano en el agarre de objetos gobernada a través de señales EMG.

2.1. Requerimientos

Con el fin de averiguar si las ventajas de una prótesis virtual de mano de 5 dedos con mecanismos de auto-adaptación podrían contribuir a mejorar la percepción de los usuarios en cuanto a la facilidad de gobierno de la misma a partir de señales EMG superficiales, se establecen algunos requerimientos en cuanto a software que pueden dividirse en: 1) software para la ingeniería de diseño mecánico 3D, 2) simulación dinámica y 3) programación e implementación de algoritmos de control. Ver figura 2.1. A lo anterior debe agregarse que la elección de cada software también debe ser acorde con las especificaciones del equipo de cómputo disponible en la realización del proyecto.

En cuanto al software de ingeniería de diseño mecánico 3D, es donde se concebirá el diseño desde cero de cada una de las piezas que componen el mecanismo de cada uno de los dedos de la prótesis de mano, así como el ensamblaje de todas ellas. Para lo anterior se requiere que sea un software de fácil manejo, que permita probar modelos en movimiento y que el prototipo final se pueda exportar en un tipo de formato aceptado por el software de simulación dinámica.

Por lo que se refiere al software de simulación dinámica, es donde se simula el proceso de la planta (mano) y se construye el entorno virtual. Debe permitir exportar el ensamblaje del diseño mecánico 3D de la prótesis de mano, además de establecer los tipos de articulaciones, realizar mediciones por medio de sensores a las entradas y salidas de la planta en cada instante de tiempo; así como de usar actuadores que permitan aplicar pares rotacionales a las articulaciones activas. Se requiere por lo tanto un motor de física que permita la simulación del efecto de la gravedad, el contacto entre piezas, el efecto de fuerzas entre cuerpos, la admisión de dinámicas tanto lineales como no lineales y la generación de un modelo dinámico de la planta (que se usará posteriormente en la etapa de control). En suma, un

software que permita la simulación del agarre de objetos de diferente forma y tamaño por parte de una prótesis de mano, bajo la cual esté presente la física del mundo real.

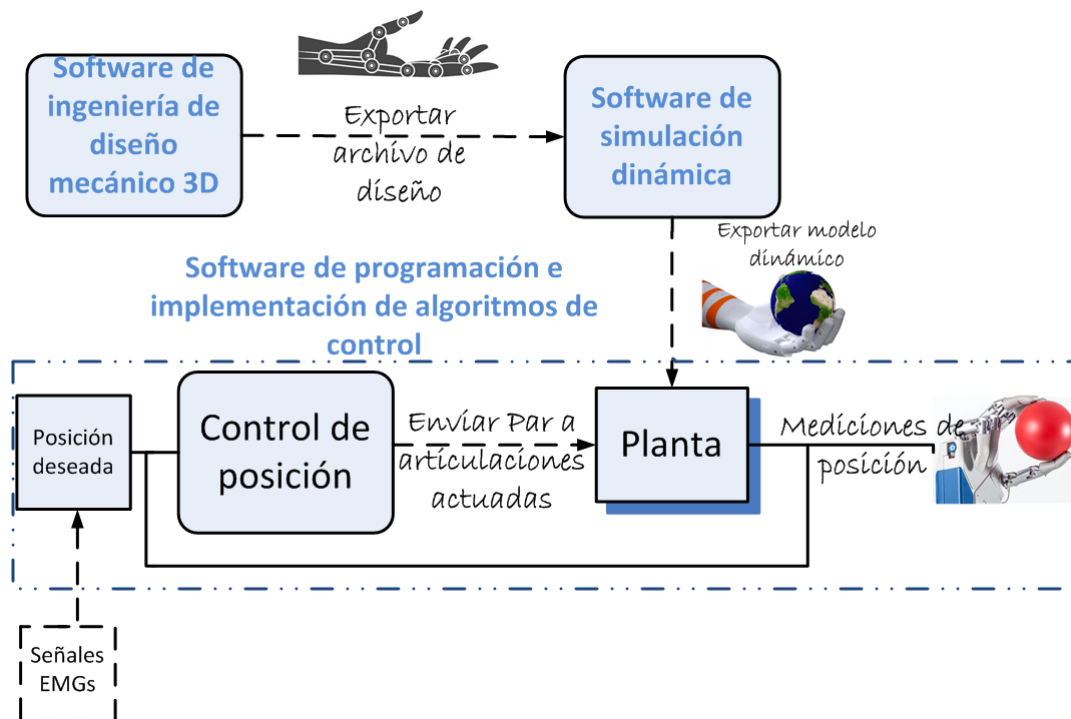


Figura 2.1: Software requerido para el proyecto.

Ahora bien, existe un requerimiento especial entre el software de simulación dinámica y el de programación, esto es, que el primero permita exportar el modelo dinámico de la planta y una comunicación mutua en tiempo real con el segundo, de tal forma que la prótesis de mano pueda tener como entrada del controlador los pares de las articulaciones actuadas, y como salidas las posiciones angulares y/o cartesianas según lo requiera el esquema de control.

2.2. Elección de software

Una vez claros los requerimientos, se procede a realizar una búsqueda de software. En la tabla 2.1 y 2.2 se observan algunas opciones para cada uno de los softwares mencionados anteriormente.

SOFTWARE DE INGENIERÍA DE DISEÑO MECÁNICO 3D			
SOLIDWORKS	CATIA	Solid Edge	FreeCAD
<p>Convierte una geometría de boceto en geometría 3D. Algunas de sus funcionalidades le permiten agregar y eliminar material de varias maneras diferentes, incluido el uso de Extrudes, Revolves, Filletsy y Chamfers. Además permite el ensamblaje de animación. Algunos tipos de archivo de exportación son: ACIS (.sat), Adobe Illustrator (.ai), DFX/DWG (.dfx, .dwg, IGES (.iges, .igs), Parasolid (.x_t,.x_b), Step (.step) y STL (.stl).</p>	<p>Ofrece herramientas de modelado 3D / sólido(herramientas de edición paramétrica, de forma libre, proceso de edición simétrica, etc.), herramientas de renderizado, herramientas básicas de dibujo e importación y exportación de archivos. Algunos formatos de exportación son: .stl, .igs, .model, .stp, 3dmap, 3dxml, .cgr, .wrl, etc.</p>	<p>Sistema integral de diseño 3D con tecnología síncrona para acelerar el diseño. Funcionalidades: Traductores 2D y 3D, dibujos automatizados, ensambles despiezados y de animación, ensamblaje básico, parte básico, simulación de movimiento básico, diseño de superficie, chapa, soldaduras, diseño de marco, simulación express, referencia de ingeniería, fotorrealismo, biblioteca de maquinaria, Simulación de Insight, diseño de arnés de cables, XpresRoute.</p>	<p>FreeCAD presenta un entorno de trabajo similar a CATIA, SolidWorks y SolidEdge. Utiliza técnicas de modelado paramétrico (utiliza parámetros para definir sus límites o acciones) y está provisto de una arquitectura de software modular. Además del propio formato de archivo de FreeCAD, pueden manejarse los siguientes formatos de archivo: DXF, SVG (Scalable Vector Graphics), STEP, IGES, STL (STereoLithography), OBJ (Wavefront), DAE (Collada), SCAD (OpenSCAD), IV (Inventor), IFC, Format, FCStd, FCMat, FCMacro, STPZ , BREP, DXF, DWG, SVG , OCA, DAE, y BMS.</p>

Tabla 2.1: Software para ingeniería de diseño mecánico 3D disponible.

SOFTWARE SIMULACIÓN DINÁMICA MULTICUERPO		
ADAMS	ANSYS	Vrep
<p>Es un software de simulación dinámica multicuerpo equipado con los solucionadores numéricos Fortran y C++. Permite la creación o importación de geometría de sólidos 3D, refinamiento del modelo con flexibilidad de piezas, sistemas de control automático, fricción y deslizamiento de las juntas. Se permite gestionar las complejas interacciones entre disciplinas, incluidos el movimiento, las estructuras, el rendimiento y el control para optimizar los diseños de productos en cuanto a rendimiento, seguridad y comodidad. Los módulos opcionales disponibles con Adams permiten a los usuarios integrar componentes mecánicos, sistemas neumáticos, hidráulicos, electrónicos y sistemas de control para construir y probar prototipos virtuales que reflejen con precisión las interacciones entre estos subsistemas. Cuenta con un módulo adicional que permite el modelado y simulación del sistema de control de lazo cerrado SISO Y MIMO, está disponible a través de ADAMS / Controles. Archivos de importación: Parasolid(.xmt.txt, .x.t, .xmt_bin, .x_b), CATIAV4(.model, .exp, .session), CATIAV5(.CATPart, CATProduct), IGES(.igs, .iges) y STEP(.stp, .step)</p>	<p>Las competencias o capacidades de este programa están enfocadas a las soluciones estructurales en cuanto a: estática, dinámica, propiedades no lineales del material, vibración, modal y análisis armónico, fatiga, térmico; contacto no lineal avanzado y otras capacidades de análisis para sistemas mecánicos y de materiales. Algunas herramientas son: Herramientas de simulación de fluidos (análisis de flujo, propagación de fuego y humo, transporte de agentes químicos y biológicos, transferencia de calor, calidad del aire y otros problemas de flujo), multifísica (electromagnética, acústico; interacción de estructura fluida), dinámica explícita (impacto de objetos y vehículos en edificios, deformación de la estructura bajo explosión), soluciones especializadas (solución dedicada para el modelado de flujo de aire, enfriamiento del centro de datos, ingeniería civil) y soluciones de simulación de ingeniería para el entorno de construcción de la Industria. ANSYS ofrece un paquete completo de software que abarca todo el rango de la física. Archivos de importación: IGES(.iges), Parasolid (.x.t o .xmt.txt) Parasolid (* .x.t, * .xmt.txt, * .x_b, * .xmt_bin).</p>	<p>Es un simulador de robots, con entorno de desarrollo integrado, se basa en una arquitectura de control distribuido: cada objeto / modelo puede controlarse individualmente a través de un script embebido, un complemento, un nodo ROS o BlueZero, un cliente API remoto o una solución personalizada. Los controladores se pueden escribir en C / C ++, Python, Java, Lua, Matlab u Octave. Tiene 4 motores de física (Bullet Physics, ODE, Newton y Vortex Dynamics) para cálculos de dinámica rápidos y personalizables, para simular interacciones físicas y de objetos del mundo real (respuestas de colisión, agarre, etc). Permite cálculos de cinemática inversa y directa para cualquier tipo de mecanismo. Se admiten los siguientes formatos de archivo: URDF, COLLADA, DFX,3DS, OBJ, STL, (ASCII y binario). V-REP utiliza mallas triangulares para describir y mostrar formas, por lo tanto solo importará formatos que describan objetos como mallas triangulares. Sin embargo, si desea importar objetos descritos como superficies paramétricas (por ejemplo: IGES, STEP, etc), primero deberá convertir el archivo a un formato de malla triangular apropiado.</p>

Tabla 2.2: Software simulación dinámica disponible.

En la tabla 2.1 presento el software de ingeniería de diseño mecánico 3D más utilizados y disponibles en el mercado. El software elegido fue Solidworks versión 2017, elección que hizo parte del trabajo de grado de maestría titulada: "SÍNTESIS DE UN MECANISMO DE BARRAS PARA ACTUACIÓN DE UNA PRÓTESIS DE MANO SUBACTUADA ANTROPOMÓRFICA CON CAPACIDAD DE AUTOADAPTACIÓN " del ingeniero Iván Gustavo Paez Chingal, en la cuál se propone un mecanismo de barras para la actuación de una prótesis de mano robótica subactuada con capacidad de autoadaptación en el agarre de objetos, tal que se logre antropomorfismo en el tamaño de la mano, movimiento y espacio de trabajo de las yemas de los dedos. Dicho mecanismo para cada uno de los dedos de la prótesis de mano, es el punto de partida para este proyecto de tesis de pregrado.

De la tabla 2.2 se ha elegido a Adams versión 2016 como el software a utilizar en la simulación dinámica, principalmente porque este programa cuenta con un módulo llamado Adams/Controls el cual está muy bien integrado en el Simulink de MathWorks mediante algunas S-function. Este módulo permite obtener el modelo dinámico de la prótesis de mano (planta) e incluir este modelo como un bloque en simulink para poder realizar el control de la planta. A continuación una explicación de cómo lo anterior puede ser posible.

El software de Adams, trabaja generando un modelo dinámico de la planta, que se construye a partir de un sistema mecánico previamente diseñado y editado en Adams. Este modelo representa y simula la dinámica presente en el sistema y proporciona una herramienta de simulación gráfica valiosa para el análisis del comportamiento de los sistemas de control y de la sensorica presente. Los sensores que incorpora Adams hacen posible que el sistema de control trabaje en lazo cerrado, lo cual genera un incremento en la precisión para seguir una señal de entrada, reduce la sensibilidad a variaciones en el modelo y el efecto de no linealidades.

Ahora bien, en cuanto a la comunicación entre Adams y simulink, un ciclo cerrado entre Simulink y Adams / Controls hace que la simulación de sistemas no LTI (Linear time-invariant theory, comúnmente conocido como sistema LTI system) sea muy simple. Se modela un modelo de planta variable no lineal en ADAMS / Controls y su comportamiento se informa a Simulink a través de una comunicación TCP / IP como realimentación, analizándolo por algún controlador dentro de Simulink y actuando a través de actuadores en la planta de ADAMS / Controls en el mismo esquema de comunicación. También a través del mecanismo de exportación de control, ADAMS / Controls puede proporcionar a Control System Toolbox de MATLAB un modelo de espacio de estado del sistema bajo estudio que también podría ser utilizado más adelante para el diseño del controlador.

En la figura 2.2 se observa la representación en bloques de la planta generada por Adams, cuya entrada es la variable manipulada que permite establecer en un valor específico la variable controlada. Por otro lado, en la figura 2.3 se puede observar, a modo general, un sistema de control en lazo cerrado utilizando el bloque de una planta generado por Adams/controls.

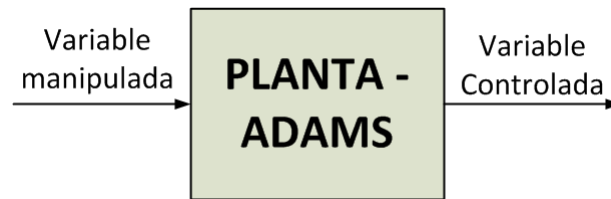


Figura 2.2: Representación de entrada salida de una planta en Adams.

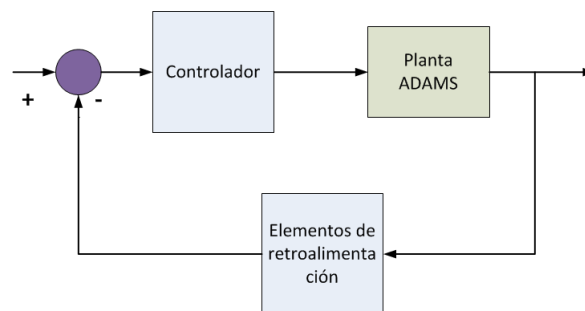


Figura 2.3: Representación de un sistema de control en lazo cerrado usando el bloque de la planta generado por Adams.

Para obtener el bloque de la planta de la figura 2.3, Adams genera un archivo en Simulink en donde se encuentra el bloque que simula la dinámica de la planta (Ver figura 2.4).

En dicho archivo encuentra:

- La S-Function, quien representa el modelo ADAMS no lineal.
- El bloque `adams_sub` (Ver figura 2.5), contiene la S-Function, pero también crea varias variables útiles en la comunicación entre Adams y MATLAB.
- El bloque State-Space representa un modelo de ADAMS linealizado.

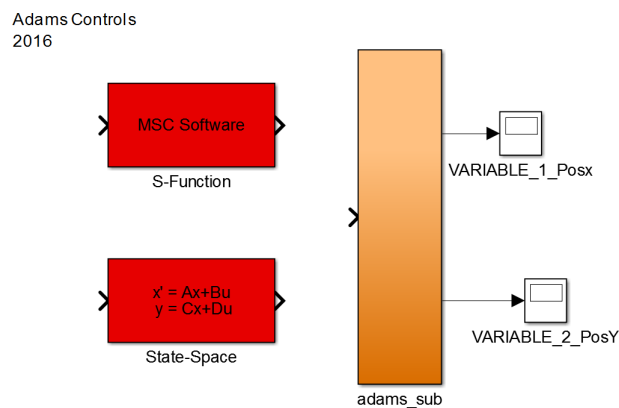


Figura 2.4: Archivo `Adams_sys` que simula la dinámica de la planta.

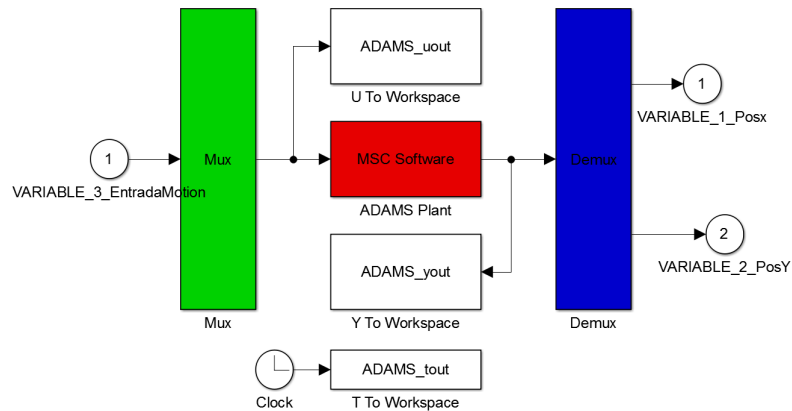


Figura 2.5: Contenido en el interior del bloque de adams_sub

En el apéndice A se encuentran más en detalle los procesos para la exportación del diseño CAD en SOLIDWORKS; la creación, importación, ensamblaje y desarrollo de una prótesis virtual de mano con interacción de objetos; y finalmente la exportación y comunicación del modelo dinámico para la prótesis de mano virtual desde Adams a Simulink.

2.3. Entorno virtual diseñado

El entorno virtual para la prótesis de mano de este proyecto tiene como objetivo visualizar la simulación y control de una prótesis de mano en el proceso de agarre de objetos de diferente forma y tamaño, a partir de las señales EMGs de un grupo de usuarios. Para lo anterior es necesario elegir objetos con geometrías que permitan comprobar que el contacto del agarre final entre la mano y los objetos es autoadaptable.

La selección del conjunto de objetos que la prótesis de mano virtual manipulará, fue basada en [20], en el cual se presenta un conjunto estandar de objetos para la investigación de la manipulación robótica y la evaluación del rendimiento, así como un marco para los protocolos de tareas estándar que realiza una prótesis de mano real. Dicha selección se basa en una encuesta de los objetos más comunes utilizados en el campo de la robótica, así como en la bibliografía sobre prótesis y rehabilitación (en la que se desarrollan procedimientos para evaluar las capacidades de manipulación de los pacientes). En este proyecto se replicarán los objetos que se pueden aproximar por formas geométricas simples, con el fin de construirlos dimensionalmente en el software de simulación dinámica.

La simulación del entorno virtual se realiza en el software de simulación dinámica multicuerpo Adams versión 2016, en el cual se observa la prótesis de mano y una mesa. El tamaño de la mesa es: 400 *mm* de longitud, 200 *mm* de altura y 300 *mm* de profundidad. Para este proyecto se proponen las siguientes figuras en dos tamaños:

- Esfera de radio 20 *mm* (Tamaño de pelota de Ping Pong) y 40*mm*.

- Cubo de 57 mm (Tamaño de cubo de Rubik) y 47 mm .

Las figuras de la prótesis de mano con los objetos mencionados anteriormente se observan a continuación. (Ver figuras 2.6, 2.7, 2.8 y 2.9)

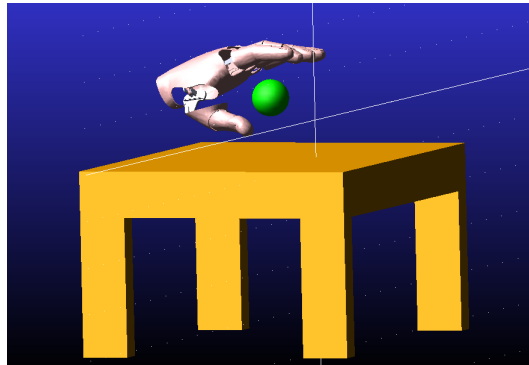


Figura 2.6: Entorno virtual de la prótesis de mano con esfera de 20 mm .

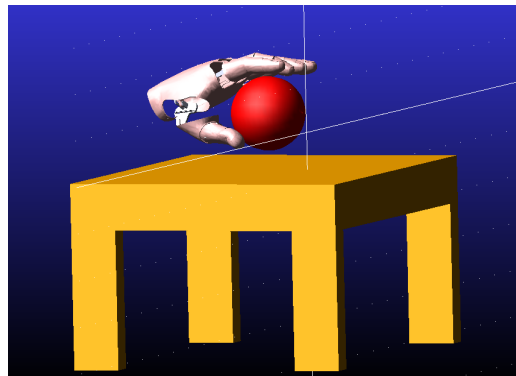


Figura 2.7: Entorno virtual de la prótesis de mano con esfera de 40 mm .

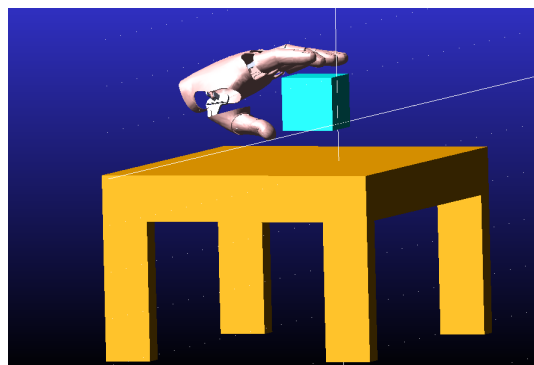


Figura 2.8: Entorno virtual de la prótesis de mano con Cubo de 57 mm .

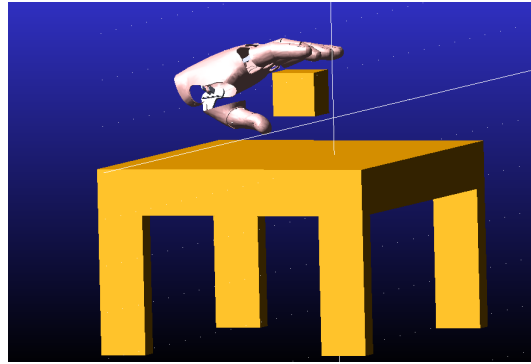


Figura 2.9: Entorno virtual de la prótesis de mano con Cubo de de 47 *mm*.

En las figuras 2.10 y 2.11 se observa la prótesis de mano controlada al realizar el agarre de una esfera de radio 40 *mm*.

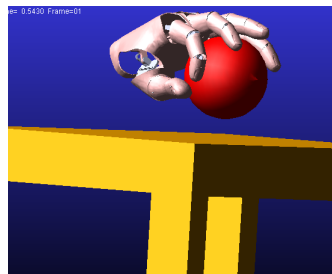


Figura 2.10: Entorno virtual de la prótesis de mano agarrando esfera de 40 *mm*.

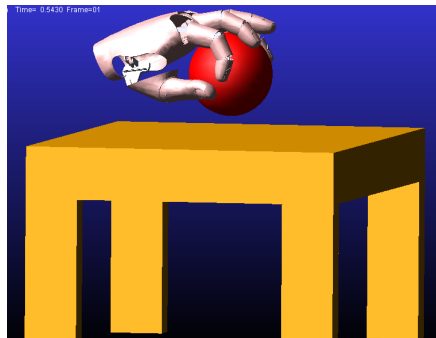


Figura 2.11: Entorno virtual de la prótesis de mano agarrando esfera de 40 *mm*.

2.4. Diseño CAD

El tema a desarrollar en esta sección es el correspondiente al diseño mecánico de la prótesis virtual de mano, el cuál tiene unos requerimientos de diseño que deben cumplirse para poder alcanzar el objetivo general de esta tesis de pregrado. Este es, averiguar si las ventajas de una prótesis sub-actuada con mecanismos de auto-adaptación podrían contribuir a mejorar la percepción de los usuarios en cuanto a la facilidad de gobierno de la prótesis a partir de señales EMG superficiales.

Por consiguiente, la obtención de una prótesis de mano virtual sub-actuada con capacidad de agarre auto adaptativo de objetos se obtiene desde el diseño del mecanismo que la va a componer. A partir de esta premisa se obtienen dos requerimientos de diseño, primero que el mecanismo sea sub-actuado y segundo que sea auto-adaptable.

La satisfacción de este requerimiento se obtiene tomando el producto final de la tesis de maestría del ingeniero Iván Gustavo Paez Chingal titulada: "Síntesis de un mecanismo de barras para actuación de una prótesis de mano subactuada antropomórfica con capacidad de autoadaptación"; cuyo objetivo global, mencionado en la sección anterior, se logra estableciendo la estructura cinemática de un mecanismo de barras para una prótesis de mano robótica subactuada con capacidad de autoadaptación en el agarre de objetos, que permitan lograr antropomorfismo en el tamaño de la mano, movimiento y espacio de trabajo de la yema de los dedos.

2.4.1. Sub-actuación y auto adaptabilidad de la prótesis

Por lo que se refiere al diseño de la prótesis de mano virtual, está compuesta por 5 dedos. Cada dedo es sub-actuado, auto-adaptable y cuenta con una sola articulación actuada. Existen dos diseños mecánicos para los dedos, el primero de ellos (Mecanismo de dedo A) es el usado para 4 de los 5 dedos y el segundo de ellos (Mecanismo de dedo B) es el usado para el dedo restante.

El mecanismo de dedo A es el usado para el índice, medio, anular y meñique. Este mecanismo permite la flexión-extensión y no la abducción-aducción de los dedos.

El mecanismo de dedo B es el usado para el dedo pulgar, permite tanto la flexión-extensión como la abducción-aducción del dedo.

2.4.1.1. Mecanismo de dedo A

El diseño de dedo se divide, como en la anatomía osteoarticular de los dedos, en falanges proximal, medial y distal. Cada una de las falanges cuenta con mecanismos de barras que permiten la transmisión de movimiento de una falange a otra hasta llegar a la falange distal donde se encuentra la yema del dedo (efector final).

Los mecanismos de las falanges proximal y medial cuentan con dos mecanismos primordiales: mecanismo de sub-actuación (SA) y mecanismo de auto-adaptabilidad (AA). El mecanismo SA y AA no funcionan simultáneamente, existe un caso para el cual solamente uno de ellos está activo.

El mecanismo SA está activo mientras que la falange proximal o la falange medial no se hayan bloqueado al hacer contacto con algún objeto. Esto es debido a que en los mecanismos sub-actuados si un objeto bloquea el movimiento de la articulación actuada, el movimiento de las articulaciones acopladas también queda bloqueado. Por lo tanto, al bloquearse algún mecanismo de las falanges, en consecuencia del

contacto de un objeto que se desea agarrar, este mecanismo por sí solo no es capaz de continuar el movimiento, es entonces cuando el mecanismo AA entra en funcionamiento. En la figura 2.12 se observa en qué casos se activa alguno de los dos tipos de mecanismos que componen el modelo de dedo A.

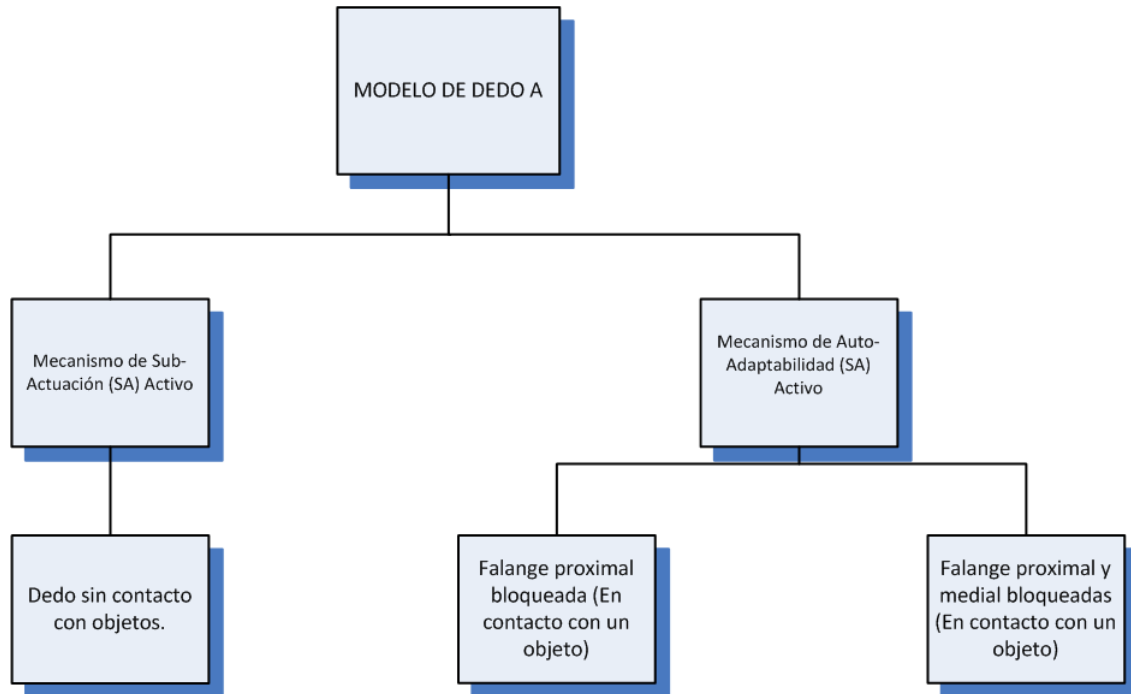


Figura 2.12: Casos en los que entra en funcionamiento alguno de los mecanismos que componen el modelo de dedo A.

Al llegar a este punto, se necesita que los mecanismos de las falanges proximal, medial o distal continúen generando y transmitiendo movimiento a pesar del bloqueo, lo cual permitirá al dedo adaptarse por sí mismo a la forma y tamaño del objeto que se desea agarrar. Lo anterior se logra mediante el método de auto-adaptación con deslizamiento de barras, en el cual la transmisión del movimiento continúa al deslizarse una barra secundaria (por medio de la articulación de perno en un ranura Figura 1.5) que permite acoplarse al actuador y mecanismo principal.

A continuación se presenta el diagrama cinemático donde se visualizarán con claridad el mecanismo SA y AA presentes en las falanges proximal, medial y/o distal.

2.4.1.1.1. Diagrama cinemático

En la figura 2.13 se presenta el diseño CAD del dedo índice en solidWorks y en la figura 2.14 el diagrama cinemático del mecanismo.

El primer paso en la elaboración del diagrama cinemático es identificar la bancada del mecanismo y notarlo como el eslabón número 1, el cuál simboliza la tierra del

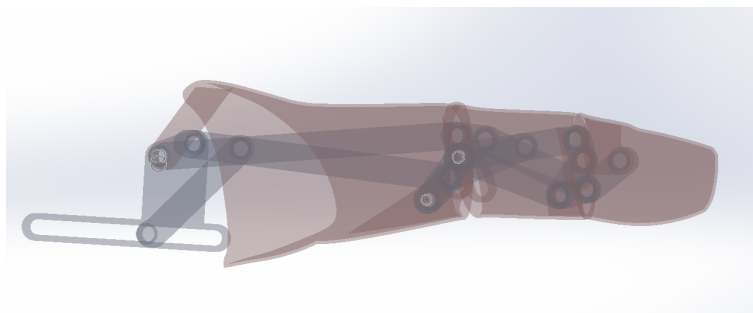


Figura 2.13: Diseño CAD del mecanismo de dedo A con motor lineal, dedo índice.

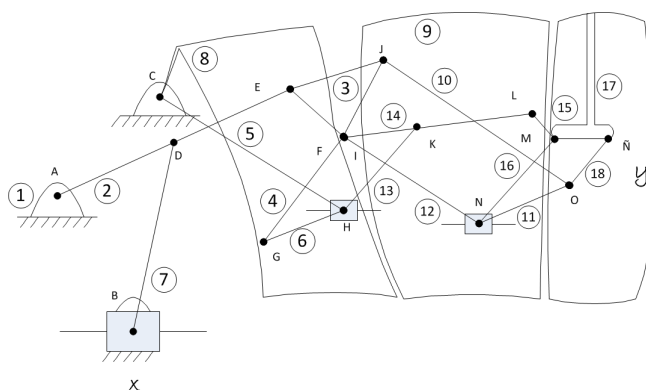


Figura 2.14: Diagrama cinemático modelo de dedo A para actuación con motor lineal.

mismo, esta es: la parte fija del dedo en la palma. Como se observa en la figura 2.14, la unión de perno en una ranura se encuentra en la bancada, y es la unión que será actuada. Por consiguiente, el movimiento de los demás eslabones se determinará en relación con el eslabón número 1. A continuación se identifican los demás eslabones del dedo, entre los más importantes y de fácil visualización se encuentran las carcasas de las falanges proximal, medial y distal, las cuales están unidas mediante articulaciones de revoluta y de perno en una ranura a los demás eslabones sencillos que componen el mecanismo. A continuación se enumeran los eslabones que componen el mecanismo y se denotan por letras las uniones entre ellos. Finalmente se identifican los puntos de interés, el primero es el movimiento del eslabón 7 a lo largo de la ranura, al cual se le nombrará punto de interés X y es el que permitirá transmitir movimiento al resto del mecanismo; el segundo es el movimiento que se busca determinar a partir del primero, este es en la yema del dedo, al cual se le denotará como punto de interés Y.

Profundizando un poco en la visualización del mecanismo, su única articulación actuada es posible gracias a un motor lineal que estará ubicado en la bancada del mecanismo (palma de la prótesis de mano) y unido al eslabón 7 por medio de la articulación B. En la figura 2.14, el eslabón 7 está conectado a la tierra del mecanismo por medio de la articulación B (perno en una ranura). El actuador transmite el movimiento a la falange proximal (eslabon 8) gracias al eslabón 7, el cual tiene la función de convertir su traslación, en uno de sus extremos, a una rotación al eslabón 2, el cuál está situado en la falange proximal del dedo. Sin embargo, en este

trabajo de grado por efectos de simulación en Adams, tanto el motor lineal como el eslabón 7 han sido omitidos y en su lugar se ha simulado la articulación *C* como la uncia articulación de revoluta o de perno actuada mediante un motor rotacional. Es de notar que el hecho de haber eliminado la articulación de perno *D* y la articulación de perno en una ranura *B*, no interfiere de ninguna manera en el funcionamiento del mecanismo. El diagrama cinemático del dedo con este pequeño cambio se observa en la figura 2.15 y el diseño Cad se observa en la figura 2.16.

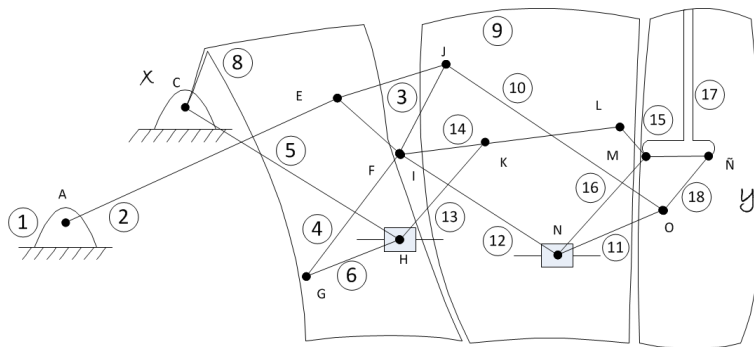


Figura 2.15: Diagrama cinemático modelo de dedo A para actuación con motor rotacional.

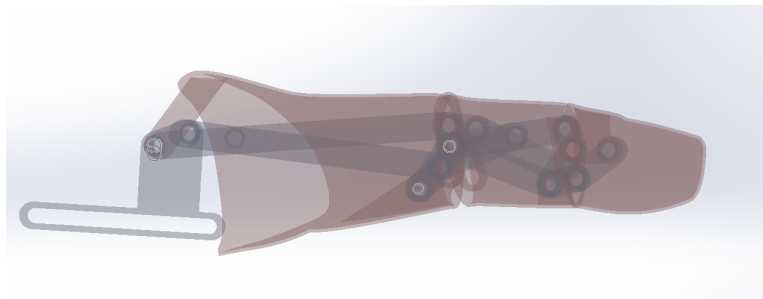


Figura 2.16: Diseño CAD del mecanismo de dedo A con motor rotacional, dedo índice.

En la figura 2.17 y 2.18 se observa en color rojo el mecanismo SA y en azul el mecanismo AA, los cuales no se modifican al cambiar un motor lineal por uno rotacional.

En cuanto a los grados de libertad o movilidad del mecanismo del tipo de dedo A, se debe recordar que el mecanismo tiene dos comportamientos, el primero es cuando está activo el mecanismo SA (antes del contacto con un objeto) y el segundo cuando el mecanismo AA está activo (durante el contacto con un objeto). En el segundo caso es cuando las articulaciones que antes estaban definidas como de perno (unión de orden inferior) ahora pasan a ser de perno en una ranura (unión de orden superior). Por lo anterior, al aplicar la ecuación de Grueber vista en la ecuación 1.2, para hallar los grados de libertad, se tendrá un cambio en el número total

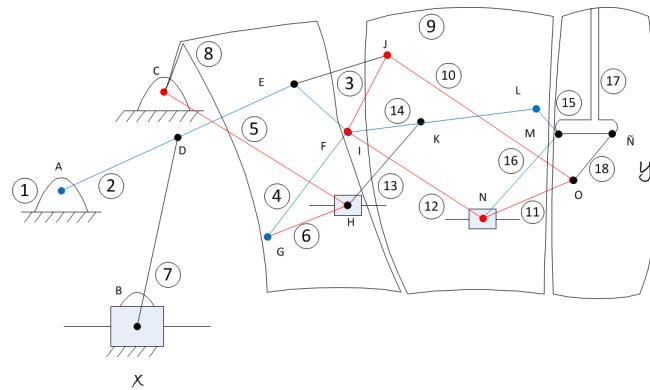


Figura 2.17: Mecanismos de SA y AA del modelo de dedo A para actuación con motor lineal.

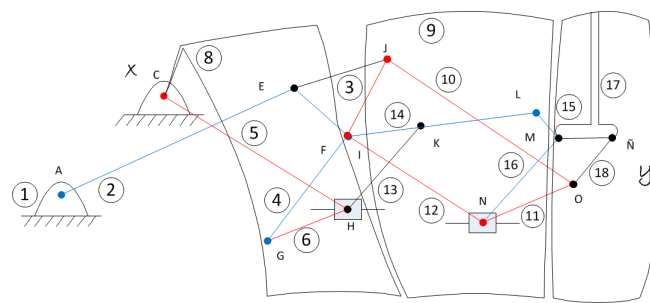


Figura 2.18: Mecanismos de SA y AA del modelo de dedo A para actuación con motor rotacional.

de uniones principales j_p y uniones de orden superior j_h . Por consiguiente se realizará un diagrama cinemático para los dos casos y se hallará el número de grados de libertad. El resultado del número de grados de libertad para cada caso debe ser de uno.

Primer caso: Mecanismo SA Activo

En el modelo de dedo A con actuación de un motor lineal (Ver figura 2.14), cuando el mecanismo SA está activo, dos acontecimientos importantes ocurren. Primero, solo una de las cinco articulaciones de perno en una ranura (unión de orden superior) está activa, las restantes se comportan como articulaciones de perno (unión de orden inferior o unión principal). Segundo, el comportamiento de algunos eslabones conectados entre sí se convierte en un solo eslabón con un número específico de juntas. Por lo anterior, se pasa del diagrama cinemático del mecanismo de la figura 2.14 al de la figura 2.19.

Las 3 articulaciones de perno en una ranura de la figura 2.14 son B, H y N, de las cuáles solo estará en funcionamiento la articulación B (ahora llamada articulación C en la figura 2.19). Por otra parte, los eslabones 2, 3, 4, y 8 de la figura 2.14 formarán el eslabón número 3 (en rojo) de la figura 2.19, y los eslabones 14, 15, 16 y 12 de la figura 2.14 formarán el eslabón número 7 (en verde) de la figura 2.19. En este nuevo diagrama cinemático, el número de eslabones n es igual a 11, el número total de uniones principales j_p es 14 y número total de uniones de orden

superior (unión de perno en una ranura específicamente) j_h es 1. Por lo tanto se tiene: $M = 3(n - 1) - 2j_p - j_h = 3(11 - 1) - 2(14) - 1 = 1$.

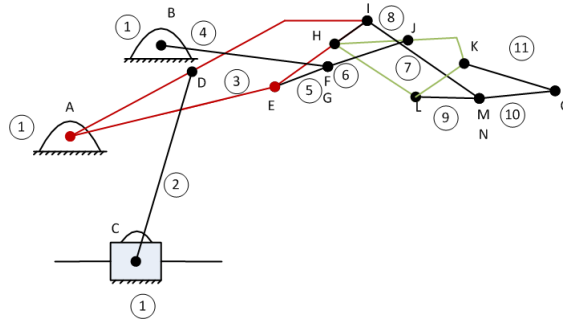


Figura 2.19: Diagrama cinemático modelo de dedo A para actuación con motor lineal antes del contacto con un objeto.

De forma similar, en el modelo de dedo A con actuación de un motor rotacional (Ver figura 2.15), cuando el mecanismo SA está activo, ocurren los mismos dos acontecimientos importantes nombrados arriba. La única diferencia del modelo de dedo A con actuación de un motor lineal, es que desaparece la articulación actuada de perno en una ranura B de la figura 2.14 y en su lugar es actuada la articulación de perno C en la figura 2.15. Por lo tanto se pasa del diagrama cinemático del mecanismo de la figura 2.15 al de la figura 2.20.

Las 2 articulaciones de perno en una ranura de la figura 2.15 son H y N, de las cuáles ninguna estará activa. Por otra parte, los eslabones 2, 3, 4, y 8 de la figura 2.15 formarán el eslabón número 2 (en rojo) de la figura 2.20, y los eslabones 14, 15, 16 y 12 de la figura 2.15 formarán el eslabón número 6 (en verde) de la figura 2.20. En este nuevo diagrama cinemático, el número de eslabones n es igual a 10, el número total de uniones principales j_p es 13 y número total de uniones de orden superior j_h es 0. Por lo tanto se tiene: $M = 3(n - 1) - 2j_p - j_h = 3(10 - 1) - 2(13) - 0 = 1$.

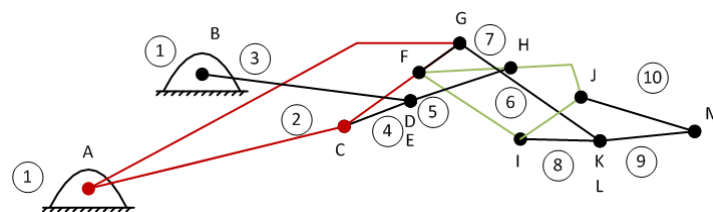


Figura 2.20: Diagrama cinemático modelo de dedo A para actuación con motor rotacional antes del contacto con un objeto.

Segundo caso: Mecanismo AA Activo

En el modelo de dedo A con actuación de un motor lineal (Ver figura 2.14), cuando el mecanismo AA está activo, se había dicho que las articulaciones que antes es-

taban definidas como de perno (unión de orden inferior) pasan a ser de perno en una ranura (unión de orden superior). Además de lo anterior existen dos comportamientos diferentes cuando el mecanismo AA está activo. El primero es cuando la falange proximal hace contacto con un objeto y y la falange medial no (Caso AA-FP). El segundo es cuando tanto la falange proximal como la medial contactan con un objeto (Caso AA-FP-FM). Para cada uno de los dos casos mencionados existe un comportamiento diferente de los eslabones y las juntas, por consiguiente se tienen dos diagramas cinemáticos totalmente distintos.

Primero, para el caso AA-FP solo dos de las tres articulaciones de perno en una ranura (unión de orden superior) están activas, la restante se comporta como articulación de perno (unión de orden inferior o unión principal). Además, el comportamiento de algunos eslabones conectados entre si se convierten en un solo eslabón con un número específico de juntas. Por lo anterior, se pasa del el diagrama cinemático del mecanismo de la figura 2.14 al de la figura 2.21.

Las 3 articulaciones de perno en una ranura de la figura 2.14 son B, H y N, de las cuáles solo estará en funcionamiento la articulación B y H (ahora llamadas articulación C y H en la figura 2.21 respectivamente). Por otra parte, los eslabones 14, 15, 16 y 12 de la figura 2.14 formarán el eslabón numero 6 (en verde) de la figura 2.21, y los eslabones 5, 6, 4 y 8 de la figura 2.14 formarán parte del eslabón tierra o bancada número 1 (en líneas punteadas) de la figura 2.21 debido al bloqueo que se genera en estos eslabones al contactar con un objeto. En este nuevo diagrama cinemático, el numero de eslabones n es igual a 10, el número total de uniones principales j_p es 12 y número total de uniones de orden superior (unión de perno en una ranura específicamente) j_h es 2. Por lo tanto se tiene: $M = 3(n - 1) - 2j_p - j_h = 3(10 - 1) - 2(12) - 2 = 1$.

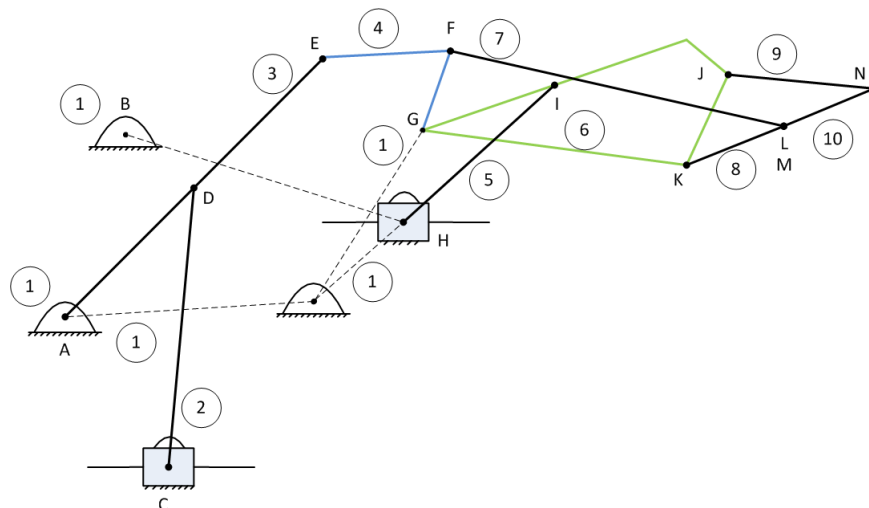


Figura 2.21: Diagrama cinemático modelo de dedo A para actuación con motor lineal, caso AA-FP.

De forma similar, en el modelo de dedo A pero con actuación de un motor rotacional (Ver figura 2.15), para el primer caso AA-FP ocurren los mismos acontecimientos importantes nombrados con la actuación de un motor lineal y se difiere únicamente en que desaparece la articulación actuada de perno en una ranura B de la figura

2.14 y en su lugar es actuada la articulación de perno C en la figura 2.15. Tomando en cuenta todos estos cambios, se pasa del diagrama cinemático del mecanismo de la figura 2.15 al de la figura 2.22.

Las 2 articulaciones de perno en una ranura de la figura 2.15 son H y N, de las cuáles solo H estará activa (Ahora llamada F en la figura 2.22). Por otra parte, los eslabones 14, 15, 16 y 12 de la figura 2.15 formarán el eslabón numero 6 (en verde) de la figura 2.22, y los eslabones 5, 6, 4 y 8 de la figura 2.15 formarán parte del eslabón tierra o bancada número 1 (en líneas punteadas) de la figura 2.22 debido al bloqueo que se genera en estos eslabones al contactar con un objeto. En este nuevo diagrama cinemático, el numero de eslabones n es igual a 9, el número total de uniones principales j_p es 11 y número total de uniones de orden superior (unión de perno en una ranura específicamente) j_h es 1. Por lo tanto se tiene: $M = 3(n - 1) - 2j_p - j_h = 3(9 - 1) - 2(11) - 1 = 1$.

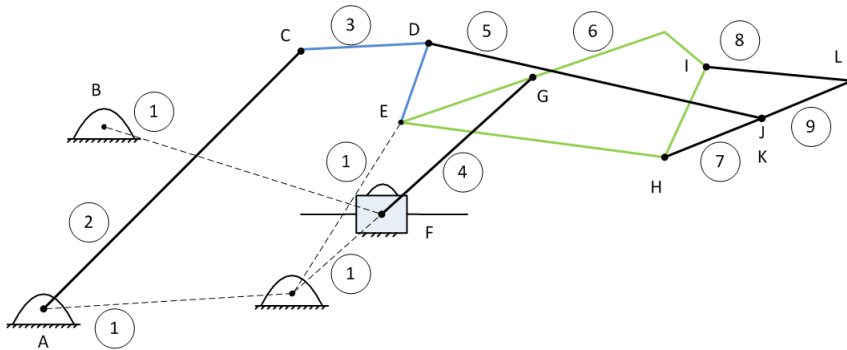


Figura 2.22: Diagrama cinemático modelo de dedo A para actuación con motor rotacional, caso AA-FP.

Segundo, para el caso AA-FP-FM tres de las tres articulaciones de perno en una ranura (unión de orden superior) están activas. Además, el comportamiento de algunos eslabones conectados entre si se convierten en un solo eslabón con un número específico de juntas. Por lo anterior, se pasa del el diagrama cinemático del mecanismo de la figura 2.14 al de la figura 2.23.

Las 3 articulaciones de perno en una ranura de la figura 2.14 que están en funcionamiento son B, H y N, (ahora llamadas articulación C, H y L en la figura 2.23 respectivamente). Por otra parte, los eslabones 15 y 16 de la figura 2.14 formarán el eslabón numero 8 (en verde) de la figura 2.23, y los eslabones 5, 6, 4, 8 y 12 de la figura 2.14 formarán parte del eslabón tierra o bancada número 1 (en líneas punteadas) de la figura 2.23 debido al bloqueo que se genera en estos eslabones al contactar con un objeto. En este nuevo diagrama cinemático, el numero de eslabones n es igual a 11, el número total de uniones principales j_p es 13 y número total de uniones de orden superior (unión de perno en una ranura específicamente) j_h es 3. Por lo tanto se tiene: $M = 3(n - 1) - 2j_p - j_h = 3(11 - 1) - 2(13) - 3 = 1$.

De forma similar, en el modelo de dedo A pero con actuación de un motor rotacional (Ver figura 2.15), para el segundo caso AA-FP -FM ocurren los mismos acontecimientos importantes nombrados con la actuación de un motor lineal y se difiere únicamente en que desaparece la articulación actuada de perno en una ranura B de la figura 2.14 y en su lugar es actuada la articulación de perno C en la figura 2.15. Tomando en cuenta todos estos cambios, se pasa del diagrama cinemático

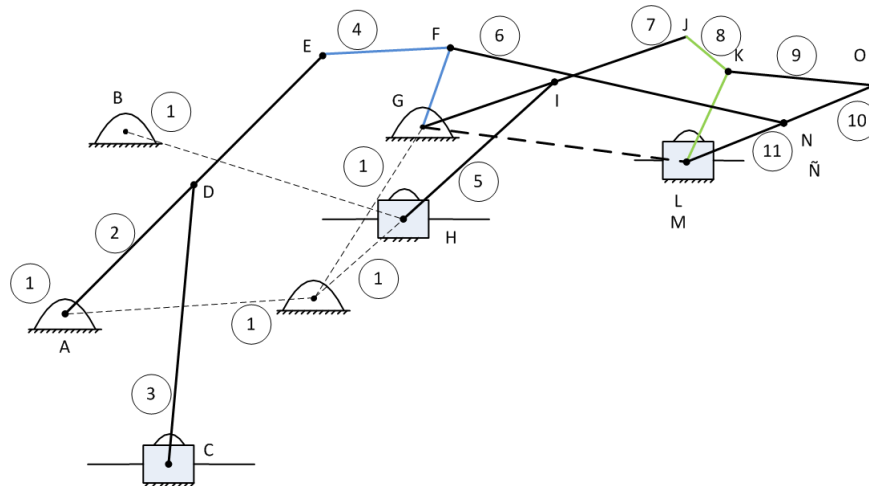


Figura 2.23: Diagrama cinemático modelo de dedo A para actuación con motor lineal, caso AA-FP-FM.

del mecanismo de la figura 2.15 al de la figura 2.24.

Las 2 articulaciones de perno en una ranura de la figura 2.15 que están en funcionamiento son H y N (ahora llamadas F y J en la figura 2.24 respectivamente). Por otra parte, los eslabones 15 y 16 de la figura 2.15 formarán el eslabón número 7 (en verde) de la figura 2.24, y los eslabones 5, 6, 4, 8 y 12 de la figura 2.15 formarán parte del eslabón tierra o bancada número 1 (en líneas punteadas) de la figura 2.24 debido al bloqueo que se genera en estos eslabones al contactar con un objeto. En este nuevo diagrama cinemático, el número de eslabones n es igual a 10, el número total de uniones principales j_p es 12 y número total de uniones de orden superior (unión de perno en una ranura específicamente) j_h es 2. Por lo tanto se tiene: $M = 3(n - 1) - 2j_p - j_h = 3(10 - 1) - 2(12) - 2 = 1$.

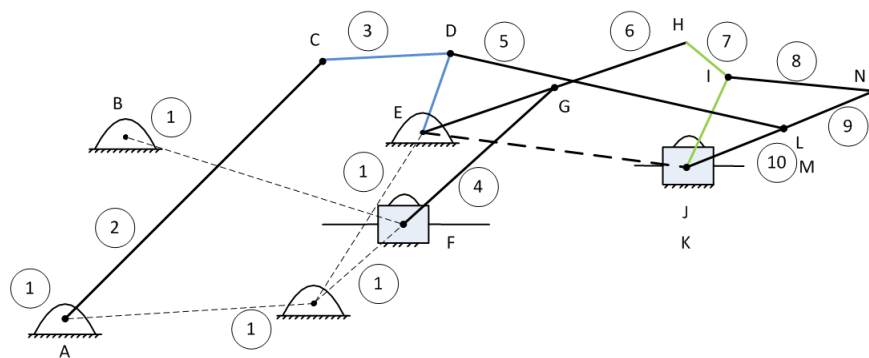


Figura 2.24: Diagrama cinemático modelo de dedo A para actuación con motor rotacional, caso AA-FP-FM.

2.4.1.2. Mecanismo de dedo B

El diseño del mecanismo del dedo pulgar se divide, como en la anatomía osteoarticular del dedo, en primer metacarpiano, falange proximal y falange distal. Cada una

de dichas partes cuenta con mecanismos de barras que permiten la transmisión de movimiento hasta la yema del dedo (el efector final).

De forma similar, tal como en el diseño de dedo A, el dedo B cuenta con dos mecanismos primordiales: mecanismo de sub-actuación (SA) y mecanismo de auto-adaptabilidad (AA). Los cuales no funcionan simultáneamente.

El mecanismo SA está activo mientras que el metacarpo y la falange proximal no se hayan bloqueado al hacer contacto con un objeto. En la figura 2.25 se observa en qué casos se activa alguno de los dos tipos de mecanismos que componen el modelo de dedo B.

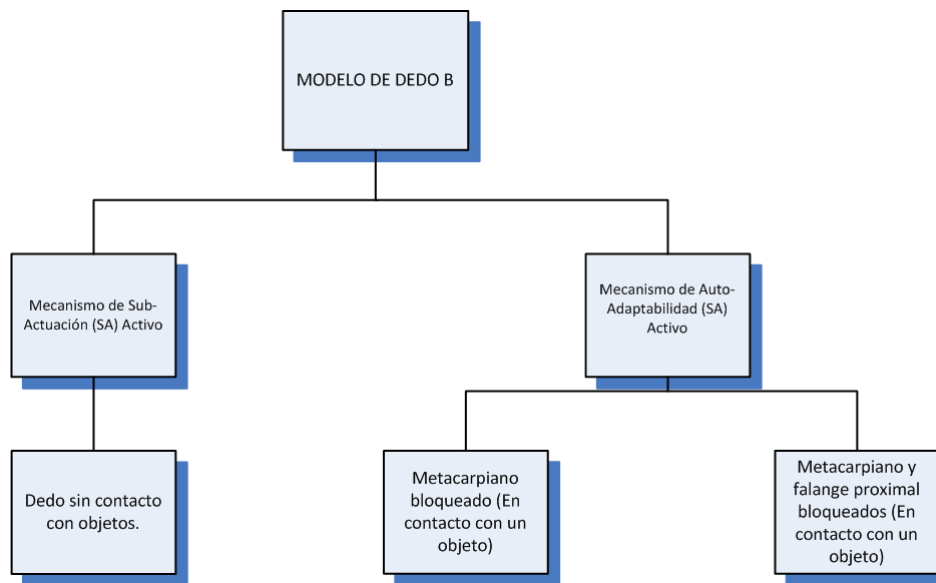


Figura 2.25: Casos en los que entra en funcionamiento alguno de los mecanismo que componen el modelo de dedo B.

A continuación se presenta el diagrama cinemático donde se visualizarán con claridad del mecanismo SA y AA presentes en el metacarpiano y las falanges proximal y distal.

2.4.1.2.1. Diagrama cinemático

En la figura 2.26 se presenta el diseño CAD del dedo pulgar en solidWorks y en la figura 2.27 el diagrama cinemático del mecanismo para actuación con motor lineal. En el diagrama cinemático de la figura 2.27 se visualiza la bancada del mecanismo notada con el número 1, los eslabones del 2 al 19, 2 uniones de perno en una ranura (A y O), y finalmente 19 uniones de revoluta (B, C, D, E, F, G, H, I, J, K, L, M, N, Ñ, P, Q, R, S, T). Adicionalmente, se observa que la articulación actuada para este mecanismo es la union A.

En este proyecto de tesis, por efectos de simulación en Adams, el motor lineal será omitido, lo que significa que las articulaciones A y D no existirán y el eslabón

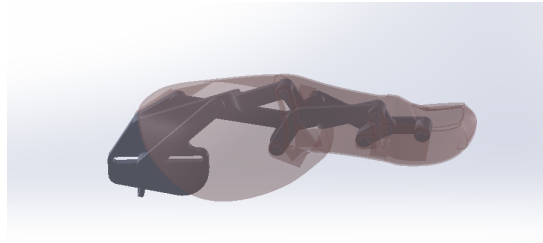


Figura 2.26: Casos en los que entra en funcionamiento alguno de los mecanismo que componen el modelo de dedo B.

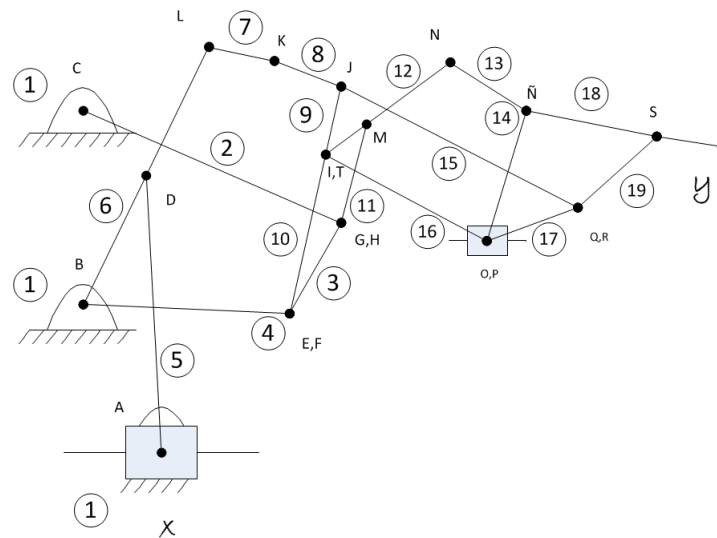


Figura 2.27: Diagrama cinemático modelo de dedo B para actuación con motor lineal.

5 será eliminado. Para garantizar el mismo comportamiento a pesar de los cambios, en el nuevo diagrama cinemático, será actuada la articulación de revoluta o de perno B. El nuevo diagrama cinemático se observa en la figura 2.28 y el diseño CAD se observa en la figura 2.29.

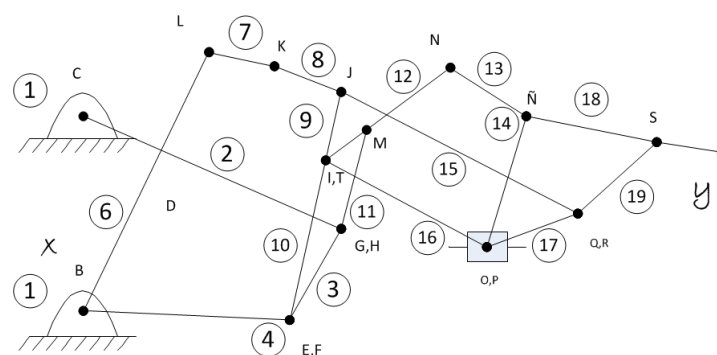


Figura 2.28: Diagrama cinemático modelo de dedo B para actuación con motor rotacional.

En la figura 2.30 y 2.31 se observa en color rojo el mecanismo SA y en azul el mecanismo AA, los cuales no se modifican al cambiar un motor lineal por uno rotacional.

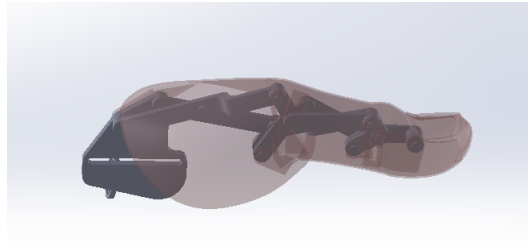


Figura 2.29: Diseño CAD del mecanismo de dedo B con motor rotacional.

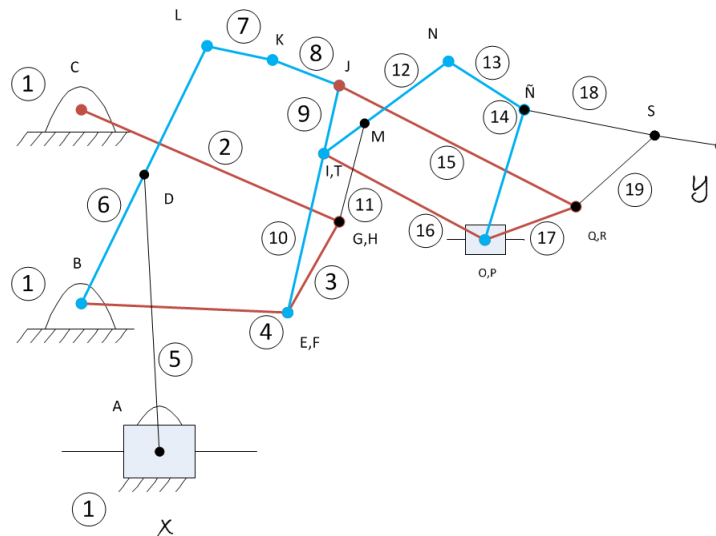


Figura 2.30: Mecanismos de SA y AA del modelo de dedo B para actuación con motor lineal.

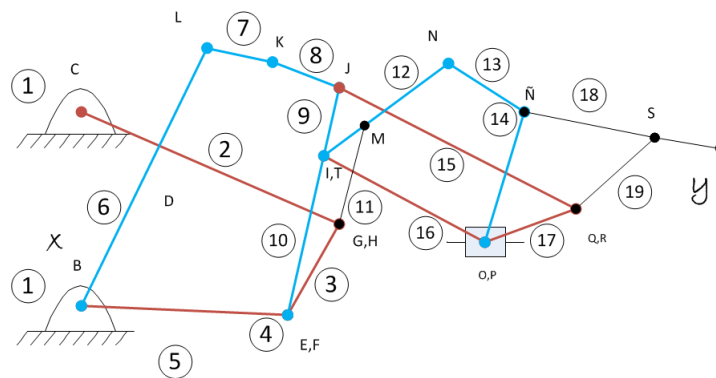


Figura 2.31: Mecanismos de SA y AA del modelo de dedo B para actuación con motor rotacional.

De forma similar al mecanismo de dedo A, este mecanismo de dedo B tiene dos comportamientos, el primero es cuando está activo el mecanismo SA (antes del contacto con un objeto) y el segundo cuando el mecanismo AA está activo (durante el contacto con un objeto). Dependiendo en cuál de los comportamientos se encuentre el mecanismo, algunas uniones que eran de perno, en el primer caso, llegarán a ser uniones de perno en una ranura durante el segundo caso. Se debe notar

nuevamente, que independientemente del comportamiento en el que se encuentre el mecanismo, los grados de libertad hallados mediante la ecuación de Grueber siempre debe ser el mismo, 1 para este caso.

A continuación se presenta el diagrama cinemático para los dos acontecimientos nombrados anteriormente, además del cálculo del número de grados de libertad.

Primer caso: Mecanismo SA activo

En el modelo de dedo B con actuación de un motor lineal, cuando el mecanismo SA está activo, dos acontecimientos importantes ocurren. Primero, solo existen articulaciones de orden inferior, específicamente articulación de perno o rotoide. Segundo, el comportamiento de algunos eslabones conectados entre sí, en la figura 2.27, se convierte en un solo eslabón (el eslabón número 4 en verde y el eslabón número 7 en naranja de la figura 2.32). En la figura 2.32 se observa el diagrama cinemático para el mecanismo de dedo B cuando es actuado por un motor lineal y cuando aun no se realiza contacto con objetos.

En este nuevo diagrama cinemático, el número de eslabones n es igual a 11, el número total de uniones principales j_p es 14 y número total de uniones de orden superior (unión de perno en una ranura específicamente) j_h es 1. Por lo tanto se tiene: $M = 3(n - 1) - 2j_p - j_h = 3(11 - 1) - 2(14) - 1 = 1$.

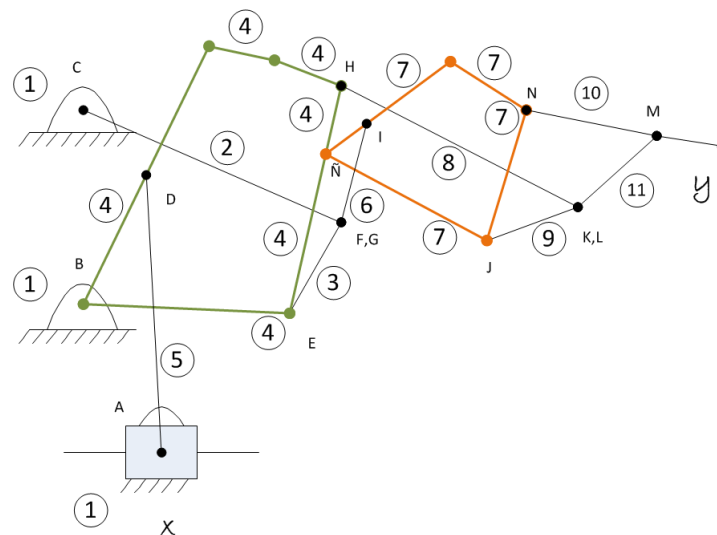


Figura 2.32: Diagrama cinemático modelo de dedo B para actuación con motor lineal antes del contacto con un objeto.

De forma similar, en el modelo de dedo B con actuación de un motor rotacional (Ver figura 2.28), cuando el mecanismo SA está activo, ocurren los mismos dos acontecimientos importantes nombrados arriba. Por lo tanto se pasa del diagrama cinemático del mecanismo de la figura 2.28 al de la figura 2.33.

En este nuevo diagrama cinemático, el número de eslabones n es igual a 10, el número total de uniones principales j_p es 13 y número total de uniones de orden superior j_h es 0. Por lo tanto se tiene: $M = 3(n-1) - 2j_p - j_h = 3(10-1) - 2(13) - 0 = 1$.

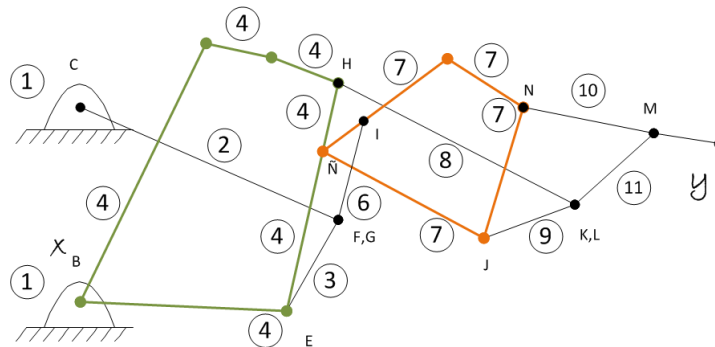


Figura 2.33: Diagrama cinemático modelo de dedo B para actuación con motor rotacional antes del contacto con un objeto.

Segundo caso: Mecanismo AA Activo

En el modelo de dedo B con actuación de un motor lineal, cuando el mecanismo AA está activo, se había dicho que dos comportamientos ocurren. El primero es cuando el metacarpiano hace contacto con un objeto y la falange proximal no (Caso AA-M). El segundo es cuando tanto el metacarpiano como la falange proximal contactan con un objeto (Caso AA-M-FP). Para cada uno de los dos casos mencionados existe un comportamiento diferente de los eslabones y las juntas, por consiguiente se tienen dos diagramas cinemáticos totalmente distintos.

Primero, para el caso AA-M solo una de las dos articulaciones de perno en una ranura (unión de orden superior), de la figura 2.27, están activas, la restante se comporta como articulación de perno (unión de orden inferior o unión principal). Además, el comportamiento de algunos eslabones conectados entre si se convierten en un solo eslabón. Por lo anterior, se pasa del diagrama cinemático del mecanismo de la figura 2.27 al de la figura 2.34.

Las 2 articulaciones de perno en una ranura de la figura 2.27 son A y O, de las cuáles solo estará en funcionamiento la articulación A (llamada también articulación A en la figura 2.34). Por otra parte, los eslabones 7, 8 y 9 de la figura 2.27 formarán el eslabón número 4 (en azul) de la figura 2.34, y los eslabones 12, 13, 14 y 16 de la figura 2.27 formarán el eslabón 7 (en rosado) de la figura 2.34. Finalmente los eslabones 2, 3, y 4, de la figura 2.27 formarán parte del eslabón tierra o bancada número 1 (en líneas punteadas) de la figura 2.34 debido al bloqueo que se genera en estos eslabones al contactar con un objeto. Finalmente, es relevante notar que la articulación C de la figura 2.27 no existe en la figura 2.34, ya que en este nuevo comportamiento, la unión llega a ser rígida y no existe rotación.

En este nuevo diagrama cinemático, el número de eslabones n es igual a 11, el número total de uniones principales j_p es 14 y número total de uniones de orden superior (unión de perno en una ranura específicamente) j_h es 1. Por lo tanto se tiene: $M = 3(n - 1) - 2j_p - j_h = 3(11 - 1) - 2(14) - 1 = 1$.

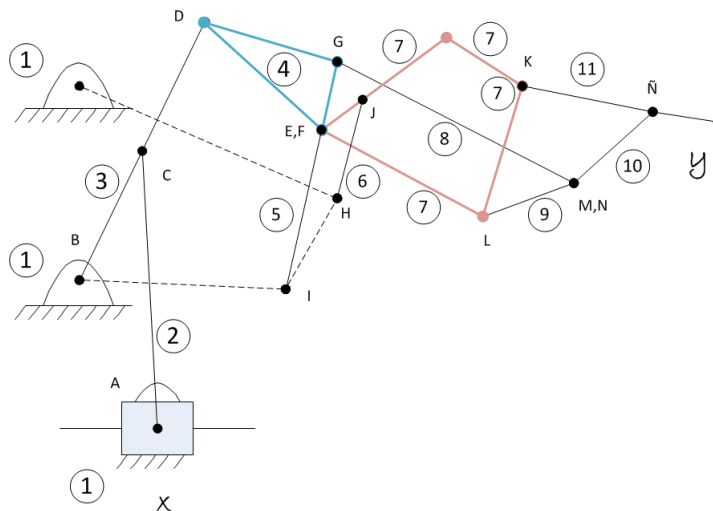


Figura 2.34: Diagrama cinemático modelo de dedo B para actuación con motor lineal, caso AA-M.

De forma similar, en el modelo de dedo A pero con actuación de un motor rotacional (Ver figura 2.28), para el primer caso AA-M ocurren los mismos acontecimientos importantes nombrados con la actuación de un motor lineal y se difiere únicamente en que desaparece la articulación actuada de perno en una ranura A de la figura 2.28 y en su lugar es actuada la articulación de perno B en la figura 2.35. Tomando en cuenta todos estos cambios, se pasa del diagrama cinemático del mecanismo de la figura 2.28 al de la figura 2.35.

La única articulación de perno en una ranura de la figura 2.28 es O, la cuál no estará activa (Ahora llamada L en la figura 2.35). Por otra parte, los eslabones 12, 13, 14 y 16 de la figura 2.28 formarán el eslabón número 7 (en rosado) de la figura 2.35. Además, los eslabones 7, 8 y 9 de la figura 2.28 formarán parte del eslabón 4 de la figura 2.35. Finalmente los eslabones 2, 3, y 4, de la figura 2.28 formarán parte del eslabón tierra o bancada número 1 (en líneas punteadas) de la figura 2.35 debido al bloqueo que se genera en estos eslabones al contactar con un objeto. En este nuevo diagrama cinemático, el número de eslabones n es igual a 10, el número total de uniones principales j_p es 13 y número total de uniones de orden superior (unión de perno en una ranura específicamente) j_h es 0. Por lo tanto se tiene: $M = 3(n - 1) - 2j_p - j_h = 3(10 - 1) - 2(13) - 0 = 1$.

Segundo, para el caso AA-FP-FM se pasa del diagrama cinemático del mecanismo de la figura 2.27 al de la figura 2.36.

Las 2 articulaciones de perno en una ranura de la figura 2.27 que están en funcionamiento son A y O (ahora llamadas articulación A y Ñ en la figura 2.36 respectivamente). Por otra parte, los eslabones 7, 8 y 9 de la figura 2.27 formarán el eslabón número 4 (en azul) de la figura 2.36, y los eslabones 2, 3 y 4 de la figura 2.27 formarán parte del eslabón tierra o bancada número 1 (en líneas punteadas)

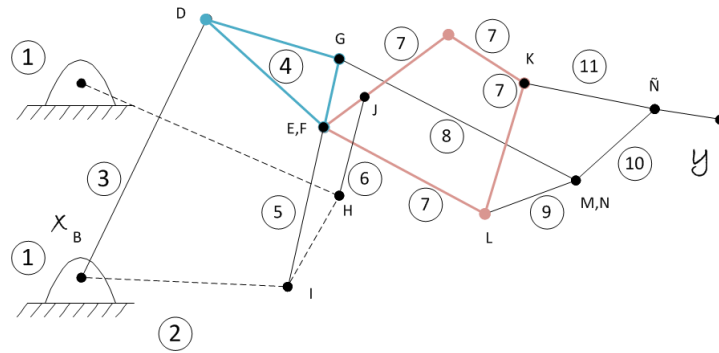


Figura 2.35: Diagrama cinemático modelo de dedo B para actuación con motor rotacional, caso AA-M.

de la figura 2.36 debido al bloqueo que se genera en estos eslabones al contactar con un objeto. En este nuevo diagrama cinemático, el número de eslabones n es igual a 14, el número total de uniones principales j_p es 18 y número total de uniones de orden superior (unión de perno en una ranura específicamente) j_h es 2. Por lo tanto se tiene: $M = 3(n - 1) - 2j_p - j_h = 3(14 - 1) - 2(18) - 2 = 1$.

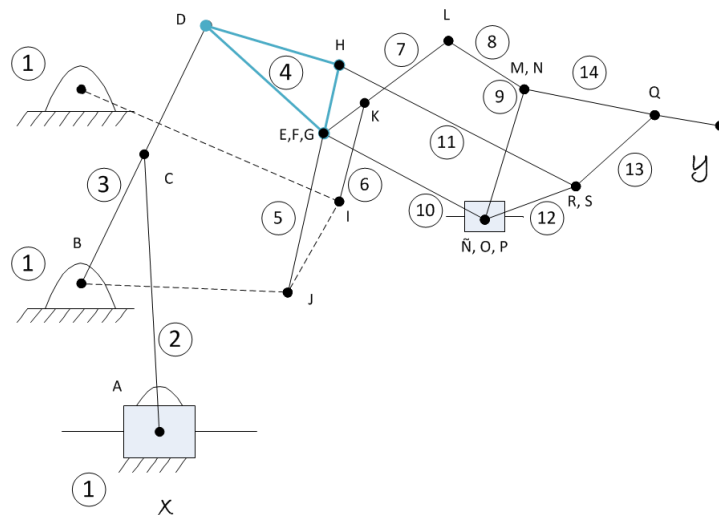


Figura 2.36: Diagrama cinemático modelo de dedo B para actuación con motor lineal, caso AA-FP-FM.

De forma similar, en el modelo de dedo B pero con actuación de un motor rotacional (Ver figura 2.28), para el segundo caso AA-FP -FM ocurren los mismos acontecimientos importantes nombrados con la actuación de un motor lineal y se difiere únicamente en que desaparece la articulación actuada de perno en una ranura A de la figura 2.27 y en su lugar es actuada la articulación de perno B en la figura 2.37. Tomando en cuenta todos estos cambios, se pasa del diagrama cinemático del mecanismo de la figura 2.28 al de la figura 2.37.

La única articulación de perno en una ranura de la figura 2.28 que está en funcionamiento es N (ahora llamada Ñ en la figura 2.37). Por otra parte, los eslabones 7, 8 y 9 de la figura 2.28 formarán el eslabón número 4 (en azul) de la figura 2.37, y los eslabones 2, 3 y 4 de la figura 2.28 formarán parte del eslabón

tierra o bancada número 1 (en líneas punteadas) de la figura 2.37 debido al bloqueo que se genera en estos eslabones al contactar con un objeto. En este nuevo diagrama cinemático, el número de eslabones n es igual a 13, el número total de uniones principales j_p es 17 y número total de uniones de orden superior (unión de perno en una ranura específicamente) j_h es 1. Por lo tanto se tiene: $M = 3(n - 1) - 2j_p - j_h = 3(13 - 1) - 2(17) - 1 = 1$.

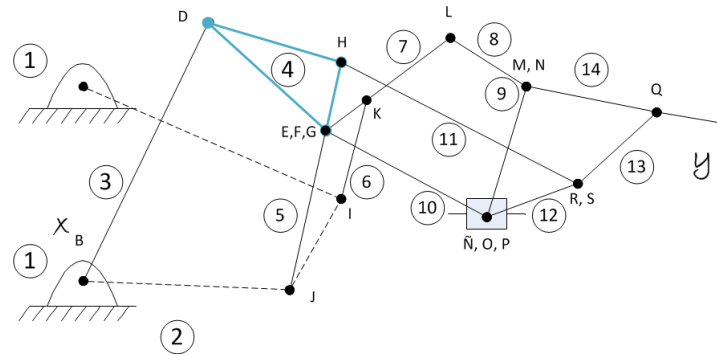


Figura 2.37: Diagrama cinemático modelo de dedo B para actuación con motor rotacional, caso AA-FP-FM.

Capítulo 3

Control de posición

El tema a desarrollar en este capítulo va dirigido a entender los esquemas de control implementados en la prótesis de mano virtual para realizar flexión y extensión de los dedos en el agarre de objetos mediante el seguimiento de una trayectoria de posición de la yema de los dedos deseada.

En la figura 3.1 se muestra el esquema utilizado para lograr el gobierno de la prótesis virtual de mano a partir de señales EMGs.

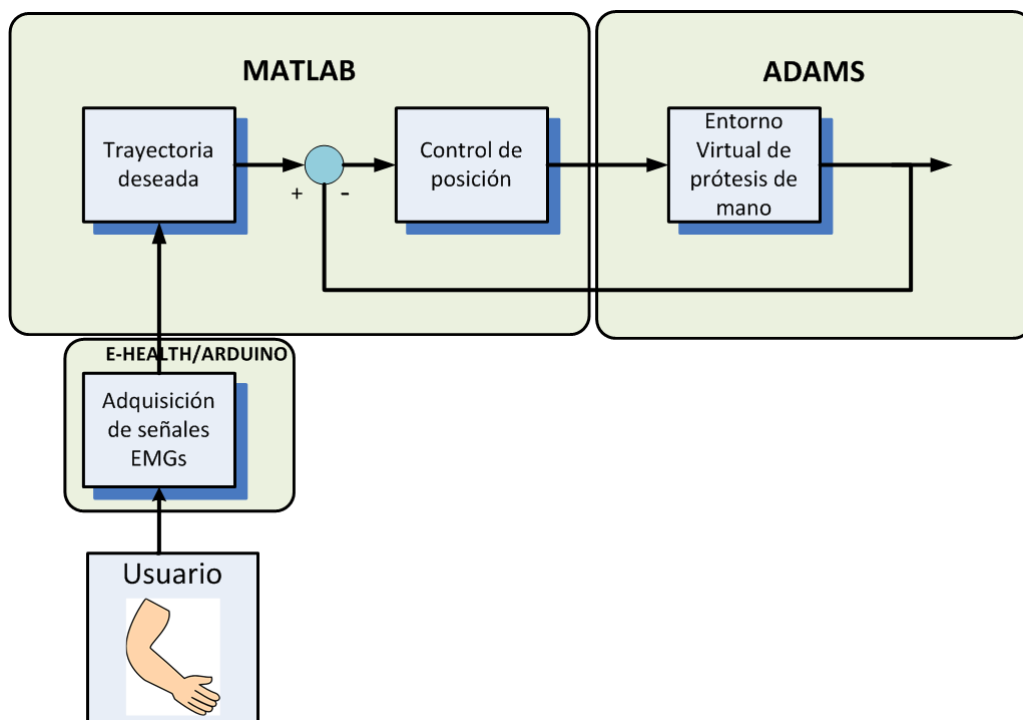


Figura 3.1: Estrategia de gobierno de prótesis de mano virtual.

3.1. Requerimientos de software

Por lo que respecta al software de programación e implementación de algoritmos de control, se requiere que se encargue de realizar el cálculo de la jacobiana, la sinto-

nización e implementación de las estrategias de control que permitan a la prótesis de mano realizar movimientos para lograr un agarre de objetos a través de señales EMGs, que finalmente se convertirán en un valor de par rotacional de las articulaciones activas de los dedos. En consecuencia se requiere un software matemático que permita la manipulación de matrices, funciones e implementación de algoritmos, la comunicación con otro softwares computacionales y dispositivos hardware.

Una vez claros los requerimientos, se procede a realizar una búsqueda de software. En la tabla 3.1 se observan algunas opciones para el software mencionado anteriormente.

SOFTWARE DE PROGRAMACIÓN E IMPLEMENTACIÓN DE ALGORITMOS DE CONTROL		
Matlab/ Simulink	GNU octave	Scilab/ Xcos
<p>MATLAB es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows, Mac OS X y GNU/Linux . Entre sus prestaciones básicas se hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware. El paquete MATLAB dispone de dos herramientas adicionales que expanden sus prestaciones, a saber, Simulink (plataforma de simulación multidominio) y GUIDE (editor de interfaces de usuario - GUI). Además, se pueden ampliar las capacidades de MATLAB con las cajas de herramientas (toolboxes); y las de Simulink con los paquetes de bloques (blocksets).</p> <p>Simulink es una herramienta de simulación de modelos o sistemas, es un entorno de programación visual y además permite implementar lazos de control sobre dichos sistemas.</p>	<p>Es un lenguaje de programación científica con una potente sintaxis orientada a las matemáticas con herramientas integradas de trazado y visualización. Es un software gratuito, se ejecuta en GNU / Linux, macOS, BSD y Windows. Drop-in compatible con muchos scripts de Matlab.</p> <p>La sintaxis de Octave es ampliamente compatible con Matlab. El intérprete Octave se puede ejecutar en modo GUI, como una consola, o invocado como parte de un script de shell.</p> <p>Resuelve sistemas de ecuaciones con operaciones de álgebra lineal en vectores y matrices.</p>	<p>Scilab es un software para análisis numérico, con un lenguaje de programación de alto nivel para cálculo científico. Es desarrollado por Scilab Enterprises, bajo la licencia CeCILL, compatible con la GNU General Public License. Las características de Scilab incluyen análisis numérico, visualización 2-D y 3-D, optimización, análisis estadístico, diseño y análisis de sistemas dinámicos, procesamiento de señales, e interfaces con Fortran, Java, C y C++. Mientras que la herramienta Xcos permite una interfaz gráfica para el diseño de modelos. El programa Scilab tiene un entorno llamado Xcos similar a Simulink de Matlab para simulación de sistemas dinámicos y resolución de sistemas de ecuaciones diferenciales. Este entorno posee varios paquetes que incluye algunas herramientas para simulación sencilla de circuitos eléctricos y termo hidráulica.</p>

Tabla 3.1: Software de programación e implementación de algoritmos de control disponible.

3.1.1. Elección de software

De la tabla 3.1 se ha elegido a Matlab/Simulink versión 2016a como el software de programación e implementación de algoritmos de control en este trabajo de grado. Una de las razones principales es que Matlab/Simulink acoge todas las funcionalidades tanto de GNU octave como Scilab/Xcos. Como dato curioso, Matlab/Simulink fue lanzado en 1984, GNU octave en 1988 y Scilab en 1994. Otra de las razones es el uso y manejo de Matlab durante el 70 % del pregrado.

3.2. Requerimientos para la estrategia de control

Para el gobierno de la prótesis de mano se requería de una estrategia de control sencilla, que no requiriera del modelo dinámico de la planta (MDI), de fácil implementación con una prótesis real. Lo anterior se debe a que uno de los objetivos de la tesis está dirigido a averiguar si las ventajas de una prótesis con mecanismos de auto adaptación podrían facilitar el control del agarre de objetos, utilizando una estrategia de control de poca complejidad.

Resulta muy complejo y poco práctico para una implementación real, la obtención del modelo dinámico inverso de un robot, ya que se requiere calcular para cada cuerpo que lo compone: el tensor de inercia, el primer momento de inercia, la masa, el centro de gravedad, las fuerzas de coriolis; además de la inercia de los actuadores, los frotamientos secos y los frotamientos viscosos de las articulaciones.

Para el caso particular de la mano bajo estudio, la obtención analítica del modelo dinámico es una tarea muy dispendiosa dada la existencia de varias cadenas cinemáticas cerradas que interactúan entre sí. Así que este enfoque se descartó desde el inicio del proyecto.

Entre las estrategias de control típicas que no requieren del modelo dinámico del robot (MDI), se encuentran aquellos que utilizan estructuras PID, dado que éstos pueden ser sintonizados mediante conocidas técnicas experimentales. Entre ellas se encuentran el PID articular, y el PID en espacio operacional. Por otro lado, una estrategia de control típica que sí requiere el modelo dinámico (MDI) de la planta es el control por par calculado (CTC) en cualquiera de sus opciones: CTC cartesiano, CTC operacional con corrección en el espacio articular, CTC operacional con corrección en el espacio operacional.

Se decide como procedimiento, implementar un controlador por cada dedo de forma independiente de los demás dedos. Para lograr la apertura y cierre de la mano de forma natural, solo se requiere cambiar la trayectoria de cada dedo de acuerdo a los rangos de variación de la articulación actuada. A continuación se describen las estrategias utilizadas en este trabajo.

3.3. Virtual Spring-Damper (VSD)

Es una estrategia utilizada en el control de brazos robóticos que tiene una inspiración biológica. No requiere del conocimiento del modelo, asume que existe una fuerza virtual que tira al efector final hacia la trayectoria deseada y en oposición se encuentra el robot. La fuerza de oposición se asimila a un resorte, mientras que la dinámica de los enlaces del robot se aproxima por modelos de amortiguadores. La ley de control consiste en calcular el par necesario en las articulaciones activas para vencer la oposición de la fuerza virtual y de los amortiguadores (Ver figura 3.2). Sus variantes incluyen términos para ajustar el efecto de la fuerza de gravedad u otras dinámicas no modeladas, mediante esquemas adaptativos con alguna acción integral del error. Se han presentado para el caso de robots de cadenas seriales, aunque el trabajo en [3] mostró que era aplicable al control de posición de un dedo actuado por mecanismo de barras simples.

Para su implementación se requiere el conocimiento de la matriz jacobiana, la medida del error de posición en el espacio cartesiano, y el ajuste de los parámetros de resorte y amortiguador.

La ley de control de esta estrategia (sin el término de compensación de la fuerza de gravedad) está definida por la ecuación 3.1

$$u = -C\dot{q} - J^T(q)k\Delta x \quad (3.1)$$

Donde k es el coeficiente de rigidez del resorte, C es el coeficiente de amortiguamiento de las articulaciones activas del robot, Δx es la elongación del resorte y J^T es la matriz jacobiana del robot. C denota una matriz definida positiva en la diagonal como sigue:

$$C = \text{diag}(c_1, \dots, c_n) \quad (3.2)$$

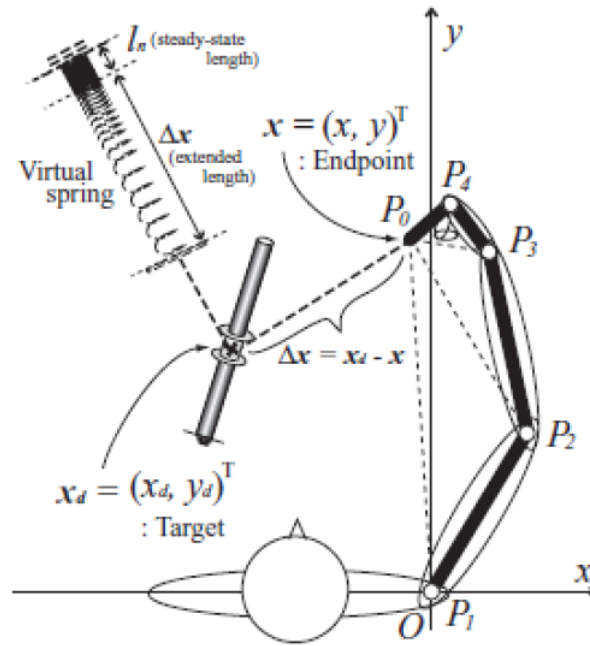


Figura 3.2: Esquema general del control VSD. Tomado de [3].

El enfoque de control de la hipótesis virtual spring damper puede extenderse a movimientos de alcance tridimensional bajo el efecto de la fuerza de gravedad. El término de gravedad que viene de la energía potencial $P(q)$ se representa en la ecuación 3.3.

$$g(q) = \left(\frac{dP}{dq} \right)^T \quad (3.3)$$

Donde $P(q)$ es la energía potencial del robot para una configuración q .

Entonces, se adiciona un estimador del término anterior al algoritmo de control de la ecuación 3.1, de manera que se obtiene la siguiente señal para controlar movimientos bajo el efecto la gravedad:

$$u = -C_0 \dot{q} - J^T(q)(\xi \sqrt{k} \dot{x} + k \Delta x) + Y \hat{\theta} \quad (3.4)$$

Donde Y representa un vector regresor función de la gravedad, y $\hat{\theta}$ una variable adaptativa para estimar el par que hace falta adicionar al par aplicado a la articulación activa, para llevar a un error de posición lo más próximo a cero posible.

3.3.1. Obtención de la jacobiana

3.3.1.1. Método analítico

Inicialmente el cálculo de la jacobiana se empezó a desarrollar mediante el método analítico expuesto en el marco teórico. Primero se procedió a hallar la tabla de parámetros geométricos para cada una de las articulaciones del modelo de dedo

A (cuando funciona el mecanismo SA), tomando el mecanismo del dedo como una cadena cinemática cerrada con restricciones, abriendo los lazos cerrados en una de las juntas pasivas y hallando los parámetros D-H para cada una de las cadenas cinemáticas abiertas o resultantes de abrir los lazos. Al respecto conviene decir que a cada cadena abierta se la considera como un robot, al cuál se le procede a hallar los parámetros DH.

Una vez hallados los parámetros D-H, se continuó con la construcción del modelo geométrico del modelo de dedo A, el cual relaciona los parámetros geométricos entre eslabones por medio de matrices de transformación. Para ello primero se procedió a hallar el modelo geométrico de cada robot (cadena serial) mediante el toolbox de matlab para robótica desarrollado por Peter Corke, el cual proporciona funciones básicas para el estudio y simulación de sistemas robóticos seriales.

Posteriormente se procedió a encontrar las ecuaciones de restricción a partir de los parámetros geométricos del dedo. Las ecuaciones de restricción son expresiones matemáticas que representan que los eslabones de una cadena cerrada no pueden abrirse cuando se mueven. Para el caso del modelo de dedo A de este proyecto, se abrió la cadena cinemática cerrada en un punto, pero solo para facilitar el cálculo de la Jacobina del modelo cinemático de un robot. Adviértase que aunque se analiza el dedo completo como dos, tres o cuatro cadenas abiertas separadas, el mecanismo físicamente sigue siendo uno solo y cerrado, y por tanto se debe garantizar que en el punto donde se abrió siempre permanezca cerrado, tanto en el eje y como en el x . Por consiguiente se tienen dos ecuaciones de restricción, una para el eje x y otra para el eje y .

Hasta este punto se tenían los modelos geométricos de cada una de las cadenas seriales del dedo en términos de incógnitas, ya que no se conocen los parámetros de los ángulos de las articulaciones. Con el fin de hallar estos valores se procede a utilizar la ecuación de freudenstein para la cual se debe tener un mecanismo de barras (la cadena serial resultado de abrir la cadena cinemática cerrada del dedo) con dos ángulo conocidos y dos desconocidos.

El ángulo de la única articulación actuada del dedo es conocido, el cual se va a estar midiendo todo el tiempo y hallando con este valor, mas algún otro fijo por la estructura; los otros dos ángulos desconocidos. Por lo tanto la primera cadena serial debía contener la articulación actuada, para luego hallar el valor de los dos ángulos desconocidos. A su vez, a partir de alguno de estos dos ángulos conocidos que se acaban de hallar, se obtiene uno de los ángulos de la segunda cadena serial, y con uno de los ángulos hallados para la segunda cadena se pueden hallar los otros dos ángulos de una tercera cadena serial y así sucesivamente.

La estrategia de este método es dividir el dedo en tantas cadenas seriales como sea posible y que tengan como conocido un solo ángulo y desconocidos solamente dos. Al ser el mecanismo del dedo tan complejo, no es fácil elegir cuales deben ser las cadenas seriales de tal forma que se puedan hallar todos los ángulos. Sin embargo, se encontró una única forma para lo anterior, que cumpliera con todos los

requerimientos para poder hallar la jacobiana con el método analítico. Para más detalle de este método se puede consultar la tesis de pregrado titulada: Aplicación de la hipótesis virtual spring damper en el control de posición de un dedo de prótesis de mano robótica sub-actuada: estudio de simulación [3].

Algunos de los requerimientos más difíciles de cumplir para poder usar este método fue el de tener tantas articulaciones pasivas como restricciones, por lo que se debía ser muy hábil a la hora de dividir los mecanismos del dedo para tener igual número de restricciones que articulaciones pasivas. Por su parte el número de articulaciones pasivas nunca iba a cambiar, pero si se podía obligar, según en cuantas cadenas se abriera el mecanismo, a que el número de restricciones fueran iguales a las juntas pasivas.

Se encontró una sola forma que pudo satisfacer el requerimiento mencionado anteriormente, pero desafortunadamente al momento de hallar la jacobiana del modelo cinemático directo (compuesta por la jacobiana de las juntas pasivas (J_{cp}) y la jacobiana de las juntas activas (J_{ca})), la cual requiere la inversa de una matriz jacobiana que representa las juntas pasivas (J_{cp}), se observó que la jacobiana J_{cp} no tenía inversa. En consecuencia no fue posible hallar, por este método, la jacobiana que permite describir las velocidades del efector final como una función de las velocidades articulares de la articulación activa.

La existencia de una singularidad en la matriz Jacobiana tendría como interpretación física la existencia de singularidades (pérdida de rango de la jacobiana para valores aislados de q) en el mecanismo. Esto es posible, dado que el diseño mecánico de los dedos fue realizado para que el movimiento quede bloqueado mecánicamente al superarse ciertos ángulos de flexión y de extensión del dedo, de modo que no sea posible extender el dedo más allá de lo que extiende un dedo humano, y también para que no pueda ser flexionado más allá de lo que flexiona un dedo humano. Pero para el caso, esta singularidad de la matriz impide la obtención analítica de la jacobiana.

3.3.1.2. Método experimental

Dado que, sin embargo, en el rango de valores de ángulos de la articulación actuada correspondientes a movimientos de los dedos en flexión y extensión similares a los de los dedos humanos, el software de simulación Adams muestra que el movimiento es continuo y que no existen singularidades en el movimiento (Adams hace el cálculo numérico de las singularidades y en el caso de existir alguna, la simulación es abortada). Este hecho llevó a la posibilidad de buscar un método experimental para obtener una expresión que fuera función del ángulo de la articulación actuada.

El procedimiento se describió en el capítulo del marco teórico. Aquí se describen aspectos de implementación y resultados. Primero, se debe tener el bloque de la planta en adams exportado a **Simulink** (bloque de Adams_sub1), el cual tendrá inmersa la dinámica del dedo de la prótesis de mano exceptuando el comportamiento generado por la fuerza de gravedad. A dicho bloque le entrará un valor de un par

rotacional constante, correspondiente al valor del par que se le aplicará a la articulación actuada y que permitirá la flexión y extensión del dedo como en el proceso de agarre de un objeto. El valor del par es encontrado manualmente de tal forma que permita el movimiento de cierre del dedo de una forma suave. Por otro lado, como salida de este bloque se tiene la posición cartesiana x y y de la yema del dedo en cada instante de tiempo, la cual es obtenida gracias a un sensor de posición proporcionado por **Adams**. Además de tener como salidas las posiciones cartesianas, también se tiene la posición angular de la articulación actuada (ángulo en grados de la articulación actuada respecto a la tierra o bancada del mecanismo del dedo) que cambia gracias al par aplicado, y es obtenida por un sensor de posición angular ubicado en esta articulación. Lo anterior se puede observar en la figura 3.3. El hecho clave de este procedimiento es que, como se demostró en el estudio cinemático de los dedos, la estructura de dedo bajo estudio tiene grado de libertad efectivo igual a 1, lo que implica que para cada ángulo de la única articulación activa del dedo, existe una y solo una posición cartesiana de la yema posible.

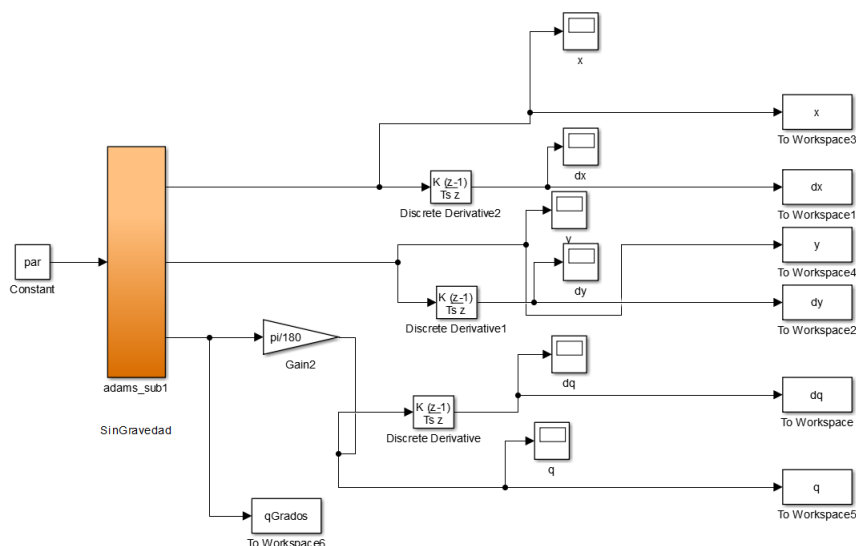


Figura 3.3: Entradas y salidas del bloque de la planta en simulink para hallar la jacobiana.

En la figura 3.3 también se observa que se derivan las posiciones cartesianas x y y y la posición angular de la articulación actuada, obteniendo como resultado las velocidades cartesianas \dot{x} y \dot{y} notadas en el diagrama de bloques por dx y dy , además de la velocidad angular $\dot{\theta}_a$ de la articulación actuada notada como dq . Una vez se simula el archivo en **Simulink** que contiene lo mostrado por la figura 3.3, se obtienen los valores de las velocidades articulares (dx y dy) y la velocidad angular (dq) en un bloque de **To Workspace** que permita realizar el cálculo de la jacobiana mediante código en **Matlab** aplicando la ecuación 1.14. Esta simulación se realiza para un valor de par de $-0,003 \text{ dyn.m}$ (con un valor de par positivo gira hacia otro sentido, el cual no es el sentido deseado) aplicado durante 1.4 segundos, tiempo suficiente para el cierre del dedo en **Adams**. Los valores de la posición x y y cartesianas también son obtenidos mediante un bloque **To Workspace** con el objetivo de poder manipularlas en el entorno de **matlab** y posteriormente guardarlas junto con la información del tiempo de simulación en un archivo **.mat** como una matriz

de 3 filas (tiempo, posición en X y posición en Y). De este archivo **.mat** mencionado se extraerá alguna posición **XY** para ser la posición deseada en algunas de las pruebas de control, así como también este archivo podrá ser tomado como la trayectoria deseada para algunas otras pruebas de control.

En la imagen 3.4 se aprecia la trayectoria cartesiana obtenida por la yema del dedo al aplicar en la articulación actuada un valor de par de $-0,003 \text{ dyn.m}$.

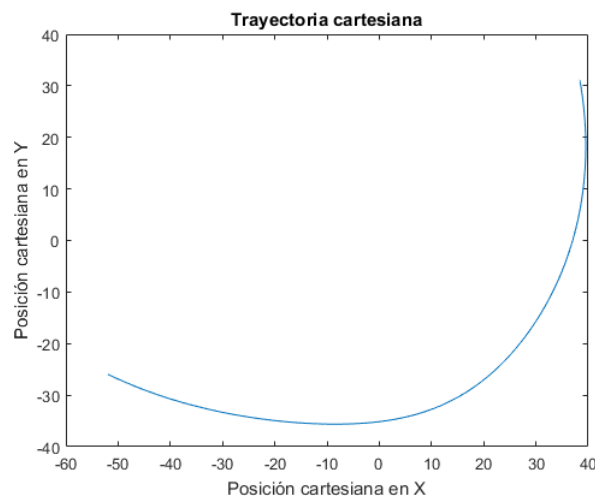


Figura 3.4: Trayectoria cartesiana obtenida para el modelo de dedo A.

A continuación, en la figura 3.5, se presenta la posición en radianes obtenida de la articulación actuada durante 1.4 segundos con un valor de par constante de $-0,003 \text{ dyn.m}$.

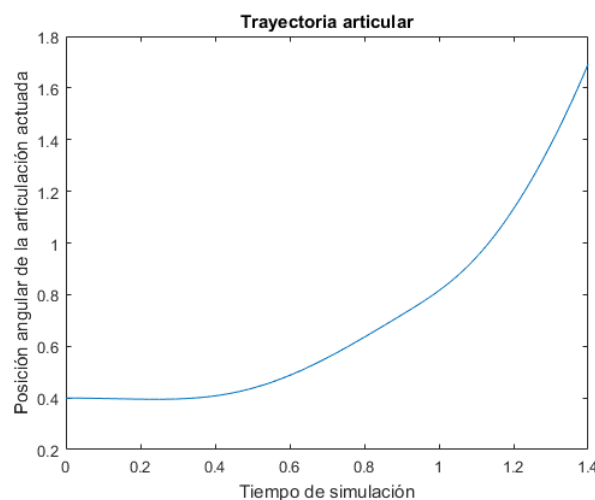


Figura 3.5: Trayectoria de la articulación actuada obtenida para el modelo de dedo A.

Al aplicar la ecuación 1.14 en **Matlab** se obtiene una jacobiana compuesta de dos filas por una columna (2×1), donde la dimensión 2 corresponde a la velocidad lineal en x y y de la yema del dedo, y la dimensión de una columna corresponde al único ángulo de la articulación activa.

Como se mencionó en el marco teórico, la jacobiana está en función del ángulo actuado o lo que es lo mismo la posición angular sensada de la articulación actuada, notada como th en la figura 3.3 del diagrama de bloques en **simulink**.

Ahora bien, hasta ahora solo se ha encontrado el valor de jacobiana como función del ángulo actuado, pero el objetivo es encontrar una función explícita que represente a la jacobiana, por lo tanto se procede a usar la herramienta de **curve fitting** llamada **cftool** en **Matlab**. Esta herramienta permite ingresar un vector para la variable independiente x y otro vector para los datos de la variable independiente y generando una función $f(x)$ que representa el comportamiento de los datos. Para este caso en específico se ingresan como parámetros th y $j1$: $cftool(th, j1)$, y posteriormente se ingresan los parámetros th y $j2$: $cftool(th, j2)$. Una vez ejecutado este comando para cada una de las filas $j1$ y $j2$ de la jacobiana, se obtiene la interfaz mostrada en la figura 3.6.

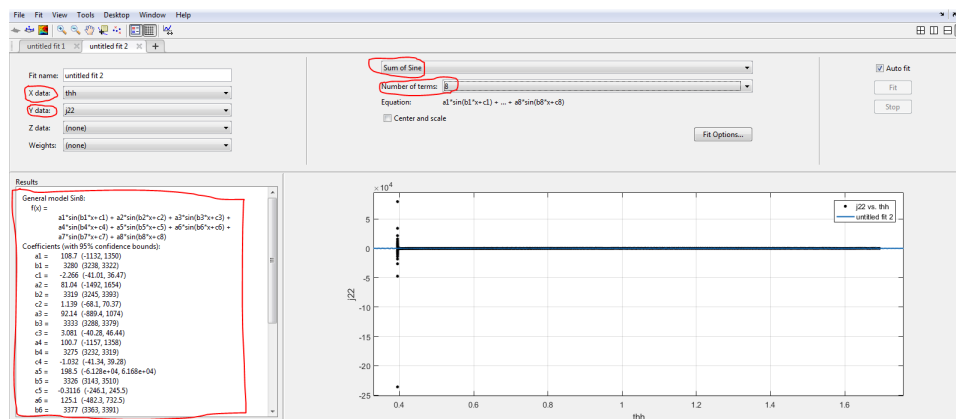


Figura 3.6: Entorno de la herramienta *cftool* para estimación de la función que representa la fila 2 de la jacobiana.

En la parte superior izquierda de la figura 3.6 se observa la variable que representa X data o la variable independiente, en este caso thh . Lo mismo sucede para la variable Y data. Además en la parte central de estas mismas imágenes se observa una casilla desplegable *Sum of Sine* la cual ofrece las siguientes opciones: *Interpolant*, *Linear Fitting*, *Polynomial*, *Power*, *Rational*, *Smoothing Spline*, *Sum of Sine* y *weibull*. Estas opciones representan el tipo de función $F(X)$ a la que se desea ajustar los datos. En este proyecto se decidió aproximar la función a una de tipo *Sum of Sine*, a la cual se le puede elegir un número máximo de términos de 8 (subrayado en color rojo en la imagen 3.6). En la parte inferior izquierda se observa la función generada $f(x)$ mediante una suma de senos de 8 términos con los valores de sus coeficientes de a_1 hasta a_8 , b_1 hasta b_8 y c_1 hasta c_8 . De esa forma se encuentra la jacobiana en función del ángulo th , es decir $J(\theta_a)$.

Para comprobar si la función estimada de la jacobiana es correcta, se crean dos archivos de **function** en **Matlab** con la función y los correspondientes valores de los coeficientes generados por **cftool**. Cada función tendrá como parámetro de entrada el ángulo sensado de la articulación actuada durante la simulación. Una vez

se le proporcione este parámetro a cada una de las funciones se debería reproducir la misma jacobiana de la que se obtuvo la función.

La comparación de la fila 1 de la jacobiana se observa en la figura 3.7 y la comparación de la fila 2 se observa en la figura 3.8.

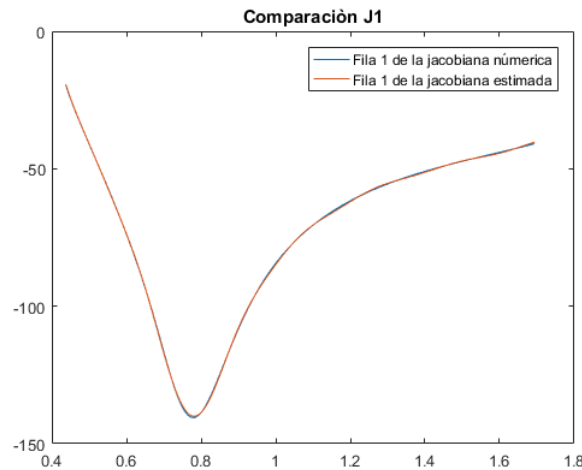


Figura 3.7: Comparación de la fila 1 de la jacobiana numérica y la función de la jacobiana estimada con entrada del parámetro th .

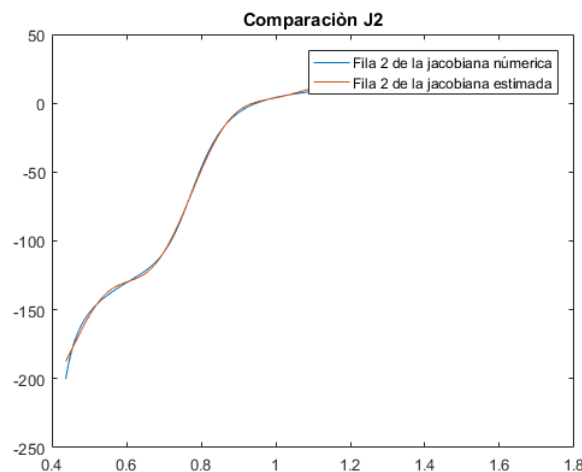


Figura 3.8: Comparación de la fila 2 de la jacobiana numérica y la función de la jacobiana estimada con entrada del parámetro th (sin información de los 500 primeros datos).

Para efectos de visualización se presentan las gráficas 3.9 y 3.10 de comparación entre las velocidades $X_{simulacion}$ y $Y_{simulacion}$ medidas por los sensores en **Adams** y las velocidades $X_{estimada}$ y $Y_{estimada}$ halladas mediante el modelo cinemático directo (ecuación 1.3) gracias a la jacobiana estimada. Cabe resaltar que las velocidades $X_{simulacion}$ y $Y_{simulacion}$ deberían ser iguales a $X_{estimada}$ y $Y_{estimada}$ respectivamente,

para comprobar que la jacobiana hallada es capaz de reproducir las velocidades cartesianas medidas por los sensores en Adams.

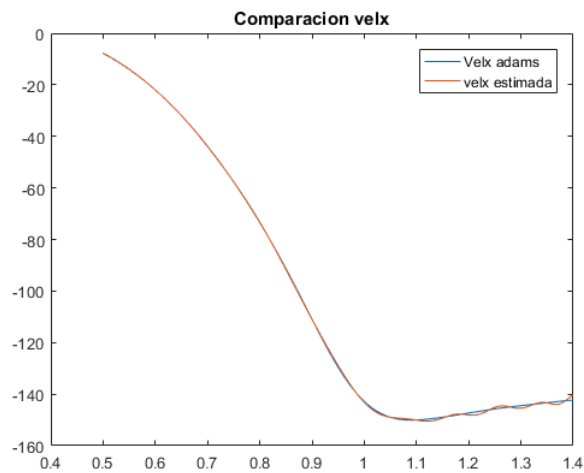


Figura 3.9: Comparación entre la velocidad cartésiana $\dot{X}_{simulacion}$ y $\dot{Y}_{estimada}$.

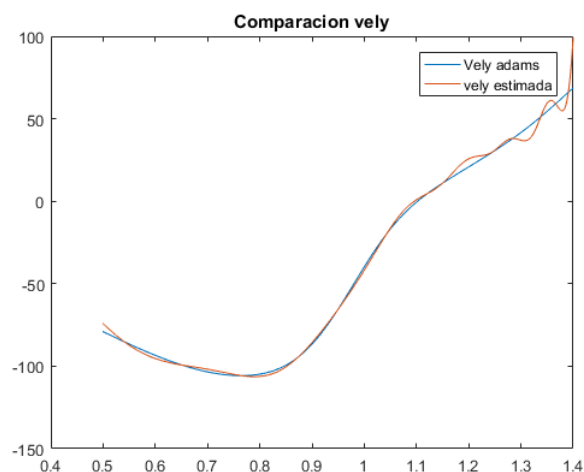


Figura 3.10: Comparación entre la velocidad cartésiana $\dot{Y}_{simulacion}$ y $\dot{Y}_{estimada}$.

Finalmente, se pueden utilizar las funciones de las jacobianas en un esquema de control.

3.3.2. Sintonía de la ley de control

El procedimiento de sintonía consiste en encontrar valores de las constantes de amortiguación y de elasticidad del amortiguador y del resorte respectivamente, así como de la ganancia de la ley adaptativa utilizada para obtener una fuerza que compense los efectos de la gravedad y dinámicas no modeladas. Este procedimiento se intentó por ensayo y error y también mediante un esquema de mínimos cuadrados tratando de encontrar la combinación de parámetros que minimizara el

error de seguimiento.

Los resultados de este intento de sintonía no se presentan en esta monografía por razones de espacio, pero la sintonía desde este enfoque no pudo realizarse. La hipótesis para este resultado fallido es la dificultad de compensar el efecto de la gravedad en el sistema. Esto por cuanto existe una gran cantidad de eslabones en cadenas cerradas, donde el efecto de la gravedad varía de forma importante en cada posición de la articulación activa, así que esta línea de trabajo debe buscar alternativas para una estimación del vector de fuerzas gravitacionales sin el conocimiento del modelo dinámico en mecanismos mas complejos que los enlaces seriales que se han contemplado en la literatura. También puede ocurrir que la aproximación de la dinámica de los eslabones en cadena cerrada por un solo amortiguador, sea insuficiente en este caso.

3.4. PID en el espacio operacional

Dados los resultados fallidos de sintonía en el caso previo, se consideró conservar el error en el espacio operacional pero en el esquema de un PID, dado que la acción integral en esta estrategia tiene una interpretación clara. El esquema se muestra en la figura 3.11 y se representa mediante la ecuación (3.5). En cierta forma, este esquema debería ser similar a lo planteado en la ley VSD, ya que el error cartesiano se transforma en un término equivalente a una fuerza a aplicar en el efector final (yema) por la ley PID. Esta fuerza se traslada a la articulación actuada mediante la matriz jacobiana, como se ve en la ecuación (3.5).

$$\gamma = J^T [K_p(X^d - X) + K_d(\dot{X}^d - \dot{X}) + K_I \int_{t_0}^t (X^d - X) d\tau] \quad (3.5)$$

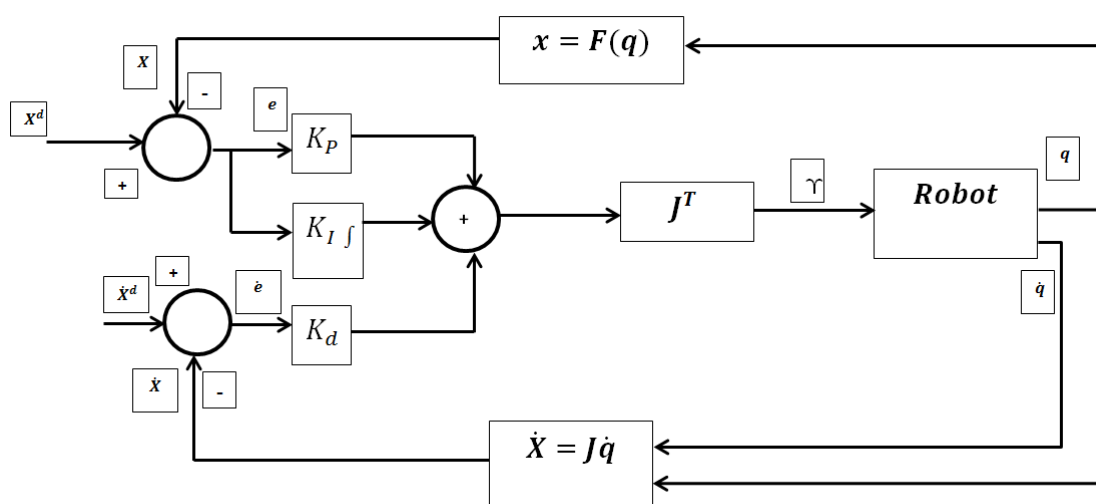


Figura 3.11: Diagrama de bloques de un esquema de control PID en el espacio de tarea. Tomado de [4].

3.4.1. Esquema de control

La idea principal del control para la prótesis de mano es poder controlar cada dedo independientemente. Para ello se obtendrán en total 5 jacobianas halladas sin el efecto de la fuerza de la gravedad desde el archivo de adams en el cuál únicamente estarán activos los mecanismos de Sub Actuación (tomando el diseño de los dedos únicamente con articulaciones de perno, sin tener en cuenta las articulaciones de perno en una ranura que son activadas en la fase autoadaptable de los dedos). Posteriormente, para la sintonización del controlador, se realizará bajo el funcionamiento únicamente de los mecanismo Sub Actuados y con el efecto de la gravedad. Finalmente para probar el controlador se hará con el archivo de Adams que contiene la prótesis de mano bajo los efectos de la gravedad y con la funcionalidad autoadaptable (articulaciones de perno en una ranura ahora estarán activadas). Lo anterior se puede observar más claramente en la figura 3.12

Por otro lado, las señales EMGs captadas por el usuario serán claves para defi-



Figura 3.12: Fases y condiciones para realizar el control PID en el espacio de tarea para la prótesis de mano.

nir la posición deseada de la mano en el proceso del agarre de objetos, y se espera que el carácter de autoadaptabilidad del mecanismo facilite la implementación de control, y que este sepa responder a la dinámica que no se tuvo en cuenta tanto en la jacobiana como en su sintonización. Se espera que el controlador sea tan robusto, que a pesar de no haber contemplado el comportamiento autoadaptable

cuando funcionan articulaciones de perno en una ranura, se pueda lograr un agarre de objetos autoadaptable (con el funcionamiento de articulaciones de perno en una ranura) en un ambiente que simula la física del mundo real.

Respecto al esquema de control, se eligió desarrollar este proyecto con un PID en el espacio de tarea (Ver figura 3.11). En este esquema de control se requiere hallar la jacobina para cada uno de los dedos de la prótesis de mano virtual, la cuál debe ser obtenida del archivo de adams sin los efectos de la gravedad y cuando los dedos de la prótesis tienen el mecanismo SA Activo (Es decir cuando el mecanismo solo tiene articulaciones de perno). Por lo anterior, en **SIMULINK** se usa el bloque en Adams que representa el comportamiento de la planta sin los efectos de la gravedad.

En cuanto al procedimiento para la implementación de control, primero se debe realizar el esquema de control en tiempo continuo y proceder con el método de Ziegler y Nichols para sintonizar controladores basado en la ganancia crítica K_{cr} y periodo critico P_{cr} . Una vez encontrados los valores de K_p , τ_i y τ_d para un PID, se procede a discretizar el controlador, que se obtuvo previamente en tiempo continuo, mediante la aproximación bilineal (trapezoidal o Tustin).

Ahora bien, en el método de sintonización de Ziegler y Nichols basado K_{cr} y P_{cr} sugiere establecer inicialmente la ganancia proporcional en un valor $K_p = 1$, $K_D = 0$ y $K_I = 0$, posteriormente empezar a multiplicar por dos o dividir entre dos el valor de K_p , hasta que el sistema oscile. En la figura 3.13 se observa la implementación de los bloques en simulink para el control de posición XY de la yema del modelo de dedo A con un PID en el espacio de tarea en tiempo continuo. Finalmente, con un valor de $KP = -2 \times 10^{-4}$, es posible que la variable controlada oscile, es decir, la posición X y Y de la yema del dedo.

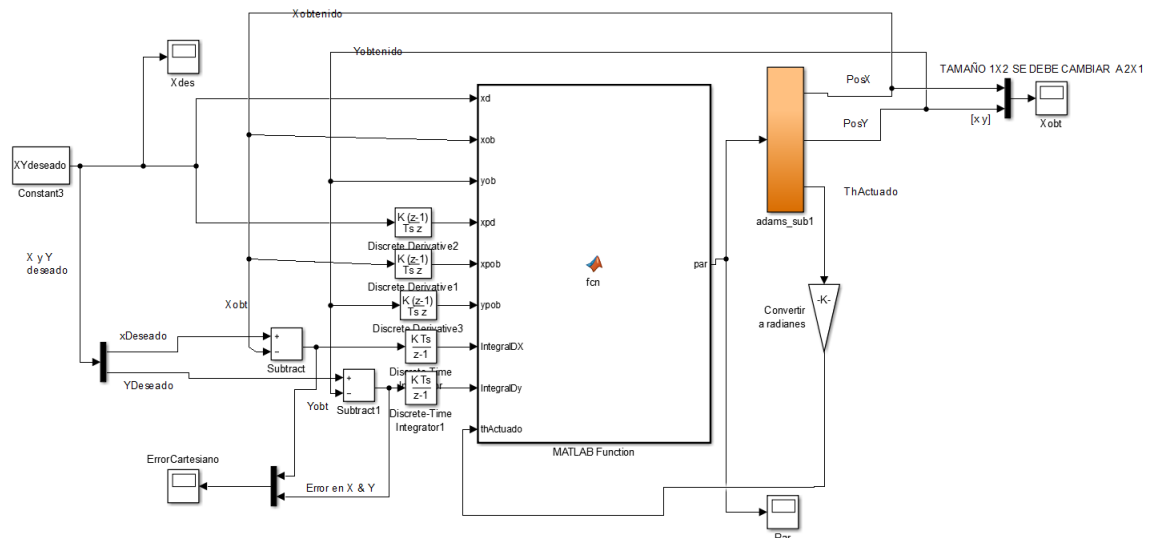


Figura 3.13: Implementación de un controlador en el espacio de tarea para el control de posición de la yema del modelo de dedo A.

Una vez la variable controlada oscila, se guarda el valor de K_p (K_{cr}) y se mide el

periodo de las oscilaciones, el cual corresponderá al periodo crítico P_{cr} . Una vez obtenidos K_{cr} y P_{cr} se proceden a usar las formulas de sintonización de la tabla 1.1 para un PID, que permiten hallar las constantes K_p , τ_d y τ_i , además de discretizar el controlador y obtener la función de transferencia $C(z)$ de la ecuación 1.23 $q_0, q_1 y q_2$.

A continuación se presenta el esquema de control en tiempo discreto en simulink (ver figura 3.14), el cual debe tener previamente definidas en un script de Matlab las siguientes variables de discretización: $q_0, q_1 y q_2$ y un tiempo de muestreo de 0,001 segundos.

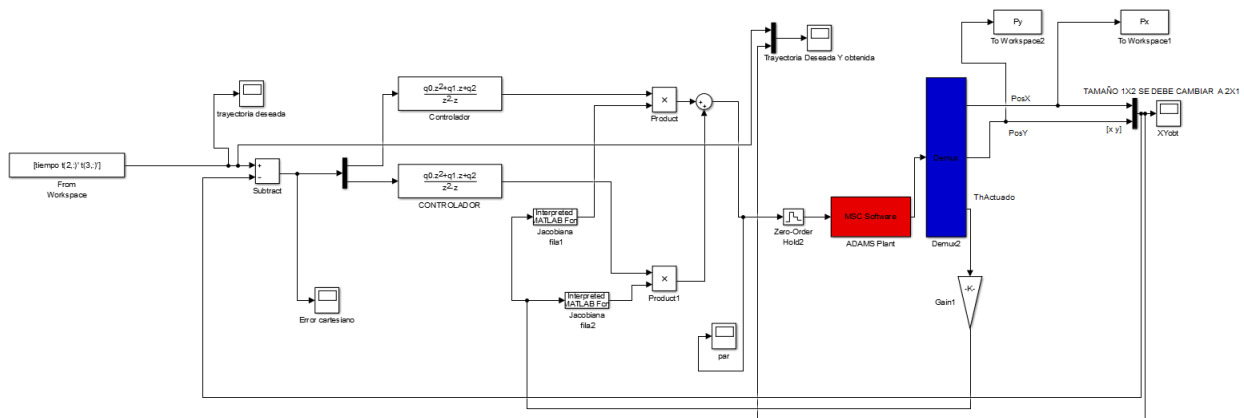


Figura 3.14: Implementación de un controlador en el espacio de tarea discretizado para el control de posición de la yema del modelo de dedo A.

Finalmente al simular el controlador implementado para una posición específica del dedo, se observa que al realizar la trayectoria de grado cinco hasta la posición de referencia se presenta un seguimiento de trayectoria bastante bueno, sin oscilaciones, pero solo en el caso en el que el dedo se encuentra sin el efecto de la gravedad. Al activar el efecto de la gravedad el seguimiento de la trayectoria no es posible, presentando fuertes oscilaciones del dedo, lo cuál no le permitiría agarrar objetos.

3.4.1.1. Conclusión

Aunque en este escenario fue posible sintonizar un controlador que logra el seguimiento de la trayectoria en ausencia de la fuerza de gravedad, la sintonía no fue posible cuando se considera el efecto de la gravedad. Este resultado es concordante con el obtenido en el enfoque de virtual spring damper, lo que refuerza la hipótesis de que en este sistema de cadenas cinemáticas cerradas que dependen unas de otras, el efecto dinámico de la fuerza de gravedad en el mecanismo posee una dinámica compleja que no se resuelve satisfactoriamente con una acción integral. Para comprobar esto, se propone identificar un modelo dinámico lineal que describa los cambios del ángulo de la articulación actuada ante cambios del par aplicado a la articulación actuada. Si al implementar un controlador para seguimiento de trayectorias a partir de este modelo identificado el desempeño del controlador mejora, se

podría concluir que los esquemas anteriormente probados fallan debido a la simplicidad de los esquemas para tener en cuenta la verdadera dinámica del sistema en presencia de la fuerza de gravedad.

3.5. Esquema de control cascada sintonizado por IMC

3.5.1. Teoría

Este esquema de control, tomado de [5], es un control simple para robots manipuladores. El esquema cascada está compuesto por un controlador de estabilización en el lazo interno y un controlador de seguimiento en el lazo externo. El lazo interno utiliza un controlador PD para estabilizar el robot, consiguiendo la ubicación de polos deseada para el modelo linealizado de la dinámica del robot. Este lazo permite hacer una corrección rápida (pero imprecisa) evitando que el dedo se aleje demasiado de la trayectoria. El lazo externo utiliza una ley PID obtenida mediante una sintonización por modelo interno que garantice el desacoplamiento entrada-salida y el seguimiento de trayectoria. Las ganancias de los controladores PD y PID se relacionan directamente con el modelo de robot linealizado mediante expresiones simples de forma cerrada. Para el dedo en estudio, el sistema resultante es SISO, con lo que el desacoplamiento no es necesario, pero sí se debe garantizar que en lazo cerrado (en ausencia de incertidumbre del modelo), se logre el seguimiento de la consigna.

Considere el esquema de la Figura 3.15.

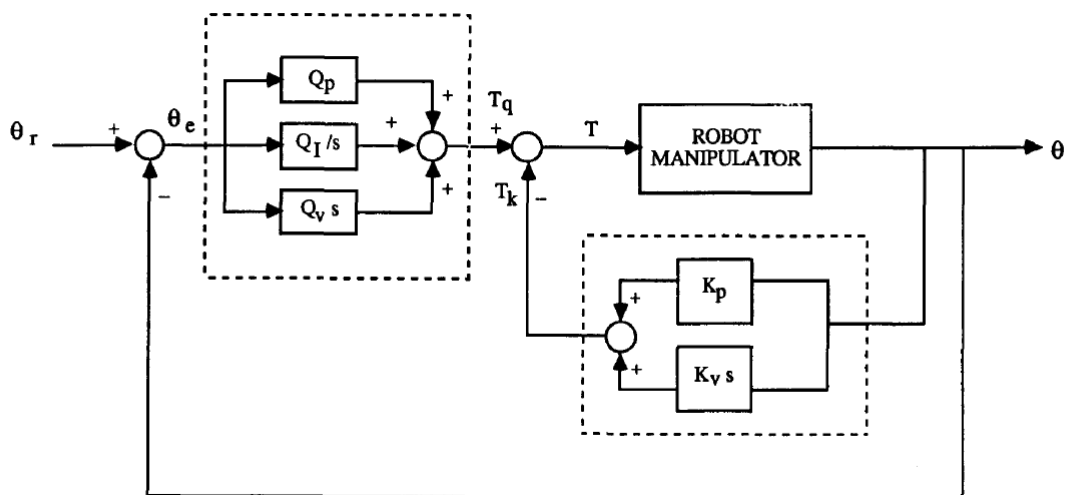


Figura 3.15: Estructura de control cascada. Tomado de [5].

La ley de control total esta dada por

$$\begin{aligned}
T(s) &= -T_k(s) + T_q(s) = -K(s)\theta(s) + Q(s)\theta_e(s) \\
&= -[K_p + K_v s]\theta(s) + \left[Q_p + \frac{Q_I}{s} + Q_v s \right] \theta_e(s) \\
T(t) &= -K_p\theta(t) - K_v\dot{\theta}(t) + Q_p\theta_e(t) + Q_I \int_0^t \theta_e(t)dt + Q_v\dot{\theta}_e(t) \quad (3.6)
\end{aligned}$$

El sistema linealizado del robot en lazo abierto corresponde a un sistema en espacio de estados de orden dos, tal que una ganancia de realimentación $T_k = -\mathbf{K}\mathbf{x}$, siendo \mathbf{x} las variables de estado del sistema identificado, ubique los autovalores de lazo cerrado en cualquier posición deseada tal que la dinámica del lazo interno sea rápida y estable. La función de transferencia de lazo cerrado del lazo interno sería de la forma de la ecuación (3.7).

$$W_1(s) = \frac{\Theta(s)}{T(s)} = \mathbf{C} (s\mathbf{I} - (\mathbf{A} - \mathbf{BK})^{-1}) \mathbf{B} \quad (3.7)$$

Donde las matrices \mathbf{A} , \mathbf{B} , \mathbf{C} y \mathbf{D} son las matrices de estado del modelo estándar lineal del robot, y \mathbf{K} la matriz de realimentación de estado. La matriz de realimentación puede calcularse a partir de los autovalores deseados, por un método de asignación de polos convencional.

Para el diseño del control PID del lazo externo, cuya función de transferencia es $Q(s) = \left[Q_p + \frac{Q_I}{s} + Q_v s \right]$, se realiza una sintonía basada en modelo interno tal que la función de transferencia de los dos sistemas en cascada se lleve al modelo de la ecuación (3.8).

$$W_1(s)Q(s) = \frac{K_t}{s} \quad (3.8)$$

Donde para este caso SISO, K_t es una constante que como veremos, define la velocidad de respuesta del controlador, y es un parámetro de sintonía del control por modelo interno.

En el caso SISO, en ausencia de ceros en $W_1(s)$, la ecuación 3.7 toma la forma de la ecuación (3.9).

$$W_1(s) = \frac{1}{\alpha_1 s^2 + \alpha_2 s + \alpha_3} \quad (3.9)$$

Con lo que, usando (3.8), se obtiene:

$$Q(s) = W_1^{-1}(s) \cdot \frac{K_t}{s} = [\alpha_1 s^2 + \alpha_2 s + \alpha_3] \cdot \frac{K_t}{s}$$

Llevando a:

$$Q(s) = \alpha_2 K_t + \frac{\alpha_3 K_t}{s} + \alpha_1 K_t s \quad (3.10)$$

$$Q(s) = Q_p + \frac{Q_I}{s} + Q_v s \quad (3.11)$$

Con esta ley bajo efecto, la función de transferencia del sistema total realimentado será:

$$\Theta(s) = \frac{1}{\frac{1}{K_t}s + 1} \Theta_r(s)$$

Con lo que este modelo interno impuesto para el controlador determina una respuesta de grado uno estable, que lleva a la salida al seguimiento de referencias constantes, con una constante de tiempo $\frac{1}{K_t}$. Si la trayectoria a seguir es lenta, el desempeño puede ser razonablemente aceptable.

El diseño se logra definiendo una constante de tiempo adecuada para el movimiento del dedo, y garantizando que el lazo interno tenga una dinámica más rápida que ésta, lo cual fácilmente se puede lograr, traduciendo esta especificación temporal a los polos deseados del algoritmo de ubicación de polos.

Ya que en el procedimiento de identificación las variables de estado obtenidas no se corresponden al ángulo y su derivada como en la Figura 3.15, en la implementación debe diseñarse un observador de estado para obtener las variables de estado del modelo a partir de la medida del ángulo de la articulación activa disponible. El diseño del observador de estado también se puede realizar mediante técnicas convencionales.

El diseño en el espacio articular ofrece posibilidades interesantes para una implementación en un prototipo real de la mano, ya que todo el diseño se puede realizar a partir de la medida del ángulo de la articulación activa, lo cual es fácil de medir en la práctica.

3.6. Implementación

La idea principal del control para la prótesis de mano es poder controlar cada dedo independientemente. Para ello se identificará el modelo en espacio de estados para los dos tipos de planta que se trabaja en la prótesis. El primer modelo corresponde a la dinámica del dedo pulgar, y el segundo correspondiente a la dinámica del dedo índice. Cabe aclarar que el modelo dinámico del dedo índice es el mismo para los dedos medio, anillo y meñique. Lo anterior debido a que en el diseño mecánico, el mecanismo de dedo A es el mismo para el dedo índice, medio, anillo y meñique.

La identificación del modelo en espacio de estados se realiza desde el archivo de adams en el cuál únicamente estarán activos los mecanismos de Sub Actuación (tomando el diseño de los dedos únicamente con articulaciones de perno, sin tener en cuenta las articulaciones de perno en una ranura que son activadas en la fase autoadaptable de los dedos). Finalmente el controlador se implementa tanto para cuando los mecanismos sub actuados están activos, como para cuando los mecanismos sub actuados y auto-adaptables están activos simultáneamente.

Por otro lado, las señales EMGs captadas por el usuario serán claves para definir la posición deseada de la mano en el proceso del agarre de objetos. Se espera que el carácter de autoadaptabilidad del mecanismo facilite la implementación de control, y que el controlador sea robusto y pueda responder a la dinámica de la auto-adaptación que no se tuvo en cuenta en la identificación. De esa manera se lograría un agarre de objetos autoadaptable (con el funcionamiento de articulaciones de perno en una ranura).

3.6.1. Identificación de las plantas

Se eligió trabajar con el esquema de control para el seguimiento de la trayectoria de los manipuladores robóticos (Ver figura 3.15). En este esquema de control, la variable controlada será la posición angular de la articulación actuada y la variable manipulada será el valor de par de la articulación actuada.

Para la implementación del esquema de control se procede a identificar la planta del dedo pulgar y posteriormente la planta del dedo indice utilizando el Toolbox **ident** de identificación de Matlab (Ver figura 3.16).

El toolbox de identificación **ident**, solicita los valores de entrada (valor de par de la articulación actuada) y salida de la planta (Angulo de la articulación actuada), así como el tiempo de muestreo. Luego de incluir esta información, se procede a definir el tipo de modelo a obtener, ya sea en función de transferencia, espacio de estados u otros. Posteriormente, se procede a seleccionar aquel estimado que muestra una mayor aproximación con el modelo real, como se muestra en la figura 3.17.

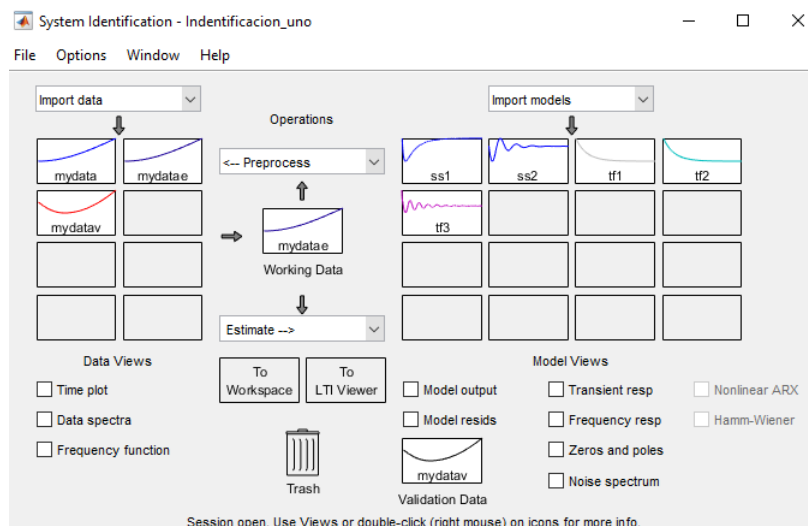


Figura 3.16: Interfaz de usuario de toolbox de identificación Ident

El diagrama de bloques en simulink para la identificación de la planta del dedo indice y pulgar se observa en la figura 3.18.

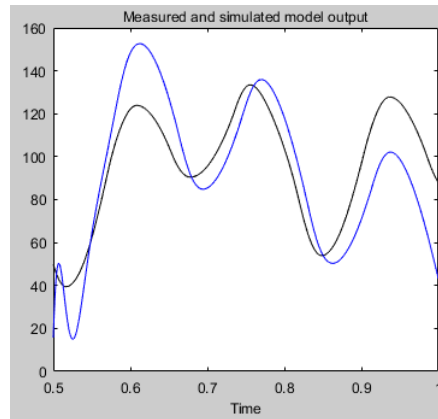


Figura 3.17: Comparación entre modelo real y estimado por ident.

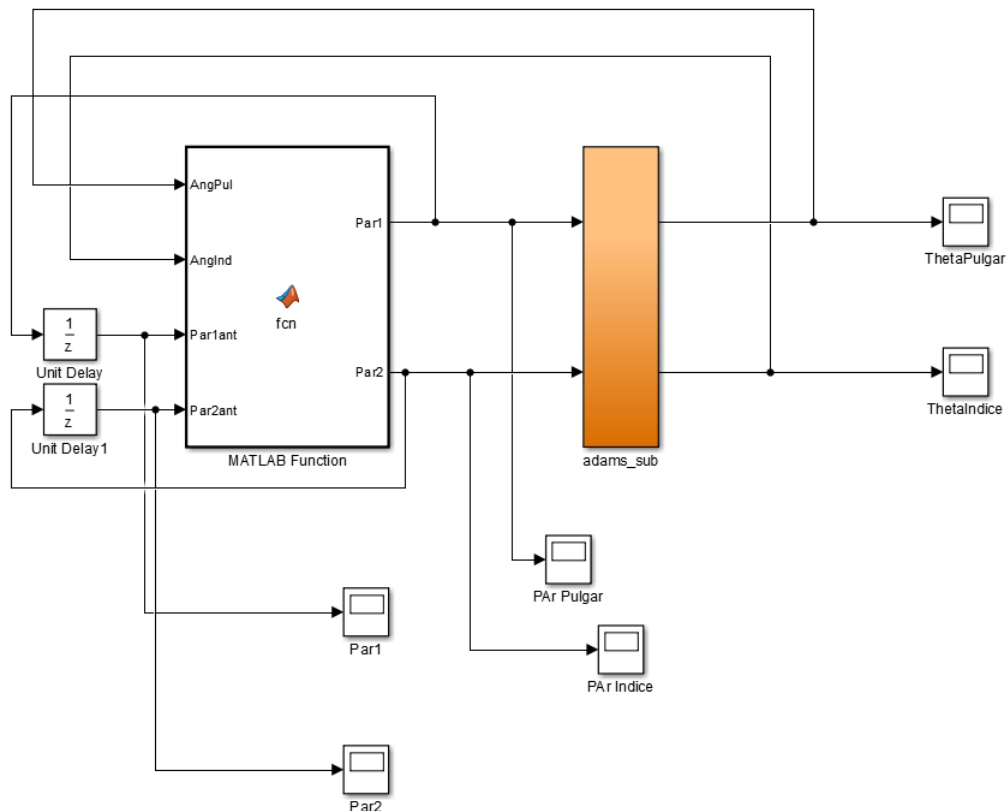


Figura 3.18: Diagrama de bloques para realizar la identificación del modelo de estados para el dedo pulgar e índice.

Para generar el bloque llamado *adams_sub*, en naranja de la figura 3.18, se debe generar un archivo *Controls_plant* desde Adams (Ver figura 3.19), donde se establezcan dos entradas y dos salidas (Ver procedimiento en el apéndice A). Las dos entradas corresponden al valor de par de la articulación actuada para el dedo pulgar e Índice, las dos salidas corresponden al valor de ángulo de la articulación actuada

del dedo pulgar e índice. Dichos archivos *Controls_plant* deben estar ubicados en la carpeta de los archivos de identificación en Matlab.

Para el proceso de identificación es importante encontrar previamente valores iniciales y adecuados del par tal que ante un cambio en la entrada (valores de par) haya un cambio en la salida (ángulo actuado) de la planta. Para ello se debe encontrar un rango de movimiento en grados en el cual los dedos realicen el movimiento de apertura y cierre con el fin de agarrar objetos. Posteriormente se buscan valores iniciales de par que permitan el movimiento de los dedos en el rango de grados deseado. Para el caso del dedo pulgar el movimiento requerido para cierre y apertura está entre 10° a -30° (ángulo de medida obtenida para la articulación actuada) respectivamente. Para el caso del dedo índice, medio, anillo y meñique, el movimiento requerido para apertura y cierre está entre 4° a -30° (ángulo de medida obtenida para la articulación actuada) respectivamente.

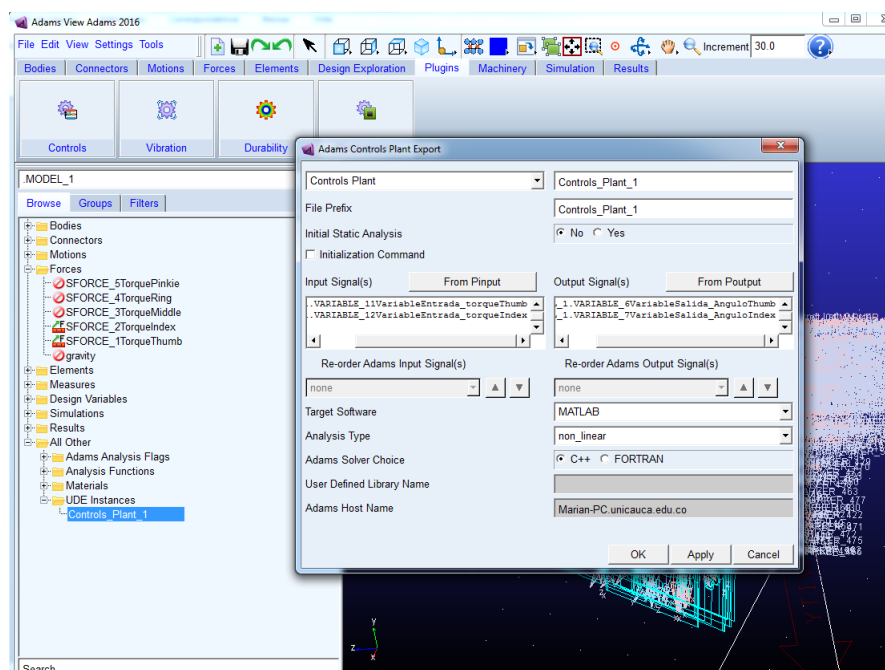


Figura 3.19: Generación del archivo *Controls_plant* desde Adams con las entradas y salidas de la planta.

Las figuras 3.20 y 3.21 corresponden al cambio de ángulo en grados de la articulación actuada (salida de la planta) causada por la excitación de par en *dyn.m* a la entrada de la planta para el dedo pulgar e índice respectivamente.

Para la excitación del par, se busca un valor inicial que permita el movimiento de cada uno de los dedos, el cual se debe establecer en los bloques de *UnitDelay* de la figura 3.18. Dentro del bloque *MATLAB Function* se establecen condiciones para que el sentido del par cambie en una magnitud mayor de acuerdo al valor superior e inferior del rango de movimiento en ángulos específico de cada dedo (rango de apertura y cierre de los dedos). Lo anterior con el fin de que los dedos suban y bajen dentro de un rango en grados de la articulación actuada.

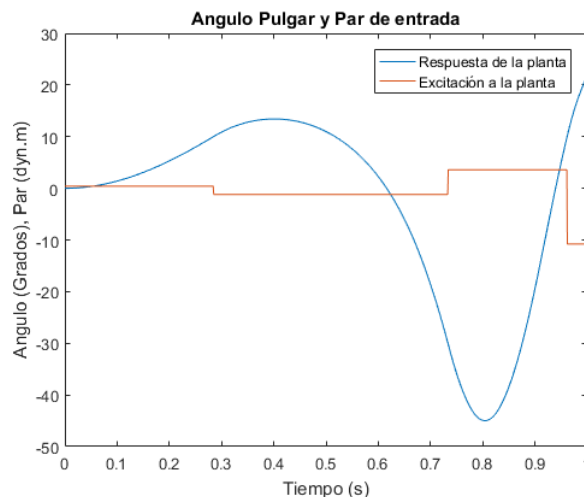


Figura 3.20: Cambio de ángulo generado por una excitación de par a la entrada de la planta del dedo Pulgar.

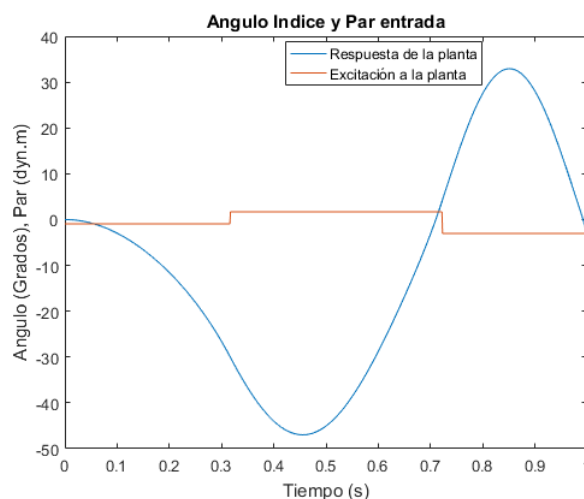


Figura 3.21: Cambio de ángulo generado por una excitación de par a la entrada de la planta del dedo Índice.

Posteriormente, utilizando la herramienta *ident* en dos ocasiones independientemente, se ingresa la información de simulación del par de entrada y el ángulo de salida de cada dedo, y se estima un modelo de la planta en espacio de estados en tiempo continuo. En 3.12 y 3.13 se observan las matrices A , B , C y D del modelo de espacio de estados para el dedo Pulgar e Índice respectivamente.

$$\begin{aligned}
 A &= \begin{bmatrix} 3,151 & -13,42 \\ 4,935 & -3,54 \end{bmatrix}; & B &= \begin{bmatrix} -0,0003329 \\ -0,09352 \end{bmatrix}; \\
 C &= \begin{bmatrix} 440,6 & -2,911 \end{bmatrix}; & D &= \begin{bmatrix} 0 \end{bmatrix};
 \end{aligned}
 \tag{3.12}$$

$$\begin{aligned} A &= \begin{bmatrix} 3,209 & -9,152 \\ 2,664 & -3,656 \end{bmatrix}; & B &= \begin{bmatrix} -0,002255 \\ -0,1281 \end{bmatrix}; \\ C &= \begin{bmatrix} 630,4 & -2,767 \end{bmatrix}; & D &= \begin{bmatrix} 0 \end{bmatrix}; \end{aligned} \quad (3.13)$$

Finalmente, como nota importante es preciso aclarar que en el archivo de *Adams* donde se encuentra la simulación de la prótesis de mano, el dedo Pulgar realiza el movimiento del cierre del mismo hacia la palma de la mano con un par positivo, el dedo Índice y Medio realizan el cierre hacia la palma con un par negativo, y finalmente los dedos Anillo y Meñique realizan el cierre hacia la palma de la mano con un par positivo.

Por lo anterior, cuando se pase de espacio de estados a función de transferencia del modelo de la dinámica de los dedos, se debe incluir un signo menos en la función de transferencia únicamente para el dedo Anillo y Meñique, ya que el cierre de estos dedos se ejecuta con un par positivo, y no con uno negativo como lo hace la planta del dedo índice. De esa manera, la planta de los dedos anillo y meñique serán iguales a la del dedo Índice.

3.6.2. Esquema de control

Al tener el modelo de los dos tipos de dedo previamente identificados, se procede a implementar el siguiente esquema de control para cada dedo (ver figura 3.15), con el fin de realizar el seguimiento del ángulo deseado de las articulaciones actuadas. Se recuerda que el lazo interno es para la estabilización (Controlador PD) y el lazo externo para el seguimiento de consigna (Controlador PID).

En el lazo interno para la estabilización del sistema, se halla el valor de las constantes K_p y K_v , pertenecientes a un controlador LQR, mostrado en la figura 3.22 en color azul. Cabe aclarar que en la teoría del marco teórico correspondiente a este esquema de control, el controlador utilizado para la estabilización del sistema era un *PD*, pero en esta implementación se utilizará un controlador *LQR*.

Primero se debe hallar las constantes del controlador, utilizando el comando $K_{ap} = lqr(A, B, Q, R)$. Donde A, B, C y D son las matrices del modelo de espacio de estados, y Q y R están definidas como:

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad R = \begin{bmatrix} 0,01 \end{bmatrix};$$

Como solo se conoce una variable de estado, siendo esta el ángulo de la articulación actuada (θ), y el modelo tiene dos variables estado, que son ángulo de la articulación actuada (θ) y velocidad de la articulación actuada ($\dot{\theta}$), entonces se implementa un observador de estados para obtener la velocidad, cuyo esquema se muestra en la figura 3.22 en color amarillo.

Posteriormente se procede a hallar la ganancia que permite ubicar los polos de lazo cerrado en una ubicación deseada *po*. Para hacer esto, se usa el comando

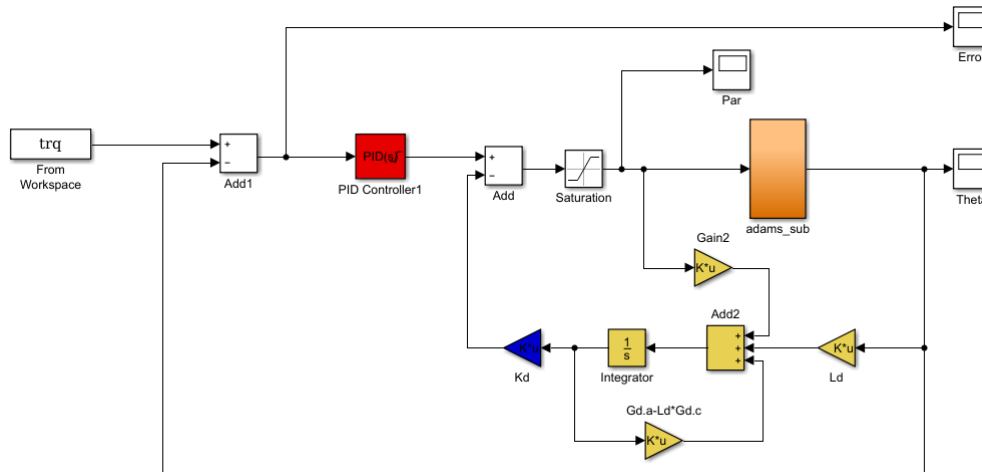


Figura 3.22: Esquema de control PID y LQR con observador de estados para un dedo.

place de *Matlab*, el cuál recibe los parámetros A y B del modelo de espacio de estados. La ganancia llamada Ld se halla como sigue:

$$Ld = place(A', C', po);$$

$$Ld = Ld';$$

Donde po son los polos deseados y se hallan en base en: $po = [5sig, 6 * sig]$, siendo $sigma$ igual a $max(real(plc))$ y $plc = eig(A - B * K_{ap})$. El comando *eig* calcula los valores propios de la matriz resultante de la operación $A - B * K_{ap}$. El comando *real* retorna la parte real de los elementos de plc , y el comando *max* devuelve los elementos mas grandes de plc .

Posteriormente se procede a hallar las constantes del controlador *PID* para el seguimiento de la consigna, mostrado en la figura 3.22 en color rojo. Para ello primero es necesario pasar del modelo en espacio de estados a función de transferencia. Las constantes Q_P , Q_I y Q_V son halladas como sigue:

$$Q_P = den(2)/num(3) * K$$

$$Q_I = den(3)/num(3) * K$$

$$Q_V = den(1)/num(3) * K$$

Donde K depende del tiempo de establecimiento, definido como $ts = 0,0058$ y $K = 1/(ts/6)$.

Para realizar el control de los 5 dedos de la prótesis de mano virtual se debe tener en cuenta que el bloque de *Adams_sub* de la figura 3.22 generado desde *Adams* (el cuál permitirá la comunicación de *Simulink* y *Adams*) debe tener 5 entradas y 5 salidas correspondientes a los pares de entrada y a los ángulos de salida respectivos de los cinco dedos de la prótesis de mano virtual (Ver figura 3.23).

Una vez generado el bloque de *Adams_sub* se procede a implementar 5 veces el control de la figura 3.22, teniendo en cuenta que las constantes de control para los

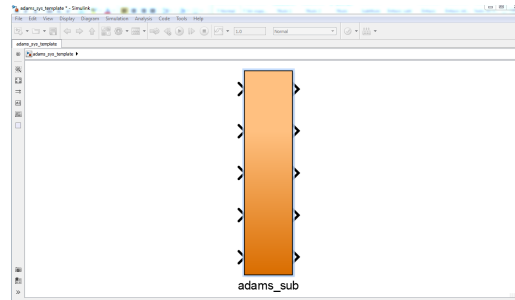


Figura 3.23: Bloque de *Adams_sub* generado desde Adams con 5 entradas y 5 salidas.

dedos Índice, Medio, Anillo y Meñique son exactamente iguales, ya que estos tienen el mismo modelo de espacio de estados. Es decir que las únicas constantes de control que son diferentes son las del dedo pulgar, ya que tiene un modelo dinámico diferente a los 4 dedos anteriormente mencionados.

Hasta aquí ya se tienen definidas las constantes del controlador, lo siguiente es definir una trayectoria grado cinco que permita a cada dedo realizar un movimiento suave hasta el ángulo de la articulación actuada deseado. Para el caso de la figura 3.24 se genera una trayectoria de ángulo de 0° a -30° para el dedo Índice. Para la figura 3.25, se genera una trayectoria de ángulo de 0° a 10° para el dedo Pulgar.

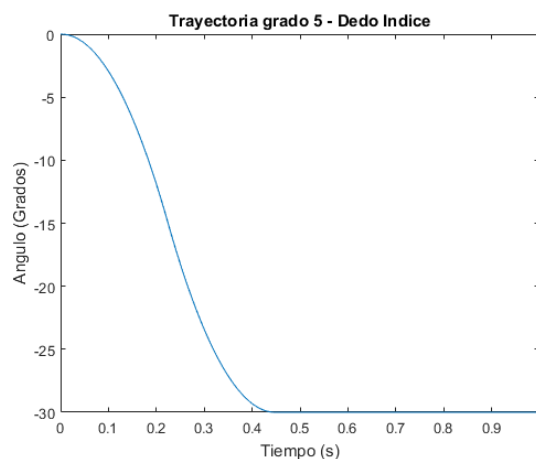
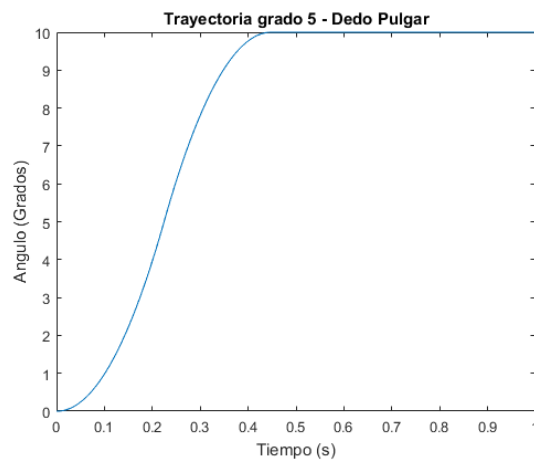
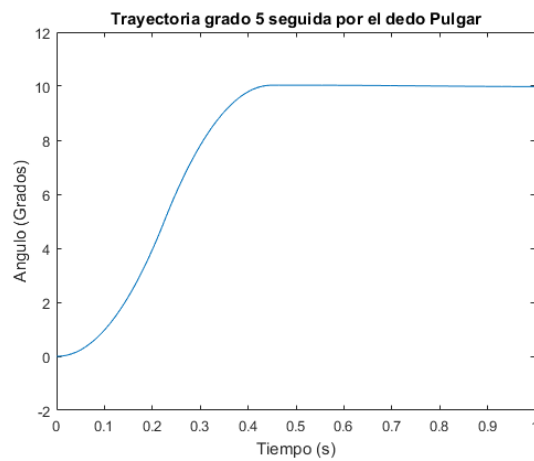
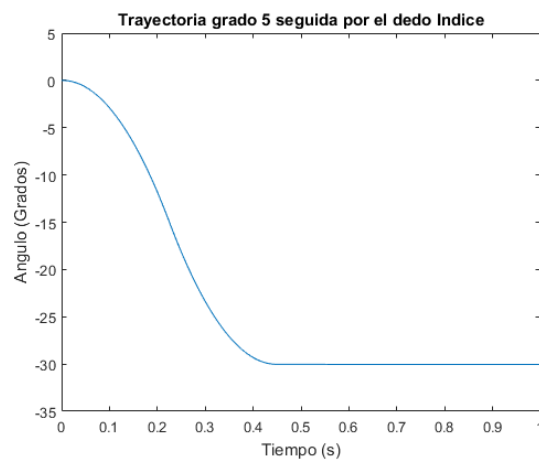


Figura 3.24: Trayectoria grado 5 para el dedo Índice, desde 0° hasta -30° .

Al realizar la simulación de control, se aprecia que el controlador sigue la consigna establecida para el dedo pulgar e índice (ver figuras 3.26 y 3.27 respectivamente), con un error de régimen permanente con tendencia a cero (ver figuras 3.28 y 3.29 respectivamente).

Figura 3.25: Trayectoria grado 5 para el dedo Pulgar, desde 0° hasta 10° .Figura 3.26: Trayectoria grado 5 seguida por el dedo Pulgar, desde 0° hasta 10° .Figura 3.27: Trayectoria grado 5 seguida por el dedo Indice, desde 0° hasta -30° .

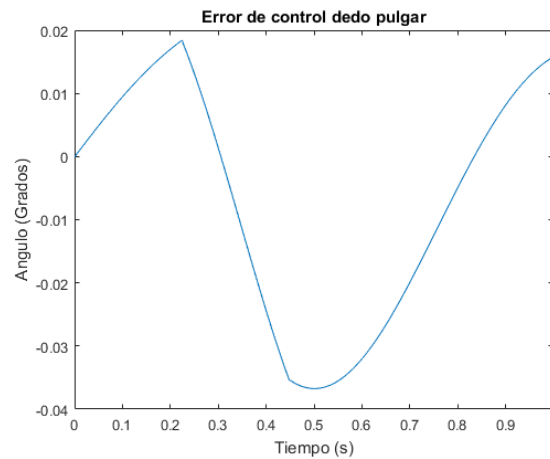


Figura 3.28: Error de seguimiento de la trayectoria grado 5 para el dedo Pulgar.

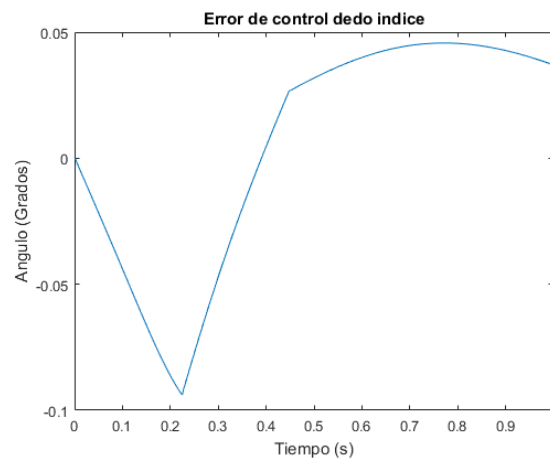


Figura 3.29: Error de seguimiento de la trayectoria grado 5 para el dedo Índice.

En la figura 3.30 se observa la posición final de los dedos de la prótesis de mano en *Adams* para un ángulo deseado de 10° para el dedo Pulgar y de -30° para el dedo Índice, Medio, Anillo y Meñique.

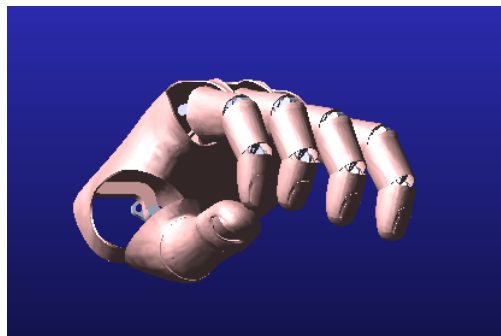


Figura 3.30: Prótesis de mano en *Adams* controlada en posición final de 10° para el dedo Pulgar y de -30° para el dedo Índice, Medio, Anillo y Meñique.

Finalmente para el control de la prótesis de mano en el agarre de objetos, la idea principal es establecer como consigna de los dedos, los ángulos que permiten cerrar la mano. Usando las mismas consignas mencionadas anteriormente (10° para el dedo Pulgar y de -30° para el dedo Índice, Medio, Anillo y Meñique), se pretende que el objeto este situado en el centro de la palma. Los dedos nunca llegarán a la consigna deseada ya que el objeto se interpone en su camino, pero de esta forma se logra que la prótesis agarre el objeto. El controlador en su intento por llegar a la consigna deseada aumentará el par hasta donde se establezcan los límites de saturación (establecido en 200 dyn.m). En la figura 3.31 se observa la prótesis de mano agarrando una esfera de 40 mm de radio, y en las figuras 3.32, 3.33, 3.34, 3.35 y 3.36 se observa el valor de par entregado por el controlador para los dedos Pulgar, Índice, Medio, Anillo y Meñique respectivamente. Por otro lado en las figuras 3.37, 3.38, 3.39, 3.40 y 3.41 se observa el error de la trayectoria de referencia para los dedos Pulgar, Índice, Medio, Anillo y Meñique respectivamente.

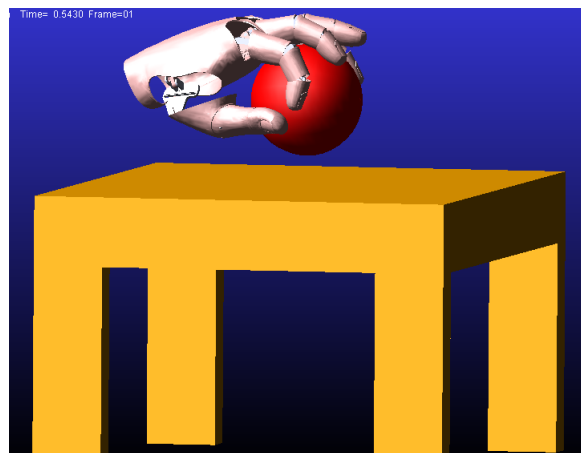


Figura 3.31: Prótesis virtual de mano agarrando una esfera de 40 mm

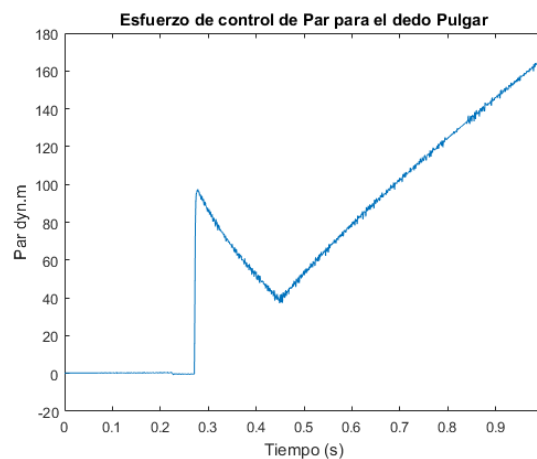


Figura 3.32: Esfuerzo de control del dedo Pulgar en el agarre de una esfera.

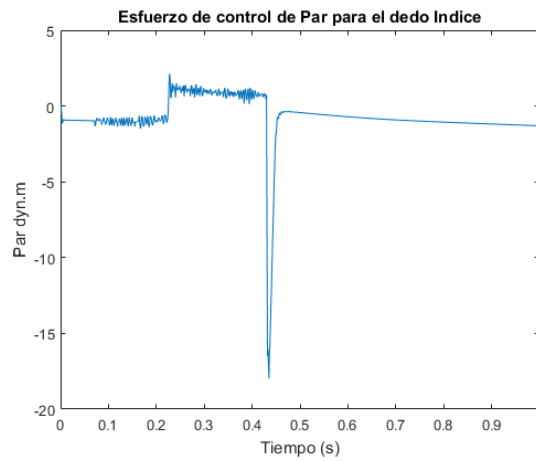


Figura 3.33: Esfuerzo de control del dedo Índice en el agarre de una esfera.

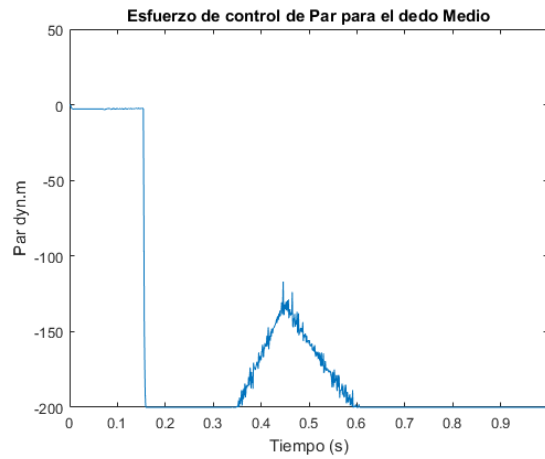


Figura 3.34: Esfuerzo de control del dedo Medio en el agarre de una esfera.

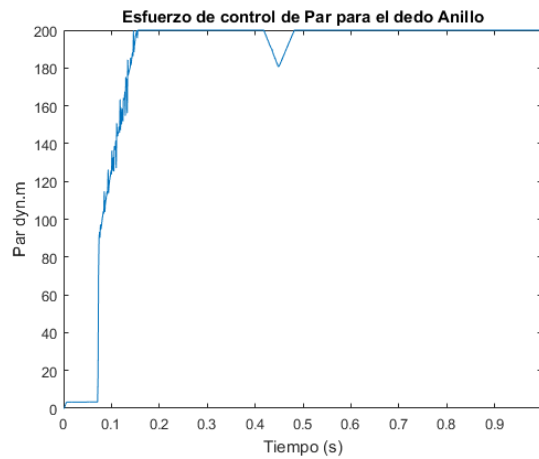


Figura 3.35: Esfuerzo de control del dedo Anillo en el agarre de una esfera.

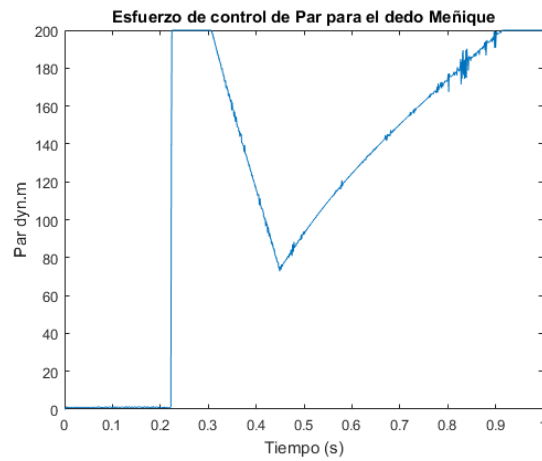


Figura 3.36: Esfuerzo de control del dedo Meñique en el agarre de una esfera.

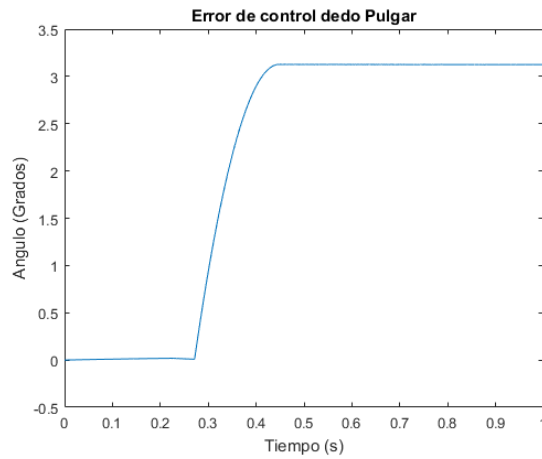


Figura 3.37: Error de seguimiento en trayectoria del dedo Pulgar en agarre de objeto.

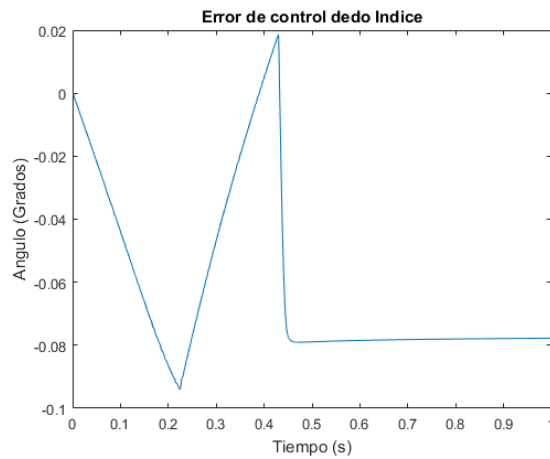


Figura 3.38: Error de seguimiento en trayectoria del dedo Índice en agarre de objeto.

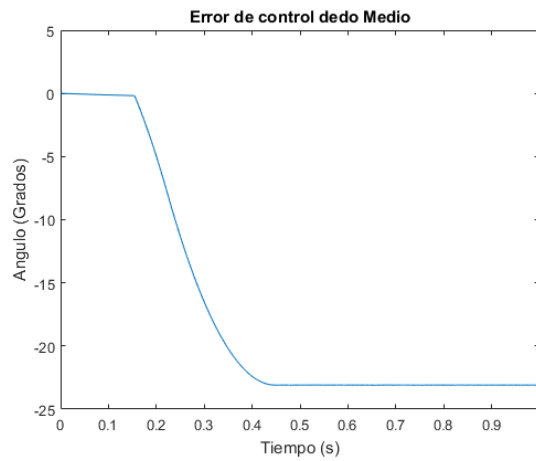


Figura 3.39: Error de seguimiento en trayectoria del dedo Medio en agarre de objeto.

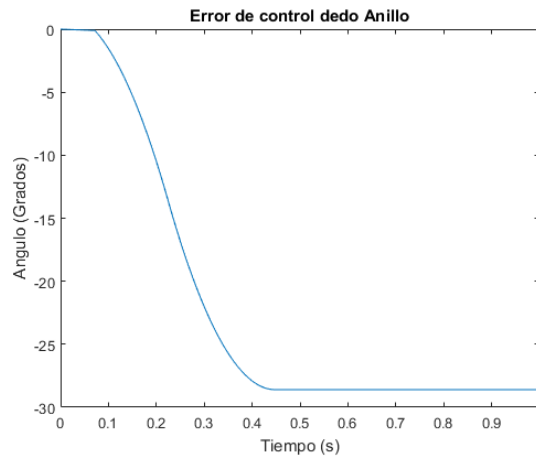


Figura 3.40: Error de seguimiento en trayectoria del dedo Anillo en agarre de objeto.

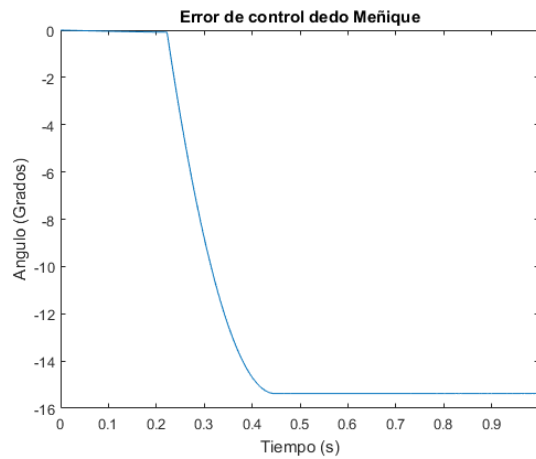


Figura 3.41: Error de seguimiento en trayectoria del dedo Meñique en agarre de objeto.

3.6.2.1. Conclusión

Se logra realizar flexión y extensión de una prótesis de mano virtual de 5 dedos usando 5 actuadores.

El control basado en el modelo dinámico de la planta resulta ser más preciso que el control no basado en modelo.

Se logra realizar flexión y extensión de una prótesis de mano mediante la utilización de una sola trayectoria de referencia para cada dedo.

Se logra controlar el seguimiento de trayectorias para una prótesis de mano virtual con un error de estado estable aceptable.

Se logra realizar el agarre de objetos de la prótesis virtual de mano cuando el mecanismo auto-adaptable no está activado.

Adams es una gran herramienta computacional para realizar tipos de análisis dinámicos multi-cuerpos, ya que fue creado precisamente para trabajar problemas en este campo de aplicación. Además, se observó grandes ventajas a la hora de la aplicación a este trabajo de grado, como lo fueron la variedad de sensores para realizar mediciones, una interfaz sencilla, su plugin de control, la importación de modelos CAD prediseñados en múltiples formatos adicionalmente, la facilidad con la que se puede realizar la co-simulación con MATLAB para crear sistemas complejos de control.

Capítulo 4

Gobierno con EMG

El objetivo de este capítulo es exponer qué requerimientos se necesitan y cómo se implementan, de tal forma que se logre un control sencillo de una prótesis virtual de mano a partir de un solo canal de señales de electromiografía superficial.

4.1. Teoría

4.1.1. Señales electromiográficas (EMG)

En el músculo podemos distinguir entre 2 unidades principalmente, la unidad anatómica y la unidad funcional; la primera es la llamada fibra muscular, mientras que la segunda recibe el nombre de unidad motora. La unidad motora (UM) (Figura 4.1) es un grupo de fibras musculares inervado por una sola neurona motora. Realizar un simple movimiento requiere la intervención de muchas neuronas tanto sensoriales como motoras, que trabajando en conjunto logran ejecutar la acción indicada; el funcionamiento es el siguiente: se genera la orden en el cerebro en forma de un impulso eléctrico; este impulso viaja a través de la espina dorsal y llega hasta las motoneuronas responsables de inervar el músculo que realizará la acción estimulando las fibras musculares, produciendo así la expansión o inhibición del músculo. Cabe mencionar que para realizar un movimiento complejo, digamos la extensión o flexión del brazo, intervienen muchas motoneuronas, y no puede lograrse solo estimulando un solo músculo, el cual está ligado a más de una UM.

4.1.1.1. Electromiografía (EMG)

La electromiografía es la medición de potenciales eléctricos creados por la contracción de los músculos. Internamente, los músculos generan tensiones alrededor de 100mV cuando se contraen. Estas tensiones son muy atenuadas por el tejido interno y la piel, ya que son débiles pero medibles en la superficie de la piel. Las señales EMGs, son colectadas típicamente mediante electrodos bipolares de superficie, ubicados sobre la piel, como se observa en la figura 4.2. Estas señales han sido utilizadas para el control de prótesis de miembros superiores desde 1948. Cabe destacar que entre la electromiografía destacan dos técnicas principales para la adquisición de los potenciales generados por las unidades motoras, estas técni-

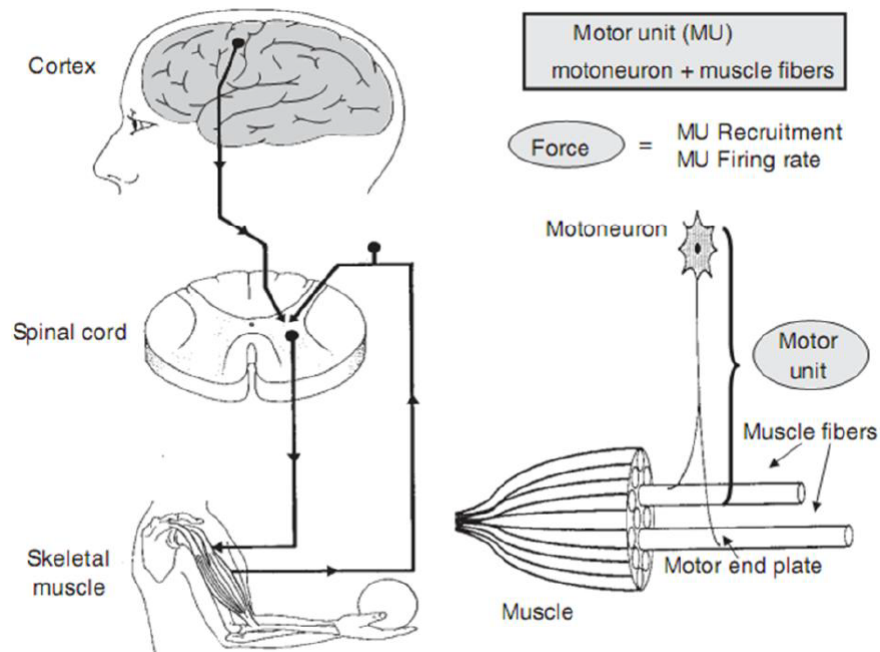


Figura 4.1: Esquema representativo del mecanismo básico de control y de los componentes de la unidad motora. Tomado de [6].

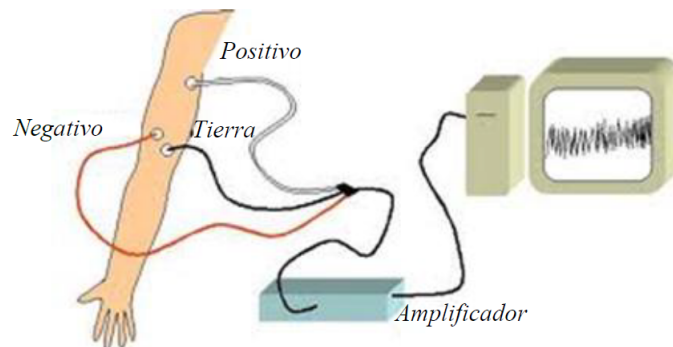


Figura 4.2: Detección de señales mioeléctricas. Tomado de [7].

cas son la electromiografía invasiva y la electromiografía de superficie (EMGs o SEMG), llamadas así por el tipo de electrodos que utilizan.

4.1.1.2. Electrodos

La manera de obtener información acerca de nuestro entorno y transferirla a algún aparato electrónico se lleva a cabo mediante un transductor, un transductor es un dispositivo capaz de transformar un tipo de energía de entrada a otro tipo de energía de salida. En el campo de la bioelectricidad, los transductores utilizados son llamados electrodos; los electrodos hacen una transferencia iónica del tejido vivo del cuerpo hacia un dispositivo electrónico, el cual se encarga de procesarla para posteriormente obtener información útil de la medición; entre las señales biológicas más estudiadas y registradas se encuentran las Electrocardiográficas (ECG), Elec-

troencefalográficas (EEG), electromiográficas (EMG), por citar algunas.

Los electrodos se clasifican en invasivos, si penetran la piel hasta estar adjuntos a los músculos, o no invasivos si se localizan en la superficie de la piel. Estos últimos pueden ser secos o húmedos, los cuales utilizan entre el electrodo y la piel ya sea una sustancia electrolítica, o un gel conductor.

Los materiales de los que se constituyen la mayoría de los electrodos de superficie son muy variables, entre ellos están la plata, el oro, acero inoxidable, platino entre otros. Una de las condiciones deseables en un electrodo, es que no sea polarizado, esto significa que el potencial en el electrodo no debe de variar considerablemente cada vez que la corriente pase a través de él; el electrodo de plata-cloruro de plata (Ag/AgCl) ha demostrado tener los estándares adecuados para lograr esta característica; además, la interfaz piel-plata o piel-Ag/AgCl tiene la mayor impedancia resistiva en el dominio de la frecuencia en EMG.

4.1.1.2.1. Perturbación de la señal captada del sensor sEMG

Los factores que influyen en la EMGs tienen causas intrínsecas y extrínsecas, siendo el último manejable por el usuario y según la tecnología usada puede ser atenuado o relativamente eliminado. Los factores intrínsecos, son los elementos de perturbación de que no puede de ninguna forma ser controlados por el usuario. Entre los que más destacan están las señales que se superponen, como el electrocardiográfico, presión sanguínea, tejido subcutáneo, entre otros muchos. En todo caso, para obtener una buena muestra de la señal EMGs se recomienda que el paciente debe estar en reposo para atenuar el ruido artefacto, el cual también disminuiría el ritmo cardiaco, con esta lógica es una manera de evitar la perturbación ante una evaluación pero estos nunca serán eliminados, solo atenuados.

Entre las fuentes de perturbaciones existentes en una sEMG se distinguen: ruido fisiológico, ruido del ambiente, ruido línea base, ruido de artefacto de movimiento.

4.1.1.3. Requisitos y recomendaciones para el estudio con electrodos EMGs

Uno de los puntos más discutidos en la EMG de superficie es la localización de los electrodos. Debido a esto, se originó una iniciativa europea para tratar de estandarizar estos factores; localización, tamaño y forma de los electrodos; es así que en 1996 surge el SENIAM (Surface Electromyography for Noninvasive Assessment of Muscles) para tratar de dar ciertas recomendaciones en cuanto a estas variables.

Según las recomendaciones producidas por el SENIAM, indica que las medidas de un electrodo en su diámetro debe ser menor o igual a 10mm y la separación entre ambos electrodos debe ser menor o igual que 20 mm. Por lo cual se llegó a adecuar los electrodos utilizados en ECG modificando su parte adhesiva mas no sobre el electrolito el cual se mantiene en su totalidad.

El SENIAM proporciona una serie de pasos antes de realiza las mediciones de

SEMG:

- Selección de los electrodos para SEMG.
- Preparación de la piel, esto implica ya sea la aplicación de algún gel conductor o limpiar con alcohol la zona donde se van a colocar los electrodos.
- Posicionar la paciente en la postura inicial, esta postura puede variar dependiendo del estudio a realizar.
- Determinar la localización de los electrodos.
- Fijar los electrodos.
- Finalmente testear las conexiones.

4.1.1.3.1. Consideraciones generales

La calidad de una medición EMG depende en gran medida de la preparación adecuada de la piel y del posicionamiento de los electrodos.

La principal estrategia de preparación de la piel es el contacto estable del electrodo y la baja impedancia de la piel. La mayoría de los amplificadores EMG modernos están diseñados para niveles de impedancia de la piel entre 5 y 50 kOhm (entre pares de electrodos).

Por lo general, es necesario realizar alguna preparación de la piel antes de aplicar los electrodos. No existen reglas generales para ello y existen varias posibilidades para alcanzar una buena condición de la piel para las mediciones de EMG. Especialmente para principiantes será de gran valor comprobar la calidad del método elegido midiendo la resistencia de impedancia real entre los electrodos con un multímetro regular o medidores de impedancia especializados.

Otra consideración importante es la condición de prueba específica y el ejercicio. Si se planifica un movimiento estático o en cámara lenta (por ejemplo, una prueba de función muscular clínica) y la idea básica de análisis es cualitativa (la amplitud cambia en términos de más / menos), una simple limpieza con alcohol puede ser suficiente. Si se planifican condiciones muy dinámicas con riesgo de artefactos de movimiento (por ejemplo, caminar rápido, correr u otros movimientos altamente acelerados), es imprescindible una preparación muy completa.

Los siguientes procedimientos se pueden considerar como pasos para preparar la aplicación de electrodos:

- Quitar el cabello: Esto es necesario para mejorar la adhesión de los electrodos, especialmente bajo condiciones húmedas o para tipos de piel sudorosa y / o condiciones de movimiento dinámico.

- Limpieza de la piel:

Método A: Se encuentran disponibles pastas limpiadoras especiales abrasivas y conductoras que eliminan las células muertas de la piel (producen alta impedancia) y limpian la piel de la suciedad y el sudor.

Método B: Alternativamente, se puede usar papel de lija muy fino: una presión suave y controlada en 3 o 4 barridos usualmente es suficiente para obtener un buen resultado. Atención: ¡evite que cualquier daño a la piel se frote demasiado! El uso de papel de lija debe combinarse con una almohadilla de alcohol.

Método C: El uso puro de alcohol puede ser otra alternativa si se usa con una toalla textil (que permite frotar suavemente). Este último método puede ser suficiente para pruebas de función muscular estática en condiciones fáciles.

Cualquiera que sea el método de preparación de la piel y la técnica de aplicación del electrodo, cuando se realiza correctamente, la piel normalmente recibe un color rojo claro. Esto indica una buena condición de impedancia de la piel.

4.1.1.4. Procesamiento de la señal- Normalización de amplitud

Un gran inconveniente de cualquier análisis EMG es que los datos de amplitud (escala de microvoltios) están fuertemente influenciados por la condición de detección: puede variar fuertemente entre sitios de electrodos, entre sujetos e incluso medidas diarias del mismo sitio muscular sobre el mismo sujeto. Una solución para superar este carácter "incierto" de los parámetros a escala de micro voltios es la normalización del valor de referencia, por ejemplo, el valor máximo de contracción voluntaria (MVC) de una contracción de referencia. La idea básica es calibrar el valor de microvoltios a una unidad de calibración única con relevancia fisiológica, el "porcentaje de la capacidad máxima de inervación" en ese sentido particular. Otros métodos normalizan al valor medio interno o a una prueba determinada o al nivel EMG de una determinada actividad de referencia submáxima. El efecto principal de todos los métodos de normalización es que se elimina la influencia de la condición de detección dada y los datos se cambian de escala de microvoltios a porcentaje del valor de referencia seleccionado. Es importante entender que la normalización de la amplitud no cambia la forma de las curvas EMG, ¡solo su escala del eje Y!

4.1.1.4.1. El concepto de MVC Normalización

El método más popular se llama MVC-normalización, que se refiere a una contracción voluntaria máxima realizada antes de los ensayos de prueba (figura 4.3).

Típicamente, las contracciones MVC se realizan contra la resistencia estática. Para realmente producir una inervación máxima, una muy buena fijación de todos los segmentos implicados es muy importante. Los sujetos normales (no entrenados) pueden tener problemas para producir un verdadero nivel de contracción MVC, sin estar acostumbrados a tales esfuerzos.

4.1.1.4.2. La práctica de la Normalización MVC

La prueba MVC se debe realizar para cada músculo investigado por separado. El primer paso es identificar un ejercicio / posición que permita una inervación máxima

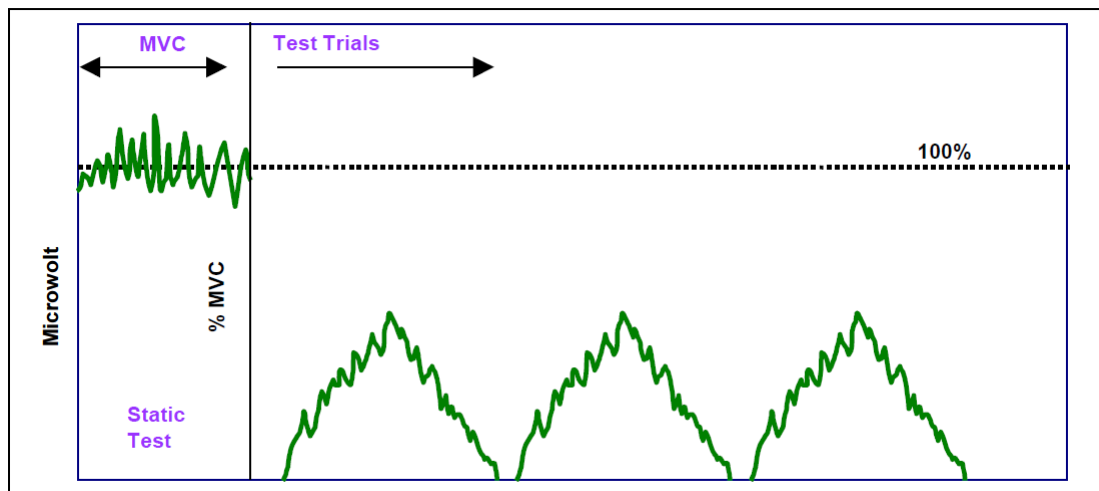


Figura 4.3: El concepto de normalización MVC. Antes de la prueba / ejercicios, se realiza una contracción MVC estática para cada músculo. Este nivel de inervación MVC sirve como nivel de referencia (= 100 %) para todos los ensayos futuros. Tomado de [8].

efectiva (¡no forzar la salida!).

Es interesante notar que dependiendo de la capacidad de coordinación individual del sujeto, diferentes ejercicios / posiciones de prueba pueden producir el mayor valor de MVC. Por lo anterior puede ser necesario probar dos o tres posibles buenos candidatos de los ejercicios de prueba y verificar, donde finalmente se puede encontrar el nivel más alto de EMG.

Después de una secuencia de calentamiento inicial (estiramiento, ejercicios aeróbicos bajos, de 5 a 10 minutos), se pide a los sujetos que comiencen a aumentar lentamente la fuerza, alcancen el esfuerzo máximo después de 3 - 5 segundos, sostener durante 3 segundos y calmar con 3 segundos. Repetir lo anterior al menos una vez, con un período de pausa de 30 a 60 segundos en el medio. Repetir esta secuencia para cada ejercicio de MVC (los estudios requieren orden aleatorio para evitar los efectos sistemáticos de la fatiga).

4.1.1.4.3. Propuesta para MVC - posiciones de prueba

el punto más importante es una muy buena fijación y contracción contra la resistencia rígida. Algunos ejercicios se pueden organizar en un banco de terapia regular, usando cinturones en combinación con resistencia manual. Las máquinas de entrenamiento pueden ser muy útiles debido a sus posibilidades de fijación. Los estudios de investigación sistemática sobre la efectividad de las posiciones de MVC aún faltan y se necesitan algunas pruebas de prueba y error en los estudios previos para confirmar la mejor disposición de prueba [8].

En la figura 4.4 se observa una posición recomendada para obtener la Máxima Contracción Voluntaria (MVC) de los músculos Flexores / extensores del antebrazo.

Se recomienda seleccionar una posición sentada o arrodillada (frente a un banco) y colocar un soporte estable para el antebrazo. Se pueden usar resistencias manuales, pesas o cables / cinturones.

Finalmente, uno de los beneficios de la normalización MVC es el cambio al porcentaje de un valor de referencia único y estandarizado para todas las materias dentro de un estudio. Elimina cualquier influencia variable de las condiciones locales de detección de señal. Además, permite una comparación cuantitativa directa de los hallazgos EMG entre sujetos. Las estadísticas grupales y los datos normativos se pueden desarrollar y verificar estadísticamente.

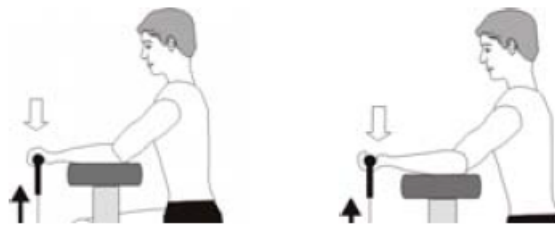


Figura 4.4: Propuesta para el arreglo de prueba MVC del brazo. La flecha delgada negra indica la dirección del movimiento, las flechas gruesas blancas la dirección de la resistencia. Tomado de [8].

4.1.1.5. Sistema básico de adquisición de señales EMG

Las señales EMGs son generadas por la contracción muscular, por lo que su adquisición requiere de una correcta identificación de las regiones musculares comprometidas en la ejecución de los movimientos de interés. Las señales recogidas por los electrodos serán demasiado débiles, por lo que se hace necesario un procesamiento previo de amplificación y filtrado antes de su análisis. Ver figura 4.5.

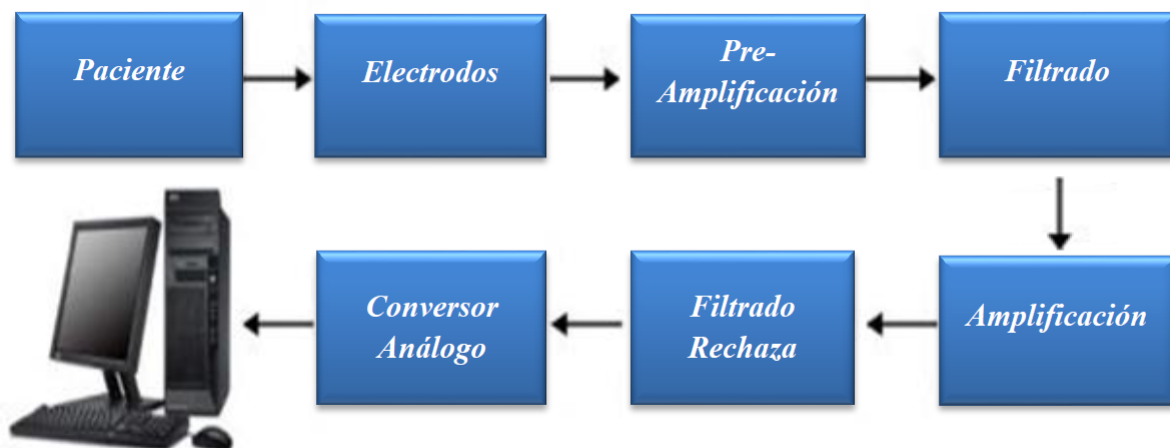


Figura 4.5: Sistema básico de adquisición de señales EMG. Tomado de [7].

4.2. Requerimientos

Los requerimientos son:

- Realizar la adquisición de señales electromiográficas mediante un solo canal, haciendo uso de electrodos no invasivos.
- Realizar lectura de electrodos de EMGs en tiempo real.
- Definir trayectorias de control de posición para los 5 dedos de la prótesis de mano virtual a partir de las señales EMGs.
- Adquirir las señales electromiograficas mediante una tarjeta E-Health.

4.2.1. E-Health Sensor Platform V2.0

E-Health Sensor Platform V2.0: Es una tarjeta que permite llevar a cabo aplicaciones para el monitoreo del cuerpo humano, ofrece la posibilidad de conexión con distintos sensores: temperatura corporal, glucómetro, respuesta galvánica de la piel, saturación de oxígeno en la sangre (SpO₂), respiración, posición del paciente (acelerómetro), electromiografía, y la presión arterial. Se puede utilizar para monitorizar en tiempo real el estado de un paciente o para obtener datos con el fin de ser analizados posteriormente, pero necesita de una tarjeta adicional para esto ya que por sí sola la e-Health Sensor Platform V2.0 no puede procesar completamente los datos provenientes de los sensores. Esta tarjeta es de hardware abierto y es compatible con Arduino y Raspberry Pi.

4.3. Implementación

4.3.1. Hardware

Para la implementación de un sistema de adquisición señales EMGs en un solo canal, se utilizó la tarjeta E-Health Sensor Platform V2.0 y una tarjeta Arduino Mega 2560.

Los electrodos utilizados en este proyecto son electrodos ECG húmedos, compuestos de Ag-AgCl (plata-cloruro de plata) con un radio de 55 mm. Son utilizados electrodos de ECG ya que solo difieren de los EMG por las dimensiones de su parte adhesiva.

Por otra parte, el primer paso para implementación del sistema de adquisición de señales es ensamblar la placa E-Health Sensor Platform V2.0 a la placa Arduino Mega 2560, posteriormente conectar los sensores de electromiografía en la placa E-Health Sensor Platform V2.0 y finalmente conectar los sensores a la piel de los usuarios que realizan el agarre de objetos con la prótesis de mano virtual.

El ensamblaje entre la tarjeta E-Health Sensor Platform V2.0 y Arduino Arduino Mega 2560 (Ver figura 4.7), la conexión de los electrodos a la placa E-Health Sensor

Platform V2.0, la conexión de los electrodos al brazo de las personas no amputadas y la adquisición de señales EMG se puede observar en detalle en el apéndice B.

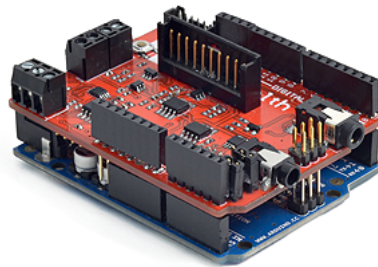


Figura 4.6: Ensamblaje de placa E-Health y Arduino.

4.3.2. Software

En la implementación software lo primero que se debe realizar es un código en Arduino (ir a apéndice B) que permita la adquisición de las señales EMGs mediante la utilización de la librería oficial **eHealth.h**. Al utilizar esta librería es posible obtener valores de 0 a 1024 para una lectura de sensores EMGs.

Posteriormente, con el fin de adquirir las señales en tiempo real, se implementa la lectura de estas señales en simulink, mediante el bloque *Query Instrument*. Este bloque realiza la lectura del puerto COM del computador al cuál está conectada la Placa de Arduino.

4.3.3. Estrategia umbral flexión-extensión

Con el fin de lograr la flexión y extensión de cada uno de los dedos de la prótesis virtual de mano en el agarre de objetos, se utiliza una estrategia que permite definir la trayectoria deseada de control para cada articulación activa en función de la señal electromiografica capturada desde usuarios sin amputación de miembro superior.

Es importante aclarar que la simulación de la prótesis virtual inicia con la mano abierta (Angulo de cada articulación activa en 0°). Cuando el usuario desee cerrarla para agarrar un objeto, debe realizar una contracción de los músculos de su mano sostenida durante 3 segundos, relajar inmediatamente y observar que la mano continua cerrándose pese a la relajación de los músculos de su mano. Para cerrarla debe oprimir un Switch virtual que permite que la mano se cierre.

Para lo anterior, se debe realizar a cada usuario, pruebas previas al agarre de objetos con la prótesis virtual de mano, con el objetivo de encontrar un valor de umbral entre 0 y 1024 que indice la mitad de la máxima contracción muscular.

Una vez obtenido el valor de umbral, se procede a realizar a un usuario, una prueba en la que agarre objetos con una prótesis virtual. Para lograr el agarre, se solicita al usuario realizar una máxima contracción voluntaria sostenida durante 3 segundos,

con lo cual, al comparar si la fuerza que ejerce es mayor que el umbral de fuerza, se ordena a la prótesis cerrar la mano al seguir la trayectoria de ángulo de la articulación activa de, 0° a 10° para el dedo Pulgar y de 0° a -30° para el dedo índice, medio, Anular y Meñique.

Por otro lado, para lograr que la mano se abra, se pide al usuario que oprima un switch virtual, el cuál ordena a la prótesis abrir la mano mediante la trayectoria de regreso hasta que cada articulación activa esté nuevamente en 0° .

Es importante mencionar que la trayectoria de abrir y cerrar la mano es una sola, y la decisión de cerrar y abrir se convierte en un problema binario de la siguiente manera: Sí el usuario supera el valor de umbral se decide utilizar la trayectoria de cerrar los dedos, sí el usuario oprime el switch se abre la mano regresando a su postura inicial, moviéndose en sentido contrario de la trayectoria de cierre, hasta establecer el ángulo de todas las articulaciones activas en 0.

El esquema de implementación de control en Simulink, es similar al de la figura 3.22, la cual solo difiere en que la trayectoria de referencia (trq) es definida mediante el bloque *Query Instrument* y una serie de decisiones (nombradas anteriormente) implementadas en el bloque de *Matlab Function* y *S - Function*. Ver figura 4.7.

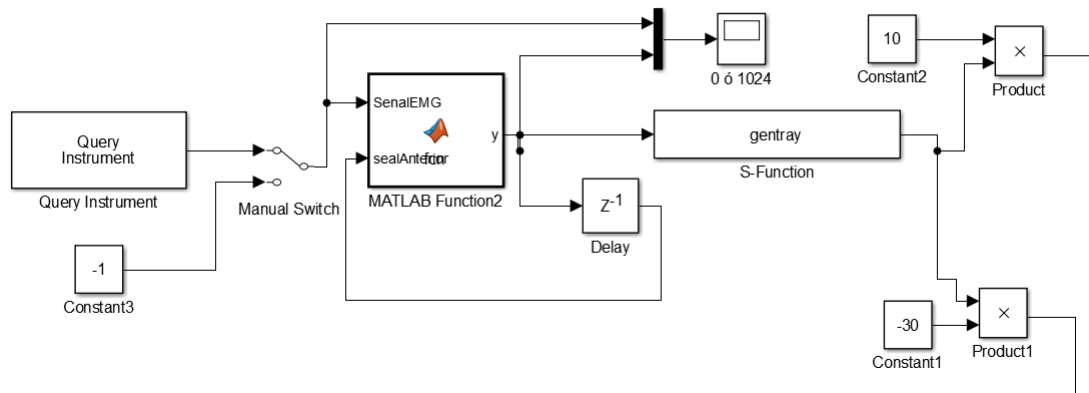


Figura 4.7: Definición de trayectoria de referencia a partir de señales EMGs en Simulink.

La señal de la figura 4.7, después del bloque *product* precedido de la constante 10 es la trayectoria de referencia de 0° a 10° para el dedo Pulgar. Similarmente, la señal de salida del bloque *product* precedido de la constante -30 es la trayectoria de referencia de 0° a -30° para el dedo índice, medio, Anular y Meñique. Estas dos señales son entrada al bloque de PID del control de cada dedo.

4.4. Objetivos no alcanzados

Uno de los principales objetivos no alcanzados, fue el de realizar el agarre de objetos con la prótesis virtual de mano a partir de señales EMGs en tiempo real cuando

los mecanismos auto adaptables están activos, debido a:

- La simulación de las articulaciones de perno en una ranura en el software Adams requiere el uso de fuerzas de contacto para que una barra se deslice sobre la otra, hecho que incrementa la carga computacional y la simulación demora horas para agarrar un solo objeto.
- Diseño del mecanismo auto-adaptable incompleto.

Sin embargo, la simulación de articulaciones de perno en el software Adams no requieren una alta carga computacional, por lo que todas las simulaciones de control en tiempo real a partir de señales EMGs fueron posibles únicamente con los mecanismos sub-actuados de la prótesis de mano virtual.

4.5. Conclusión

Se logra una flexión y extensión de los 5 dedos una prótesis de mano virtual mediante un solo canal de electromiografía.

La orden de trayectoria a generar para cada dedo es establecida por una señal EMGs.

Se utiliza una sola trayectoria por cada dedo para realizar flexión-extensión en el agarre de objetos.

Cada dedo sigue una trayectoria única, esto debido a que el diseño del mecanismo permite transmitir el movimiento desde una sola articulación actuada hasta el efector final del dedo (yema del dedo).

El control con EMGs para una prótesis virtual de mano se simplifica gracias al diseño subactuado del mecanismo de cada dedo.

Adams no es una buena opción para proyectos de simulación en tiempo real, ya que la agregación de fuerzas de contacto y otras propiedades físicas aumentan la carga computacional generando una simulación demorada.

Capítulo 5

Evaluación de aceptación de usuario

El objetivo principal de este capítulo es evaluar cuantitativamente el grado de complejidad percibido por una muestra de usuarios sin amputación de miembro superior, para el gobierno del agarre de objetos de diferentes formas y tamaños sobre el entorno virtual, usando contracciones musculares captadas desde sensores de EMG superficiales.

Lo anterior, con el fin de evidenciar que el diseño mecánico de los dedos de la prótesis, facilita el control del agarre de objetos.

5.1. Teoría

5.1.1. Usabilidad y facilidad de uso

En el campo del diseño e investigación enfocados a aplicaciones de software y hardware, es necesario realizar pruebas en las que se busca evaluar de forma cuantitativa la aceptabilidad del usuario en cuanto al uso de un producto en específico.

Para evaluar la aceptabilidad del usuario es necesario realizar cuestionarios y/o encuestas, las cuales están estrechamente ligadas a conceptos tales como: usabilidad y facilidad de uso ([21] y [22]).

La facilidad de uso percibida se refiere a el grado en que una persona cree que usar un sistema en particular estaría libre de esfuerzo. Esto se desprende de la definición de facilidad: libertad de un gran esfuerzo difícil.

Adicionalmente algunas de las definiciones entregadas por la Organización Internacional para la Estandarización (ISO) para la usabilidad, son:

- La ISO 9241 – 11 define la usabilidad como: “Usabilidad es la eficacia, eficiencia y satisfacción con la que un producto/servicio permite alcanzar objetivos específicos a usuarios específicos en un contexto de uso específico”. Es la norma enfocada a la calidad en usabilidad y ergonomía tanto de hardware como de software, esto quiere decir la capacidad que tiene un producto software

para ser utilizado con facilidad por los usuarios y su capacidad de adaptación a estos.

- La ISO 9126 – 1 define la usabilidad como: "La Usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso". En esta definición se define la usabilidad como un atributo de calidad de software esto quiere decir la capacidad de un producto software para ser utilizado con facilidad al igual que en la ISO anteriormente mencionada.

Por otro lado, la complejidad se define como el grado en que una innovación se percibe como relativamente difícil de entender y usar.

El concepto de usabilidad del dispositivo, considerado como un aspecto de utilidad, es un atributo cualitativo que evalúa la facilidad de uso de las interfaces de dispositivo-usuario.

5.1.1.1. Atributos de la usabilidad

La usabilidad tiene varios componentes y está tradicionalmente asociada con estos cinco atributos de usabilidad [23]:

- **Facilidad de aprendizaje:** El sistema debe ser fácil de aprender para que el usuario logre realizar las tareas con rapidez.
- **Eficiencia:** El sistema debe ser eficiente en su uso, de manera que una vez que el usuario aprenda como funciona pueda lograr la mayor productividad posible.
- **Recuerdo en el tiempo:** El sistema debe ser fácil de recordar, de manera que el usuario ocasional sea capaz de volver al sistema después de un periodo de no haberlo utilizado sin tener que aprender todo de nuevo.
- **Errores:** El sistema debe tener una baja tasa de errores, los usuarios deben cometer la menor cantidad de errores posibles mientras dure la utilización del sistema, en caso de error se busca que el usuario pueda recuperarse fácilmente. No deben ocurrir errores catastróficos.
- **Satisfacción:** El sistema debe ser agradable de usar, de manera que los usuarios se encuentren satisfechos. Este aspecto es el más subjetivo debido a que cada usuario tiene grados de satisfacción y necesidades diferentes.

5.1.2. Escala de medida

Las encuestas son una de las herramientas más utilizadas por los investigadores para evaluar las opiniones y actitudes de una persona frente a una innovación, producto o servicio. Existen varios tipos de escalas de medición enfocadas directamente a medir las actitudes de las personas, entre ellas, una de las más utilizadas es de la escala de Likert.

5.1.2.1. Escala tipo Likert

Es una escala psicométrica utilizada principalmente en la investigación de mercados para la comprensión de las opiniones y actitudes de un consumidor hacia una marca, producto o mercado meta. Sirve principalmente para realizar mediciones y conocer sobre el grado de conformidad de una persona o encuestado hacia determinada oración afirmativa o negativa.

Cuando se responde a un ítem de la escala de Likert, el usuario responde específicamente en base a su nivel de acuerdo o desacuerdo.

La escala de Likert asume que la fuerza e intensidad de la experiencia es lineal, por lo tanto va desde un totalmente de acuerdo a un totalmente desacuerdo, asumiendo que las actitudes pueden ser medidas. Las respuestas pueden ser ofrecidas en diferentes niveles de medición, permitiendo escalas de 5, 7 y 9 elementos configurados previamente. Siempre se debe tener un elemento neutral para aquellos usuarios que no están ni en acuerdo ni en desacuerdo [9].

En la figura 5.1 se observa un ejemplo de oración a la cuál se debe responder con un nivel de acuerdo o desacuerdo:

Creo que la innovación es el elemento más importante a la hora de iniciar una startup.

Ejemplo de la escala de Likert

Totalmente en desacuerdo <input type="radio"/>	En desacuerdo <input type="radio"/>	Ni de acuerdo, ni en desacuerdo <input type="radio"/>	De acuerdo <input type="radio"/>	Totalmente de acuerdo <input type="radio"/>
------------------------------------------------------	----------------------------------------	-------------------------------------------------------------	-------------------------------------	---------------------------------------------------

Figura 5.1: Ejemplo de escala de Likert. Tomado de [9].

La escala de Likert tiene varias adaptaciones. Una de ellas, es no tomar dentro de la escala el elemento neutral (ni de acuerdo ni en desacuerdo). Además de ello, agregar un valor numérico a la escala, esto es: 1 " para totalmente de acuerdo " , 2 para " en desacuerdo " , 3 para " de acuerdo " y 4 para " totalmente de acuerdo " . De esta manera se facilita el análisis estadístico, permitiendo procedimientos no paramétricos basados en el rango y la mediana, así como métodos libres de distribución tales como las tabulaciones, frecuencias tablas de contingencia y estadísticas de chi cuadrado.

Un ejemplo de esta variación se observa en la figura 5.2, en la que se realiza una pregunta y se presenta la escala de respuesta.

Parte	Pregunta	Escala	
Registro	Proporciona un distintivo visual los campos de datos en:	1. Muy bajo grado 2. Bajo grado	3. Alto grado 4. Muy alto grado
Información del Sistema	Organización de la información de la interfaz es:	1. Muy confusa 2. Confusa	3. Clara 4. Muy clara

Figura 5.2: Ejemplo de una adaptación a la escala de Likert. Tomado de [9].

Finalmente, en la figura 5.3 se presenta una variación de la escala de Likert en la que no se contempla el elemento neutro de la figura 5.1.

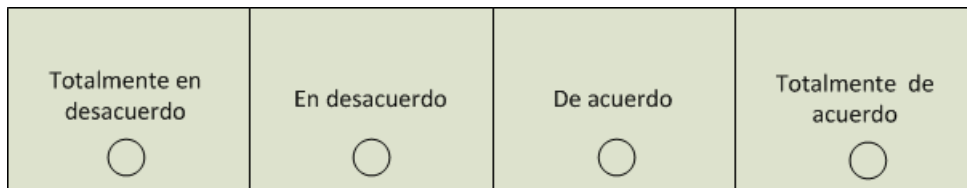


Figura 5.3: Ejemplo de una adaptación a la escala de Likert.

5.1.3. Escala de Usabilidad del Sistema SUS

La escala de Usabilidad del Sistema (System Usability Scale- SUS) es una herramienta confiable y rápida para medir la usabilidad de un sistema o producto a partir de respuestas a afirmaciones tipo Likert [24].

SUS cuantifica las respuesta desde altamente en desacuerdo hasta totalmente de acuerdo (para cada usuario) tomando en cuenta la propiedad positiva y negativa de las afirmaciones, posteriormente se normaliza la cuantificación a valores de 0 a 100 para concluir, según la figura 5.4, la calificación de la usabilidad en términos de adjetivos (peor imaginable-worst imaginable, pobre-poor, medio-ok, bueno-good, excelente-excellent y mejor imaginable-best imaginable).

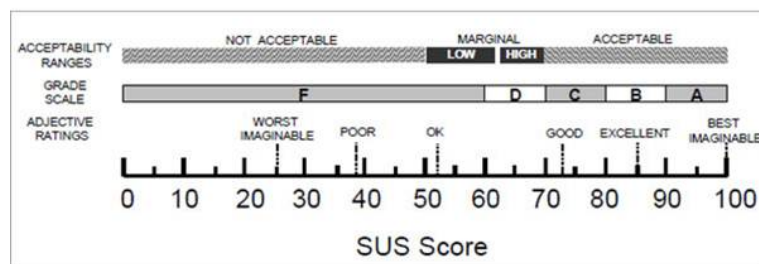


Figura 5.4: Puntuación de la Escala de Usabilidad del Sistema en términos de adjetivos.

5.2. Protocolo

Las pruebas de validación se realizarán a 10 sujetos sanos, sin amputación de miembro superior. A cada persona se le realizarán los siguientes procedimientos antes de la captura de las señales EMGs.

- Preparación de la piel con el fin de permitir una lectura adecuada de las señales EMGs.
- Ubicar a cada sujeto en una postura inicial.
- Determinar la localización de los electrodos (dos emisores y un ground).
- Fijar los electrodos sobre el musculo del sujeto.
- Testear las conexiones.

En cuanto a la limpieza de la piel se realizará una presión suave y controlada durante 3 barridos, con una lija muy fina, sobre el antebrazo y posteriormente se frotará con una almohadilla de alcohol sobre la misma zona. Así se garantiza una eliminación de células muertas y suciedad que podrían entorpecer la captura de las señales EMGs.

Posterior, se continua con la prueba del valor máximo de contracción voluntaria (MCV), para el cual se debe:

- Realizar una secuencia de calentamiento inicial que comprenda estiramiento y ejercicios aerobicos bajos de 5 a 10 minutos.
- Realizar el primer ejercicio, empezar a aumentar lentamente la fuerza, alcanzando el esfuerzo máximo después de 3 a 5 segundos, sostener durante 3 segundos y calmar con 3 segundos.
- Descansar 40 segundos y repetir el paso anterior.
- Descansar 1 minuto.
- Realizar el segundo ejercicio, empezar a aumentar lentamente la fuerza, alcanzando el esfuerzo máximo después de 3 a 5 segundos, sostener durante 3 segundos y calmar con 3 segundos.
- Descansar 40 segundos y repetir el paso anterior.
- Descansar 1 minuto.
- Realizar el tercer ejercicio, empezar a aumentar lentamente la fuerza, alcanzando el esfuerzo máximo después de 3 a 5 segundos, sostener durante 3 segundos y calmar con 3 segundos.
- Descansar 40 segundos y repetir el paso anterior.
- Descansar 1 minuto.

Una vez se haya obtenido el valor de MVC para cada usuarios, se halla la mitad del valor de del rango de variación de la señal EMG y se procede a ejecutar los siguientes procedimientos para realizar el agarre de un objeto con la prótesis virtual de mano a partir de señales EMGs.

- Reubicar al sujeto en una nueva postura
- Realizar una contracción de los músculos del antebrazo y sostener durante 3 segundos para realizar el agarre de un objeto virtual.
- Relajar los músculos inmediatamente.
- Oprimir un switch virtual para abrir la mano virtual.

El anterior procedimiento se repite para el número de objetos a agarrar esperando dos minutos entre cada prueba.

5.3. Encuesta de evaluación

Se realizará una encuesta para averiguar que tan fácil le resulta a un sujeto realizar un agarre a objetos con la prótesis virtual diseñada, a partir de señales electromiográficas de su propio brazo. Lo anterior con el fin de averiguar si el diseño mecánico del dedo facilita el control del agarre de objetos mediante señales EMGs.

La encuesta se realizará en línea a través de la herramienta de Google Forms, que permite crear y analizar encuestas de forma gratuita, permite responder desde cualquier dispositivo con conexión a internet y, además, las respuestas a las encuestas se recopilan de forma automática y ordenada en Formularios, con gráficos y datos en más profundidad exportando los datos a hojas de cálculo.

Las preguntas de la encuesta realizadas para este proyecto de grado se basaron en las investigaciones de [21] y [9]. En [21] se desarrolla y valida nuevas escalas de medición para predecir la aceptación de la tecnología por parte del usuario para dos variables específicas, la utilidad percibida y la facilidad de uso percibida, que se hipotetizan como determinantes fundamentales de la aceptación del usuario. Las definiciones de estas dos variables se utilizaron para desarrollar elementos de escala que se probaron previamente para la validez del contenido, y luego, se probaron para la confiabilidad y la validez de construcción en dos estudios con un total de 152 usuarios y cuatro programas de aplicación. Por otro lado en [9] se realizan protocolos de pruebas de validación y encuestas para evaluar la utilidad potencial, la usabilidad y la facilidad de uso percibida de un software, por parte de una muestra de usuarios de 14 personas de la Pontificia Universidad Javeriana.

5.3.1. Encuesta diseñada

ENCUESTA DE VALIDACIÓN

Este cuestionario busca evaluar el grado de complejidad, percibido por un grupo de usuarios sin amputación de miembro superior, al realizar agarres de objetos con una prótesis virtual de mano a partir de las contracciones musculares captadas desde sensores de electromiografía superficial (EMGs). Para ello, califique cada una de las siguientes afirmaciones teniendo en cuenta la experiencia vivida con la prótesis

virtual: 1= Totalmente en desacuerdo, 2= En desacuerdo, 3= De acuerdo, 4= Totalmente de acuerdo.

Nombre:

Edad:

1. A menudo me confundo cuando uso los electrodos EMGs para realizar agarres de objetos con la prótesis.

1. 2. 3. 4.

2. Yo cometo errores frecuentemente cuando realizo el agarre de objetos con la prótesis virtual a través de electrodos EMGs.

1. 2. 3. 4.

3. Interactuar con los electrodos EMGs para realizar agarres de objetos con la prótesis virtual es frecuentemente frustrante.

1. 2. 3. 4.

4. Interactuar con los electrodos EMGs para realizar agarres de objetos con la prótesis virtual requiere mucho de mi esfuerzo mental.

1. 2. 3. 4.

5. Encuentro fácil lograr que la prótesis haga lo que quiero hacer por medio de los electrodos EMGs.

1. 2. 3. 4.

6. La prótesis de mano frecuentemente se comporta de forma inesperada

1. 2. 3. 4.

7. Me resulta incomodo realizar agarres de objetos con la prótesis virtual a partir de los electrodos EMGs.

1. 2. 3. 4.

8. Es fácil para mi recordar como desempeñar el agarre de objetos con la prótesis de mano virtual.

1. 2. 3. 4.

9. Es fácil para mi aprender a desempeñar el agarre de objetos con la prótesis virtual de mano.

1. 2. 3. 4.

10. Es fácil para mi entender mi interacción con la prótesis virtual de mano.

1. 2. 3. 4.

5.4. Resultado

5.4.1. Resultados de la encuesta

La encuesta de validación se aplica a 10 usuarios, su rango de edad varia desde los 19 años hasta los 59 años. Entre los usuarios, se cuenta con estudiantes, amas de casa y profesores.

Antes de presentar el resultado de la encuesta, es importante mencionar que para los dos primeros usuarios de la prueba se debió de realizar bastantes pruebas iniciales para verificar que el hardware de adquisición estuviera leyendo valores correctos de EMGs, esto debido a que el movimiento de los electrodos sobre la placa E-Health presentaba falsos picos de fuerza, e incluso variaciones de fuerza cuando el usuario relajaba su mano, lo cuál debería resultar en un valor constante entre 0 a 1024 y no a una significativa variación.

A continuación se presenta el gráfico de torta para los resultados de cada una de las preguntas contenidas en la encuesta de validación.

A menudo me confundo cuando uso los electrodos EMGs para realizar agarres de objetos con la prótesis

10 respuestas

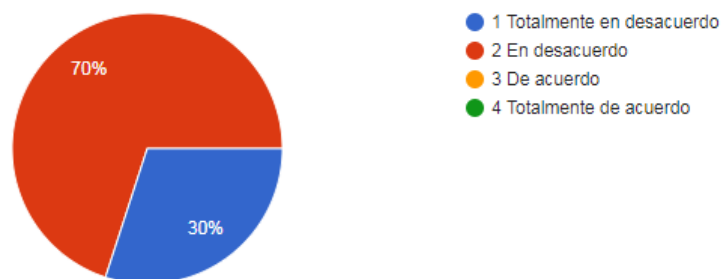


Figura 5.5: Respuesta a pregunta 1 de la encuesta de validación.

Yo cometo errores frecuentemente cuando realizo el agarre de objetos con la prótesis virtual a través de electrodos EMGs

10 respuestas

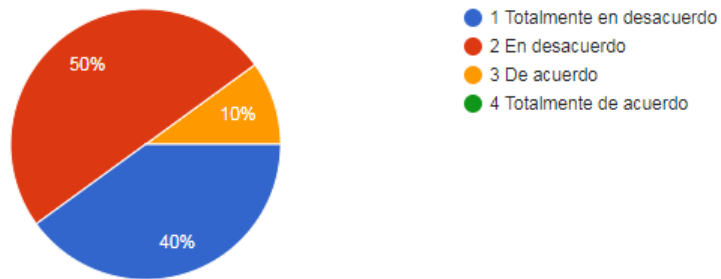


Figura 5.6: Respuesta a pregunta 2 de la encuesta de validación.

Interactuar con los electrodos EMGs para realizar agarres de objetos con la prótesis virtual es frecuentemente frustrante

10 respuestas

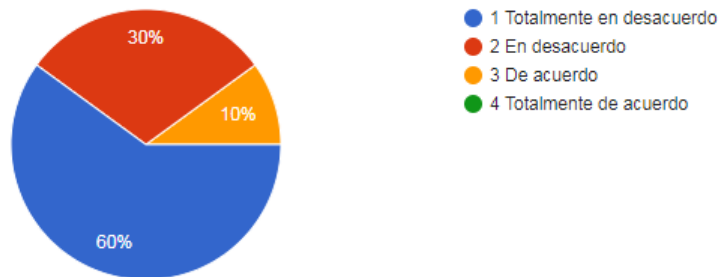


Figura 5.7: Respuesta a pregunta 3 de la encuesta de validación.

Interactuar con los electrodos EMGs para realizar agarres de objetos con la prótesis virtual requiere mucho de mi esfuerzo mental

10 respuestas

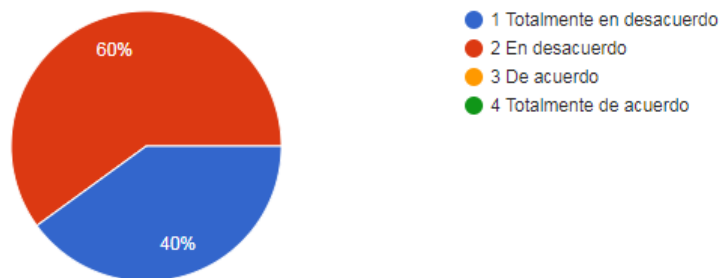


Figura 5.8: Respuesta a pregunta 4 de la encuesta de validación.

Encuentro fácil lograr que la prótesis haga lo que quiero hacer por medio de los electrodos EMGs

10 respuestas

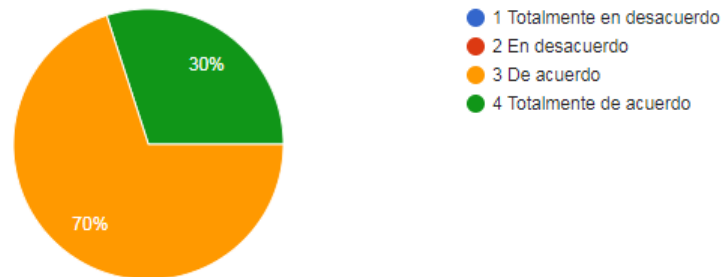


Figura 5.9: Respuesta a pregunta 5 de la encuesta de validación.

La prótesis de mano frecuentemente se comporta de forma inesperada

10 respuestas

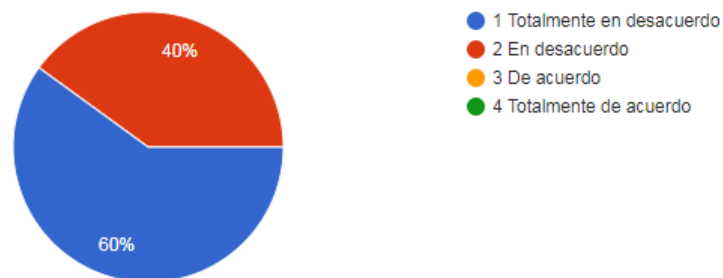


Figura 5.10: Respuesta a pregunta 6 de la encuesta de validación.

Me resulta incomodo realizar agarres de objetos con la prótesis virtual a partir de los electrodos EMGs.

10 respuestas

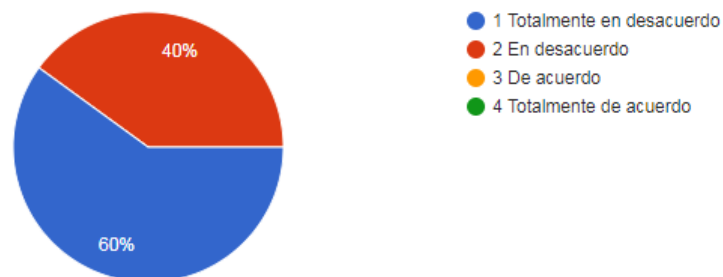


Figura 5.11: Respuesta a pregunta 7 de la encuesta de validación.

Es fácil para mi recordar como desempeñar el agarre de objetos con la prótesis de mano virtual

10 respuestas

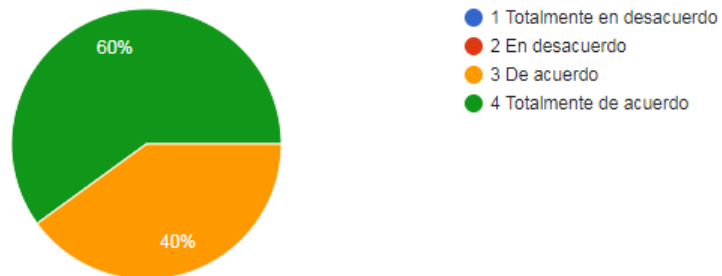


Figura 5.12: Respuesta a pregunta 8 de la encuesta de validación.

Es fácil para mi aprender a desempeñar el agarre de objetos con la prótesis virtual de mano

10 respuestas

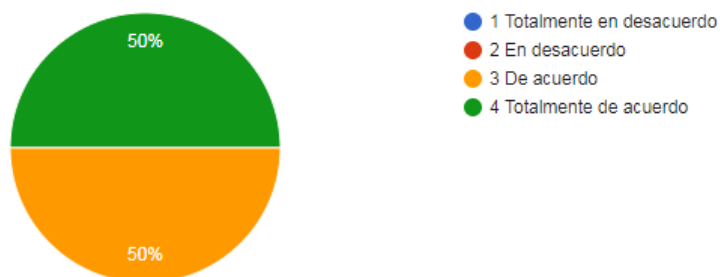


Figura 5.13: Respuesta a pregunta 9 de la encuesta de validación.

Es fácil para mi entender mi interacción con la prótesis virtual de mano

10 respuestas

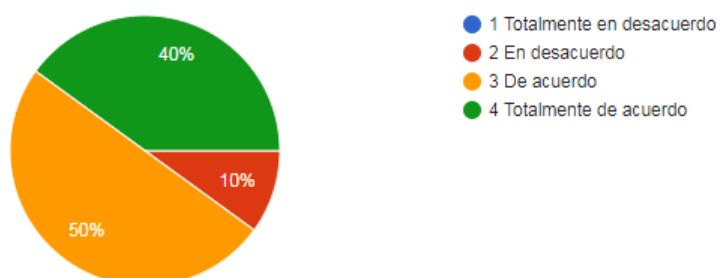


Figura 5.14: Respuesta a pregunta 10 de la encuesta de validación.

5.4.2. Resultados de la Escala de Usabilidad del Sistema - SUS

Una vez obtenidas las respuestas tipo likert a las 10 preguntas de la encuesta se procede a hallar la escala de usabilidad del sistema para los 10 usuarios individualmente.

En la figura 5.15 se observa la escala de usabilidad del sistema para la prótesis de mano virtual analizada para 10 usuarios sin amputación de miembro superior.

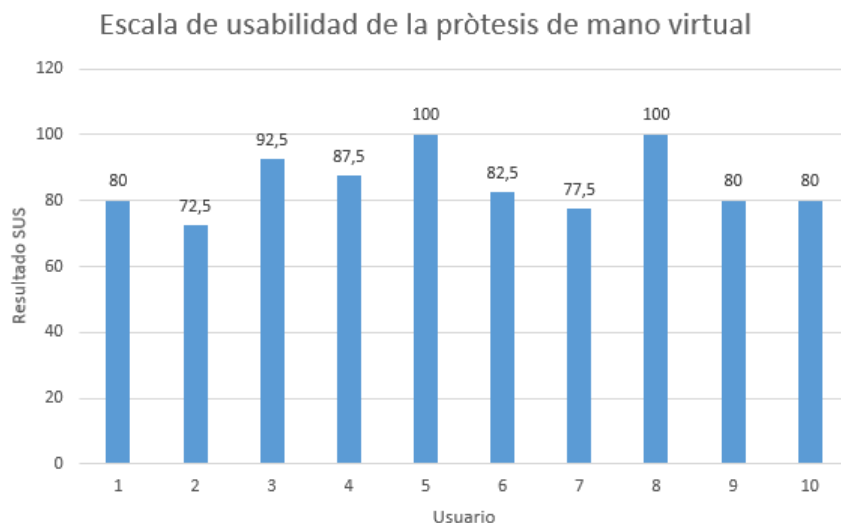


Figura 5.15: Escala de usabilidad de la prótesis de mano virtual.

Al realizar el promedio de la escala de usabilidad de la prótesis para los 10 usuarios, se obtiene 85,25, lo cuál significa que la usabilidad percibida por los usuarios, respecto a la prótesis de mano virtual gobernada por EMGs, es excelente.

Finalmente al haber evaluado la usabilidad y sus atributos (facilidad de aprendizaje, eficiencia, recuerdo en el tiempo, errores y satisfacció) se concluye que, el uso de la prótesis de mano virtual gobernada mediante EMGs se realiza con facilidad.

5.5. Conclusiones

El 100 % de los usuarios percibe que es fácil aprender a desempeñar agarres de objetos con la prótesis virtual de mano diseñada a partir de señales EMGs, sin requerir mucho de su esfuerzo mental para lograrlo.

El 100 % de los usuarios no perciben como una tarea compleja el agarre de objetos virtuales a partir de EMGs.

El 10% de los usuarios consideran que cometen errores frecuentemente cuando realizan el agarre de objetos con la prótesis virtual de mano, y que es frecuentemente frustrante realizar el agarre, esto es debido principalmente a la falsa lectura de los electrodos causada por el movimiento de los cables de conexión a la placa.

Para el 100 % de los usuarios es fácil lograr que la prótesis haga lo que se quiere hacer por medio de EMGs.

El uso de la prótesis de mano virtual gobernada mediante EMGs, se realiza con facilidad para un grupo de usuarios sin amputación de miembro superior.

Para 5 de los usuarios, la usabilidad de la prótesis de mano gobernada por señales EMGs es excelente.

Para 4 de los usuarios, la usabilidad de la prótesis de mano gobernada por señales EMGs es la mejor imaginable.

Para 1 de los usuarios, la usabilidad de la prótesis de mano gobernada por señales EMGs es buena.

Trabajos Futuros

A partir de los resultados obtenidos y de la experiencia adquirida a lo largo de los diferentes intentos de control de la prótesis de mano, se proponen algunas trabajos futuros que podrían contribuir a realizar una simulación más eficiente y con menos carga computacional que permita una verdadera simulación en tiempo real.

- Usar otro software de simulación dinámica que permita una conexión con Matlab en tiempo real pese a la agregación de propiedades físicas.
- Realizar la prueba de facilidad de uso a una muestra de personas sin amputación de miembro superior.
- Realizar el control de flexión-extensión de los dedos de la a prótesis de mano bajo los efectos de la gravedad.
- Realizar el control de flexión-extensión en el agarre de los dedos de la a prótesis de mano en el agarre de objetos cuando los mecanismos de auto adaptabilidad están activados.
- Retomar el control VSD para controlar la posición de la yema de los dedos de la ultima versión de diseño mecánico de la prótesis de mano, con un esquema adaptativo que permita compensar el efecto de la gravedad.
- Implementar una estructura física estable que permita la adquisición de señales EMGs sin que el movimiento de los cables de los electrodos produzcan falsos picos de fuerza.

Apéndice A

Procedimientos en SolidWorks, Adams y Matlab

A.1. Importación de un modelo CAD de SolidWorks en Adams

El diseño mecánico de las piezas y el ensamblaje de la prótesis de mano fue desarrollado en el software SolidWorks 2016. Posteriormente fue necesaria su exportación a Adams con el fin de generar la simulación de la planta (prótesis de mano) y posteriormente realizar una ley de control que permita a la prótesis realizar agarres de objetos. Se trabajó con el modulo Adams/View, una interfaz que permite la creación de modelos multi-cuerpo o la importación de modelos CAD ya hechos. Para llevar a cabo esta exportación desde SolidWorks, se ejecuta en Adams 2016 con licencia craqueada, ya que en su versión estudiantil existe una restricción del número de piezas a utilizar. Sin embargo, la versión estudiantil se puede adquirir gratuitamente en la página web de Msc.Adams después de pasar por un proceso de registro, se descarga un archivo ejecutable para Windows, con arquitectura de 64 bits el cual tiene una validez por un año y ciertas restricciones. Una vez instalado Adams, se procede a abrir su interfaz la cual despliega una ventana, la cual muestra las opciones de inicio como se muestra en la figura A.1.

Una vez ubicados en la ventana de la figura A.1, lo siguiente es seleccionar la opción **New Model**, con lo que se habilitan nuevas opciones para la creación del modelo mecánico, tales como las unidades con las cuales se va a trabajar, la dirección del directorio de trabajo, el valor y sentido de la gravedad. Ver Figura A.2.

Efectuada esta tarea, lo siguiente es la creación del modelo mecánico, como bien se había indicado, el modelo CAD ya se encuentra hecho así que se importara la geometría existente desde el menú File-Import de la barra de herramientas del programa. El archivo a importar en Adams ha sido previamente exportado desde el archivo de ensamblaje de SolidWorks. Algunos de los formatos que permite exportar SolidWorks son PARASOLID (*.x_t), IGES (.iges), STEPAP214 (*.step, *.stp) y STEPAP203 (*.step, *.stp). Por otro lado, en Adams, se ofrecen distintas alternativas para la importación de archivos, entre los que se encuentran: PARASOLID, STEP, IGES, DFX, STEREOGRAPHY entre muchos otros (Ver Figura A.3).

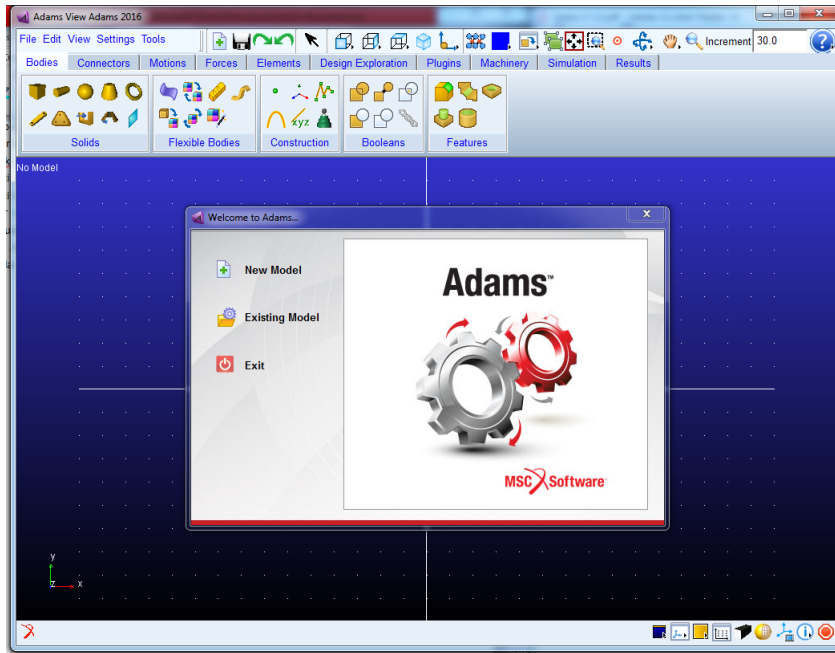


Figura A.1: Inicio del software Adams.

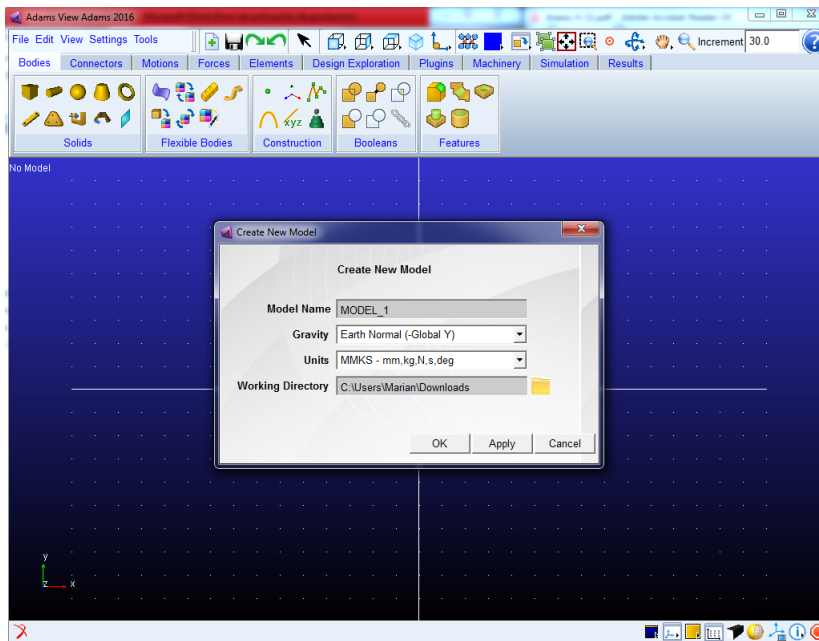


Figura A.2: Creación de un nuevo modelo en Adams.

Como comentario adicional, al generar el archivo en formato IGES, STEP214 y STEP204 se presentaron problemas para poder asignar en Adams características másicas a algunas piezas. Es importante recalcar que si al menos una pieza en Adams no puede poseer las características másicas no se puede realizar una simulación. En este proyecto, el único archivo que permitió la asignación de características másicas a todas las piezas y por tanto una correcta simulación fue PARASOLID (*.x_t).

A.1. IMPORTACION DE UN MODELO CAD DE SOLIDWORKS EN ADAMS 107

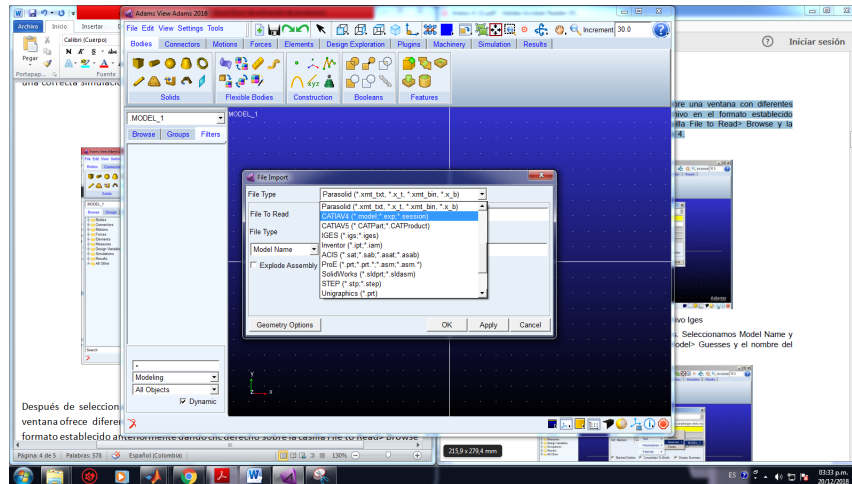


Figura A.3: Importación del modelo CAD.

Después de seleccionar en **File Type** el formato *Parasolid* (ver figura A.3), la misma ventana ofrece diferentes campos a completar, se debe buscar el nombre del archivo en el formato establecido anteriormente dando clic derecho sobre la casilla **File to Read** > **Browse** y la carpeta donde se encuentra el archivo *Parasolid* (Ver Figura A.4)).

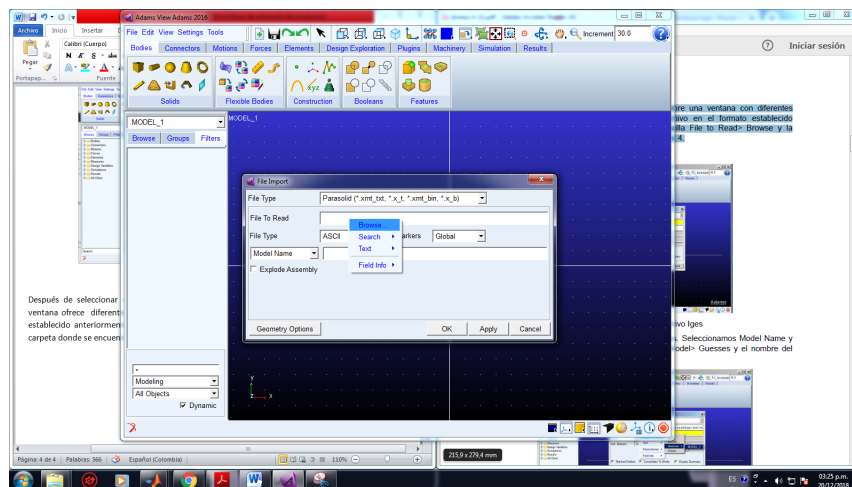


Figura A.4: Elegir el archivo Parasolid exportado desde SolidWorks.

Luego de esto, se completan los campos faltantes. En frente de la casilla Model Name hacemos clic derecho, elegimos *Model* > *Guesses* y el nombre del modelo creado en ADAMS (**MODEL_1**, para este caso). Ver Figura A.5.

Después se define la locación y la orientación que se desea tener en el modelo, se selecciona **Apply** y **OK**. Inmediatamente aparecerá, para este caso, la prótesis virtual de mano importada en Adams. Ver figura A.6.

En la figura A.6 se observa la prótesis de mano en posición cerrada, y será durante toda la simulación la posición inicial. Esta posición inicial depende de la última posición en la que se haya establecido el ensamblaje en SolidWorks antes de su

debida exportación.

Adicionalmente, la prótesis de la figura A.6 no está en una vista renderizada, sin embargo es posible configurar otro tipo de vista. Con el fin de apreciarla con más claridad y realismo, se selecciona con un click el icono redondo-amarillo ubicado en la parte inferior derecha de la interfaz del programa. Finalmente se obtiene la vista de la figura A.7.

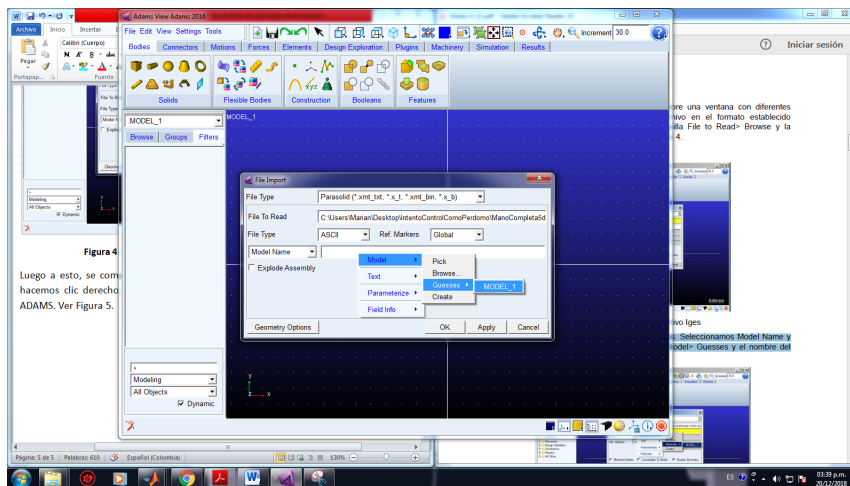


Figura A.5: Elegir el archivo Parasolid exportado desde SolidWorks.

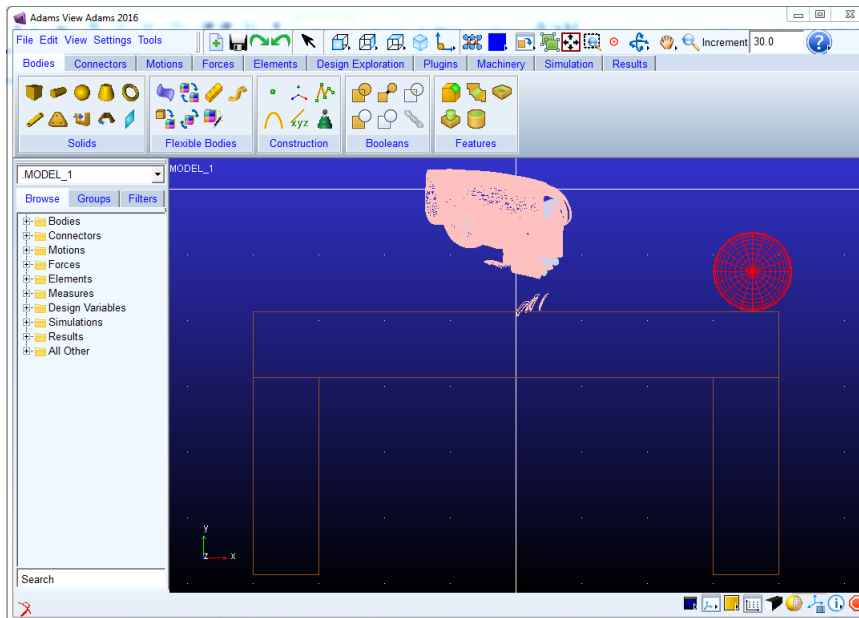


Figura A.6: Diseño CAD exportado en Adams en vista no renderizada.

En este punto, si se deseara realizar una simulación, no sería posible, ya que se debe seguir una serie de pasos previos. Primero se debe proporcionar propiedades másicas a cada una de las piezas que conforman la prótesis de mano. Para ello se realiza clic en la parte izquierda de la interfaz de Adams en la carpeta **Bodies**, se selecciona una de las piezas del largo listado, se realiza clic derecho y se selecciona

Modify, se despliega una ventana, se debe dirigir a *Category > MassProperties*; en **Define Mass By** seleccionar **Geometry and Material Type** y en **Material Type** dar clic derecho en *Material > Guesses* y escoger de la lista de opciones el material de preferencia. Ver figura A.8.

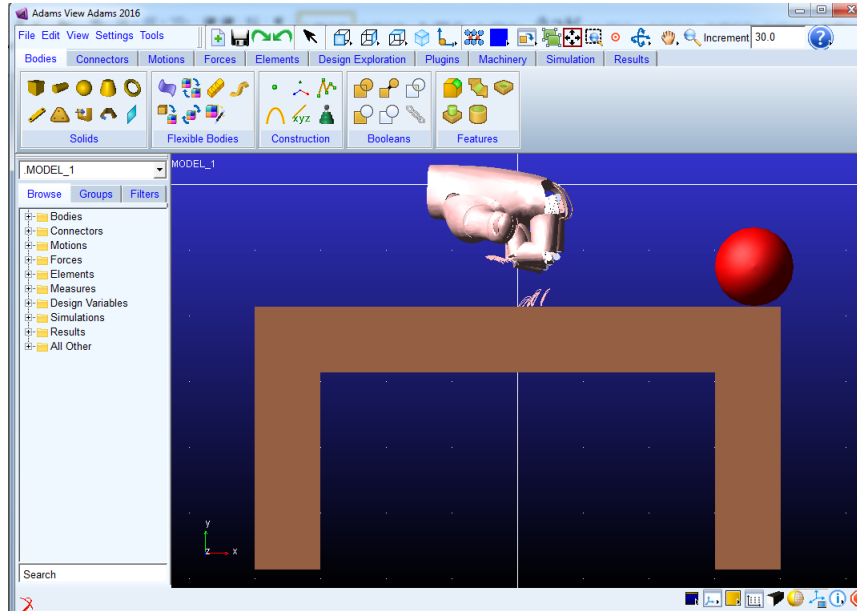


Figura A.7: Diseño CAD exportado en Adams en vista renderizada.

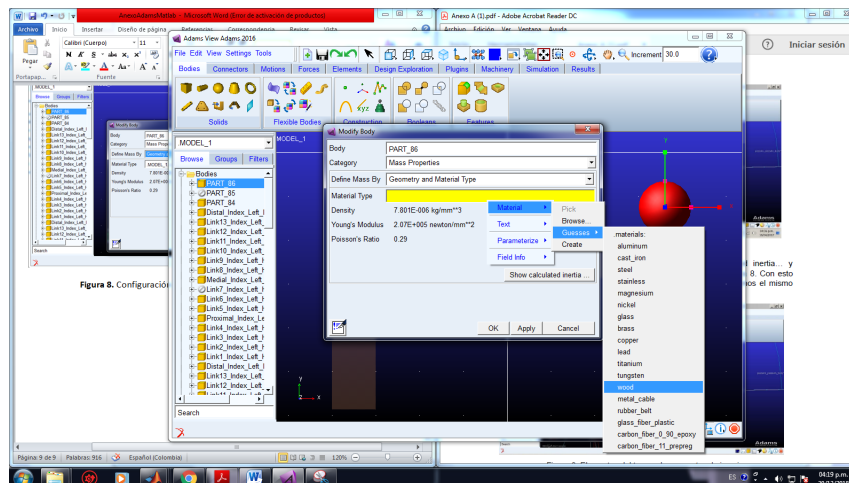


Figura A.8: Configuración de las propiedades másicas para una pieza.

Posteriormente se selecciona en **Apply** y en **Show Calculated inertia**. De esa manera se obtienen los elementos del tensor de momentos de inercia y la definición de la masa para la pieza en particular. Posterior, se debe repetir el mismo proceso para las demás piezas que componen el archivo de Adams.

Lo siguiente por definir serán las juntas de revolución que, si bien en SolidWorks se encontraban definidas por relaciones de posición, estas características no se tras-

pasan a Adams y hay que definir las de nuevo. Para ello, se debe dirigir a la barra de herramientas a la opción **Connectors**, en la cuál se visualizan una serie de opciones entre las cuales se encuentran las denominadas **Joints** o tipos de juntas, tales como **Fixed joint** correspondiente a una unión fija las cuales se pueden aplicar sobre dos cuerpos distintos en un único punto de aplicación y **revolute joint** la cual permite definir entre dos cuerpos que existe una junta de revolución, la cual permite rotación entre estos. Ver figura A.9, donde se encuentran seleccionadas en rojo todas las juntas que ofrece Adams. Para definir una junta se debe elegir el tipo de junta y las dos piezas que intervienen en dicha unión.

Posterior se deben realizar todas las juntas que garanticen el movimiento deseado del mecanismo.

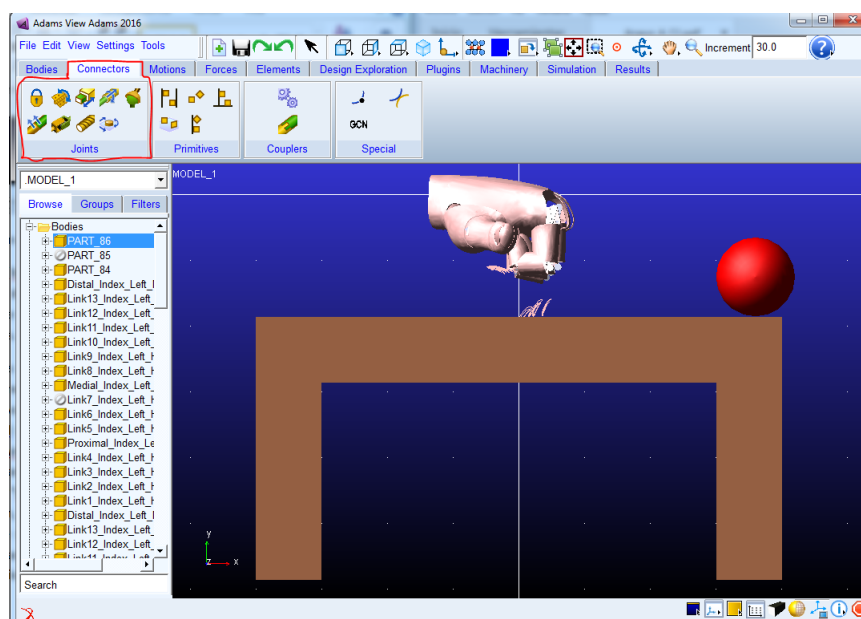


Figura A.9: Tipos de juntas en Adams.

En este proyecto de Adams, existen 5 motores rotacionales, es decir que se tienen 5 juntas actuadas (una por cada dedo). Cada motor es simulado con un torque rotacional, por tanto para este proyecto es necesario crear 5 torques rotacionales. Para definir un torque rotacional se debe dirigir a barra de herramientas y seleccionar la pestaña **Forces**, se dirige a la subpestaña **Applied Forces** y selecciona el icono verde seleccionado por un círculo rojo (Ver figura A.10). Para crear el torque se pide seleccionar un cuerpo y un punto de rotación. Por defecto Adams colocara un valor constante de rotación; Sin embargo, si se desea manipular externamente el movimiento, como por ejemplo desde Simulink, se debe dirigir a la carpeta de la parte izquierda de la interfaz de Adams, llamada **Forces** y elegir de la lista plegable el nombre de la fuerza creada, dar clic alterno y elegir **Modify**, dirigirse a **Function** y escribir **VARVAL(x)**, donde x será una variable de estado previamente creada, que permitirá la comunicación entre Adams y Simulink. Ver figura A.11.

Para este proyecto es necesario medir los 5 angulos de las articulaciones actuadas, con el fin de realimentar el sistema de control y cerrar el lazo. Estas mediciones se deben configurar mediante la colocación de un sensor en las uniones de los cuerpos

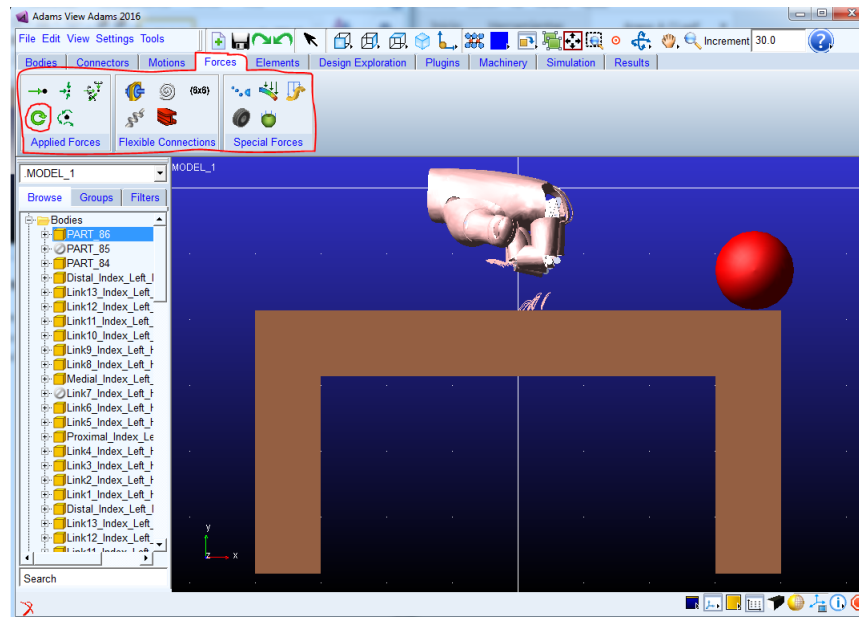


Figura A.10: Sección de **Forces** para aplicar un Torque rotacional.

conectados por las juntas de movimiento. Adams permite colocar estos sensores en las uniones definidas entre sólidos, permitiendo la medida de diferentes variables como fuerzas, pares, desplazamientos, velocidades lineales relativas, velocidades angulares relativas, aceleraciones lineales relativas y aceleraciones angulares relativas. La configuración de la medición de ángulo resulta sencilla. Se debe dirigir a carpeta de la parte izquierda de la interfaz de Adams, llamada **Connectors**, posicionarse sobre la junta actuada, dar clic alterno y elegir **Modify**, con lo cual aparecerá la pestaña **Modify Joint**. En esta pestaña se debe elegir el icono de realizar medidas, subrayado en rojo en la figura A.12. Con lo cual aparecerá una nueva pestaña llamada **Joint Measure** en la que se debe primero dirigir a **Measure Name** y asignarle un nombre a la medida de ángulo, después se debe desplegar la lista de **Characteristic** y elegir la opción **Displacement** (Ver figura A.13), aplicar **ok** y cerrar las pestañas.

Adicionalmente, sobre la pestaña **Joint Measure** se puede observar sobre qué marcadores (puntos de referencia sobre las piezas) se está realizando la medida de ángulo. Para ello seleccionar el botón de **Orientation** y elegir sobre la nueva ventana emergente, llamada **Orientation Measure**, los marcadores sobre los cuales se medirá el ángulo (Ver figura A.14). También se pueden mantener los marcadores por defecto ubicados en las casillas **To marker** y **From Marker**. Para modificar los marcadores por defecto, hacer clic alterno sobre la casilla de **To marker** y/o **From Marker**, elegir **Marker > Browse** y se abrirá una tercera ventana (ver figura A.15) en donde nos parecerá una base de datos (ver figura A.16) en la que se encontrarán los marcadores de todo el mecanismo. Se repite el procedimiento para los otros dos marcadores con el fin de formar el ángulo de la junta a medir.

En este punto ya podemos realizar una simulación para verificar que la prótesis se mueve correctamente y que los sensores de los ángulos funcionan, para esto se debe dirigir a la opción **Simulation**, y elegir el icono subrayado en rojo de la figura A.17, con lo cual se despliega una ventana llamada **Simulation Control** en la cual

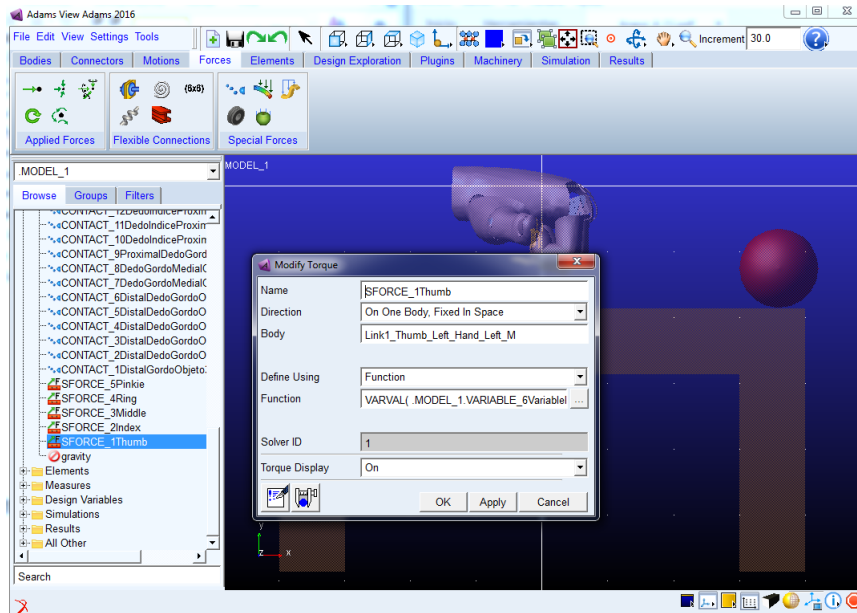


Figura A.11: Pestaña de modificación de un torque.

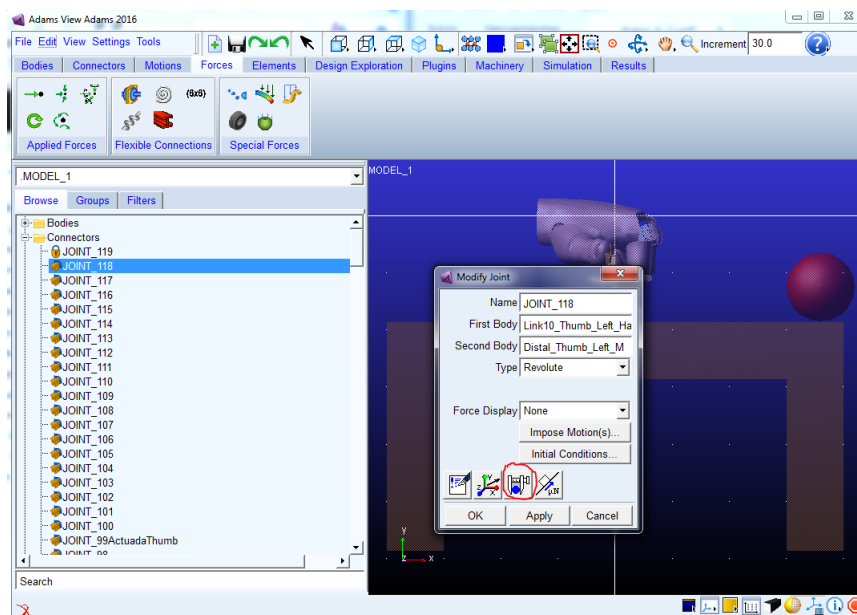


Figura A.12: Selección del icono de medidas en la Pestaña **Modify Joint**.

se asigna el tiempo de simulación y los pasos o tamaños de pasos que se desee para el proyecto.

Verificado que el movimiento es el deseado y que los sensores miden los ángulos de cada articulación, se procede a crear todas las variables de estado, las cuales son todas las entradas y salidas del sistema una vez se exporte a Simulink, para esto en la opción **Elements** seleccionamos el icono **x** encerrado en un círculo rojo en la figura A.18. Al colocar el puntero del ratón sobre este icono se verá la siguiente descripción del icono: **create a state variable defined by an algebraic equation**.

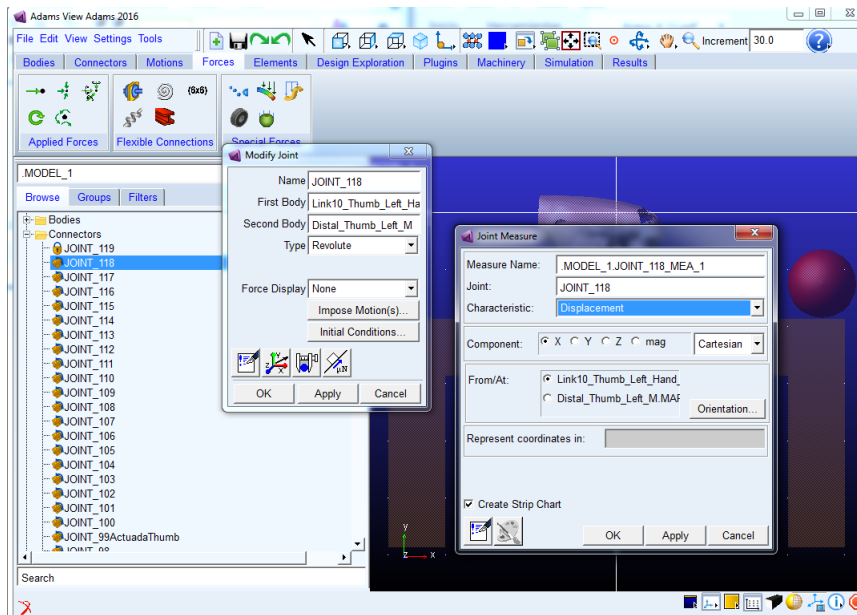


Figura A.13: Pestaña **Joint Measure** para crear medidas de ángulo, fuerzas, velocidades, etc.

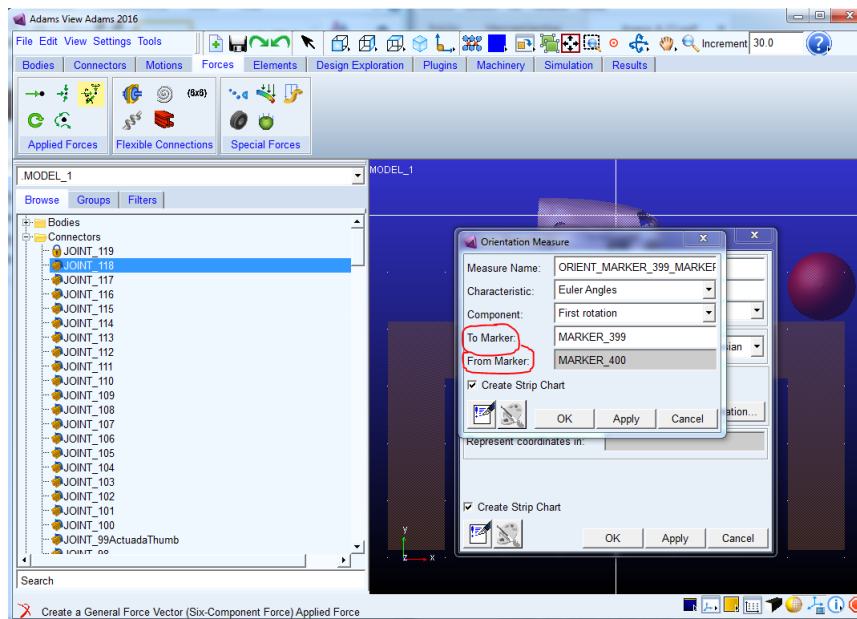


Figura A.14: Pestaña **Orientation Measure** para modificar los marcadores de medida de ángulo.

Al seleccionar el icono, se despliega una ventana, llamada **Create State Variable...**, como la de la figura A.19, en la cual debemos asignar un nombre a la variable de estado y seleccionar **OK**. Se debe crear tantas variables de estado como entradas y salidas de la plata se tengan, teniendo en cuenta que las entradas son los actuadores y las salidas, los sensores (para este proyecto serán 5 entradas y 5 salidas).

A modo de aclaración, para el sistema de control de este proyecto se necesitan 5 torques de entrada que permitan generar el movimiento deseado de la prótesis de

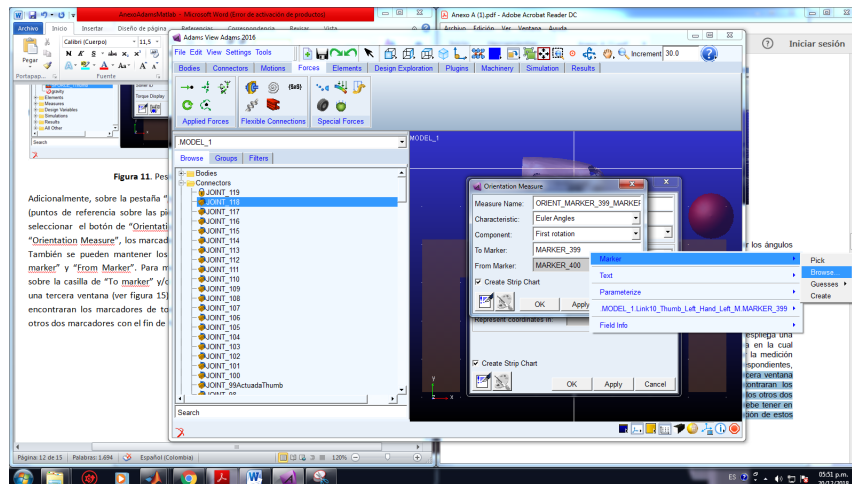


Figura A.15: Medir ángulos cambiando los marcadores por defecto.

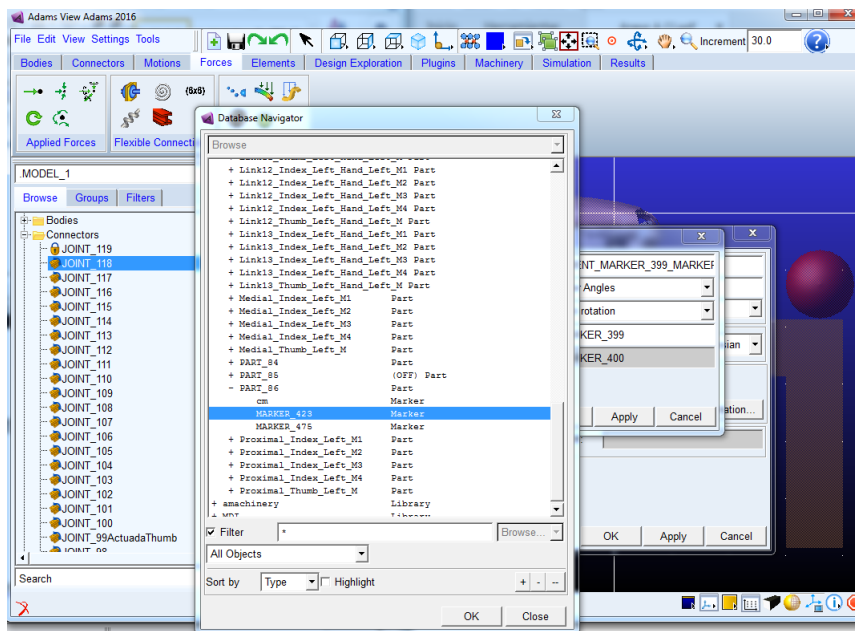


Figura A.16: Base de datos de todos los marcadores del proyecto en Adams.

mano, por lo tanto se necesitan crear 5 variables de estado, las cuales se asignan a los torques rotacionales con **VARVAL**(*Nombre_de_Variable_de_Estado*), como se explicó anteriormente. Lo anterior con el fin de que el controlador desde Simulink envíe el valor de torque necesario para seguir una trayectoria deseada. Adicionalmente, se deben crear 5 variables más de salida, las cuales son las medidas de ángulo de la articulación actuada, las cuales serán leídas desde Simulink y permitirán cerrar el lazo de control.

Hasta este punto solo las variables de estado de entrada de la planta (Valor de Torque) han sido asignadas como entrada en el torque rotacional creado, usando **VARVAL**, sin embargo no se ha realizado ningún procedimiento para relacio-

A.1. IMPORTACION DE UN MODELO CAD DE SOLIDWORKS EN ADAMS 115

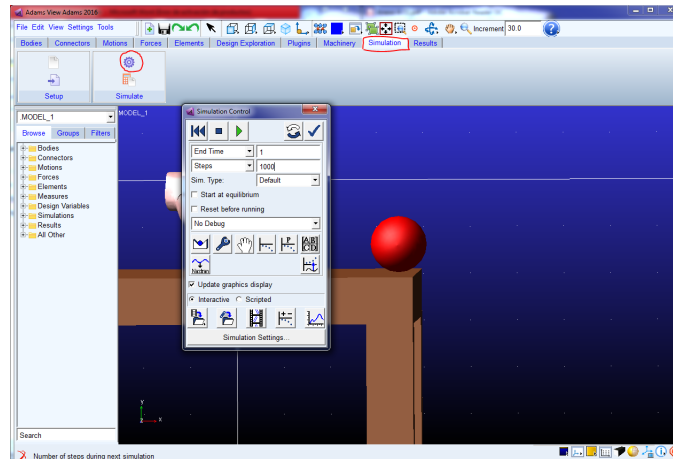


Figura A.17: Pestaña de **Simulation Control** para iniciar la simulación.

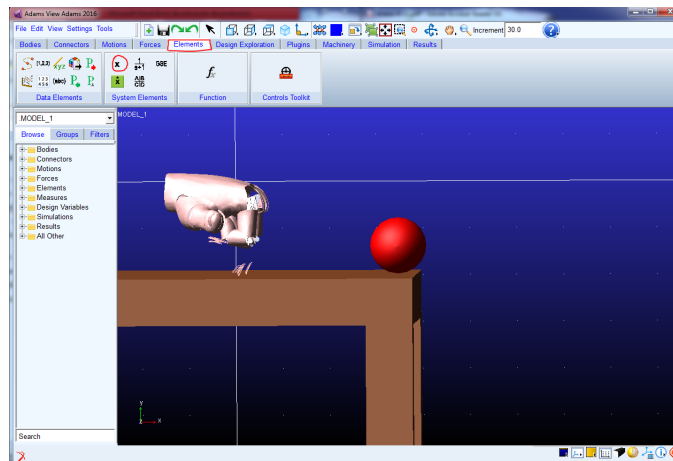


Figura A.18: Menú **Element** para crear una variable de estado.

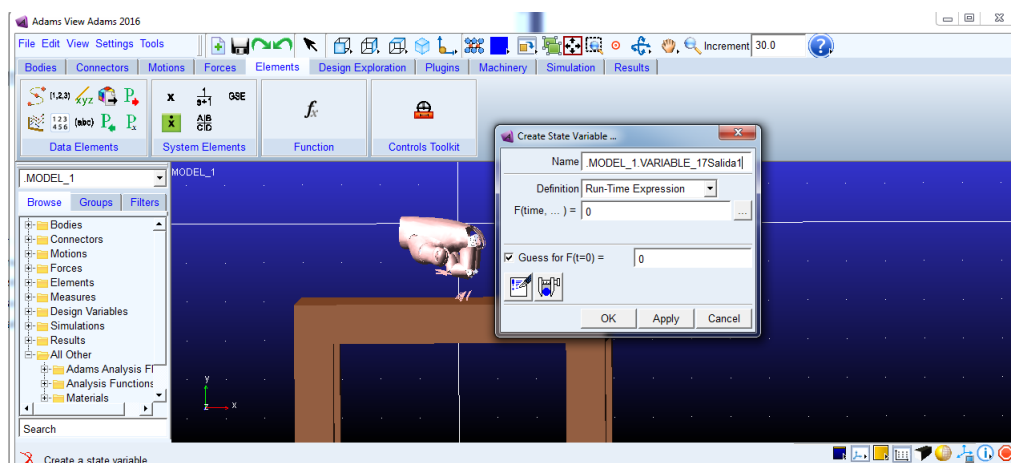


Figura A.19: Creando una variable de estado.

nar las medidas de ángulo creadas con el nombre de las variables de estado que representan las 5 variables de salida (ángulo de las 5 articulaciones actuadas). Para ello, ahora se procede a asignar todas las variables de estado creadas, a ca-

da uno de los componentes de torque y de medición, para esto se debe dirigir a *Tools > DataBaseNavigator*, se despliega una ventana, una vez ahí se busca el nombre de la variable de estado en específico, para este ejemplo **10SalidaAnguloMedio** que corresponde a la medición del ángulo de la articulación actuada del dedo medio, realizar doble clic y aparece una ventana de información, buscar la pestaña **Modify**, seleccionarla y se despliega una ventana de configuración para variables de estado, pero esta vez seleccionados en la opción **F(time,...)** y pulsar en el botón de los puntos suspensivos (...).

Posterior se despliega otra ventana en la que se va a construir las funciones de medición, hacer clic en **Getting Object Data** y buscar **Measures**, en la casilla contigua a este hacer clic derecho *Runtime_Measure > Guesses* y la medición que se desea se vincula a esta, en este caso **MedidaActuadaMiddle**, aplicar cambios, dar **OK** y repetir este proceso para todas las variables de estado restantes. Ver figura A.20.

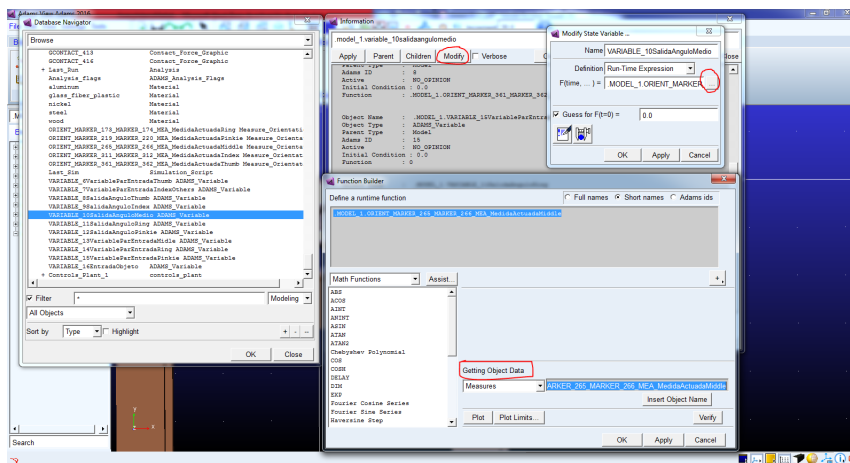


Figura A.20: Asignación de las variables de estado a los componentes de medidas.

A.2. Comunicación de la planta en Adams con Matlab

En el último de los pasos para obtener el modelo dinámico, para este proyecto de simulación es la cosimulación entre Adams y Matlab, para ello se debe hacer uso del plugin de control que ofrece Adams. Dirigirse a la opción **Plugins** y seleccionar el correspondiente a **Controls**, hacer clic **Plant Export** y se despliega la ventana de la figura A.21. Ubicarse en la opción **Input Signals** y haciendo clic derecho en el espacio en blanco *ADAMS_Variable > Guesses >* y la variable de entrada, en este caso *.MODEL_1.VARIABLE_6VariableParEntradaThumb*. Así seleccionar todas las variables de estado que intervengan en la simulación de entrada y procedimiento análogo en **Outputs Signal** para las variables de salida que son todas las mediciones de ángulo realizadas.

En **Target Software** seleccionar **Matlab** y **OK**, con estos pasos Adams crea archivos de Matlab que se encontraran en el directorio de trabajo donde se guardó el proyecto de Adams, abrir Matlab y en la ventana del **Command Window** escribir el

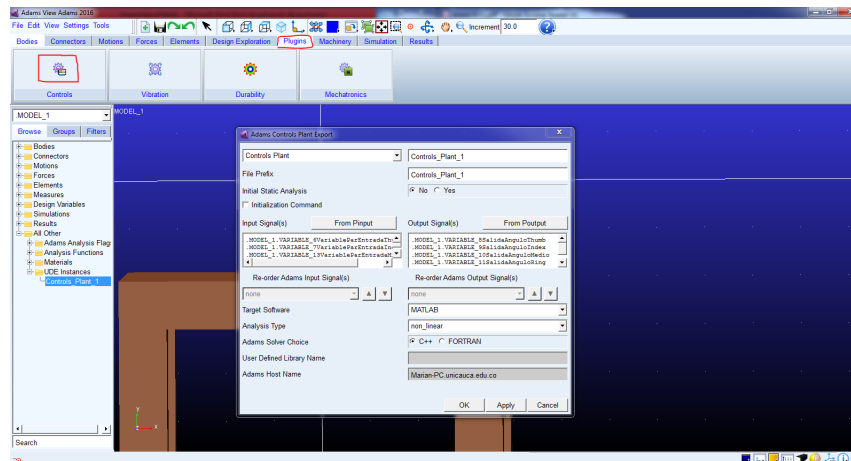


Figura A.21: Plugin Adams Controls.

nombre del archivo que genera Adams, *Controls_Plant_#* el numero depende de las veces que exportemos la planta a Matlab en nuestro caso es *Controls_Plant_1*, pulsar enter y aparecerá la descripción de las variables que participaran en la simulación. Ver figura A.22.

Posteriormente se ingresa el comando *Adams_sys* y se pulsa Enter, se genera un archivo de Simulink el cual contiene el modelo de la planta y con el cual se trabajará en el estudio de simulación. Ver figura A.23.

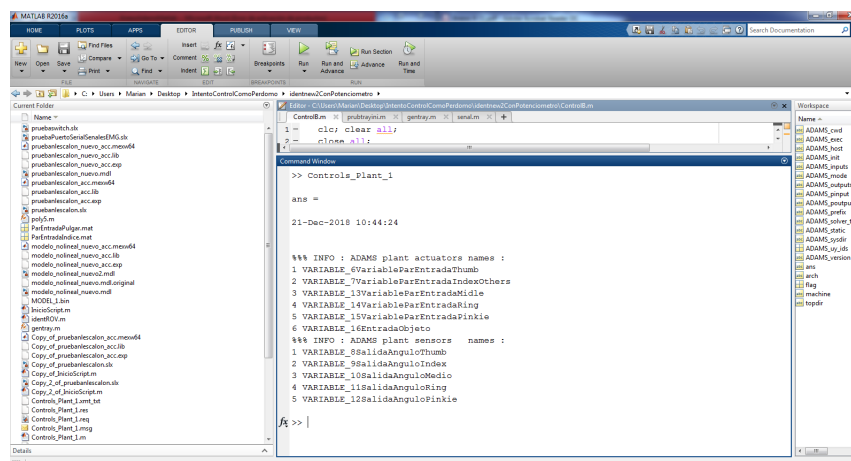


Figura A.22: Carga de variables Adams en el Workspace de Matlab.

Se procede a copiar el bloque *adams_sub* en un archivo en blanco de Simulink y se cierra *Adams_sys*. La simulación puede ser interactiva o de tipo batch, la interactiva se simula junto con Adams mostrando la simulación en cada instante de tiempo, mientras que la de tipo batch no muestra ninguna animación. Esto se puede configurar en *Adams_sub > ADAMSPlant*, doble clic, se abre un bloque de parámetros, se va a **Animation mode** y se selecciona alguna de las dos opciones dichas anteriormente. Ver figura A.24.

Siguiendo estos pasos se ha generado el modelo dinámico para la prótesis de

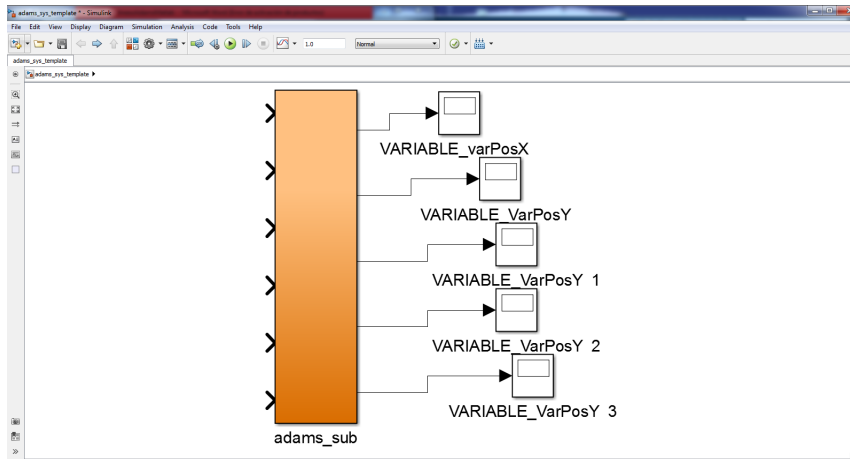


Figura A.23: Archivo de Simulink *Adams_sys*.

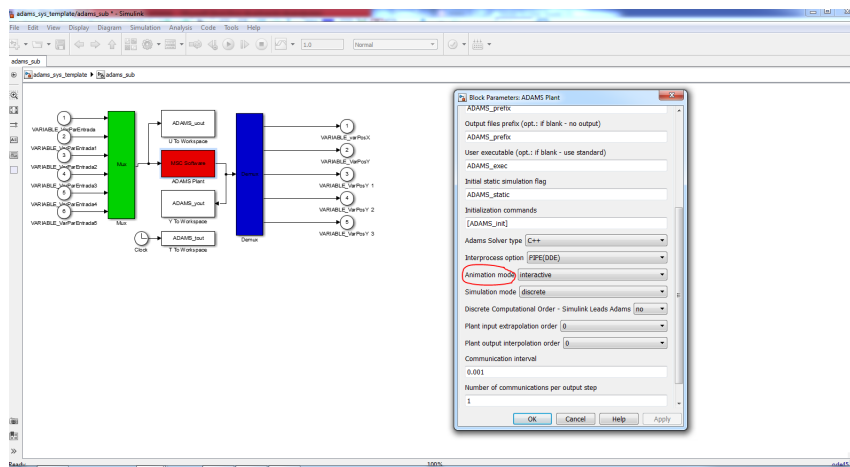


Figura A.24: Configuración de Animación ADAMS

mano, el cual se encuentra en un modelo de Simulink para ser gobernado por los sistemas de control que se diseñaron en este trabajo de grado.

Apéndice B

Procedimiento software y hardware para comunicar Arduino, E-Health y sensores EMGs

B.1. Ensamble de Arduino, E-Health y sensores EMGs

El primer procedimiento que se debe llegar a cabo es ensamblar la placa de sensores E-Health con la placa de Arduino Mega 2560. En la parte superior del ensamblaje se debe ubicar la placa E-Health y en la parte inferior la placa de Arduino (Ver figura B.1).

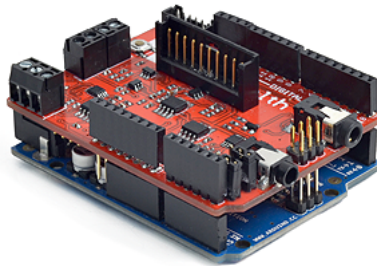


Figura B.1: Ensamblaje de placa E-Health y Arduino.

Antes de realizar las conexiones de los electrodos EMGs, se debe verificar que, en la placa E-Health, los pines 2 Y 3 ECG/EMG (se encuentran al lado de los pines analógicos) están unidos por un puente, como lo muestra la figura B.2. Al principio el puente está conectado entre los pines 1 y 2. Para poder hacer una EMG hay que mover el puente a la derecha, para que conecte los pines 2 y 3.

Una vez realizado lo anterior, se se procede utilizar los sensores EMG (Fucsia, amarillo y azul), a los que se debe conectar un electrodo adhesivo. Ver figura B.3. Posteriormente se deben conectar los tres sensores a la placa E-Health. Los tres sensores son iguales, pero dependiendo del pin donde se conecten tendrán un nombre distinto y deberán colocarse en partes diferentes del cuerpo. En este proyecto se decidió utilizar el azul como GND, el morado como MID y el amarillo como

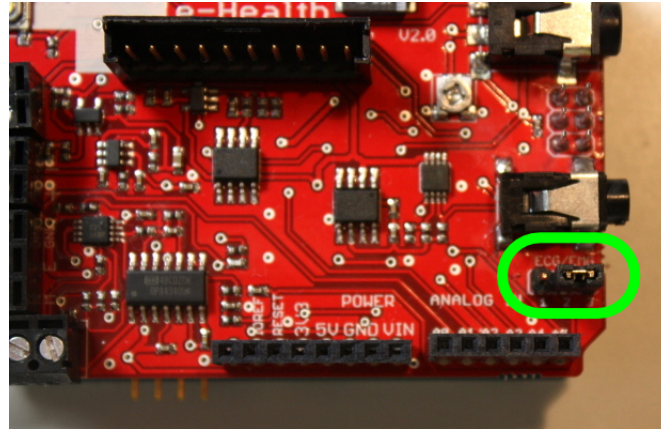


Figura B.2: Puente entre los pines ECG/EMG 2 Y 3.



Figura B.3: Sensores EMGs con electrodos adhesivos.

END. Ver figura B.4.

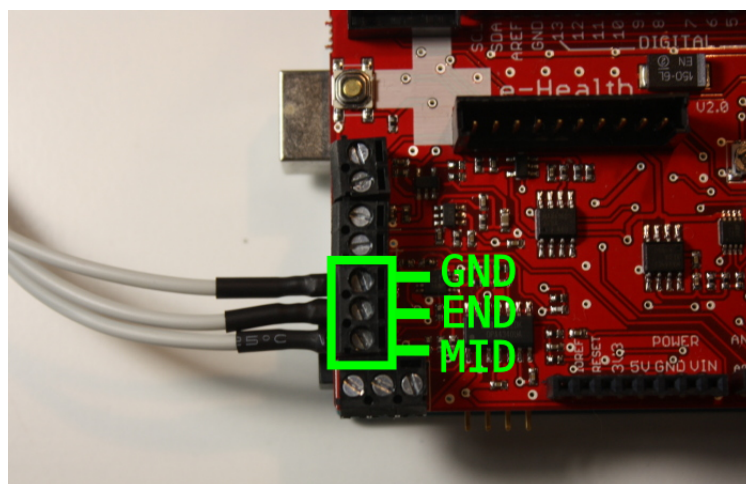


Figura B.4: Conexión de electrodos EMGs a la placa E-Health.

Hecho lo anterior se procede a separar el protector de cada electrodo (Ver figura B.5) y ubicarlos sobre la piel de los músculos a estudiar. En este proyecto se estudia

la mano, por lo tanto se conecta el GND en la palma, el MID cerca de la muñeca y el END al lado del MID, como muestra la figura B.6.

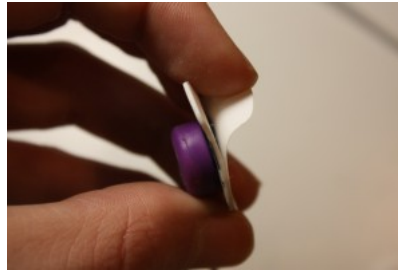


Figura B.5: Separación del protector de los electrodos.

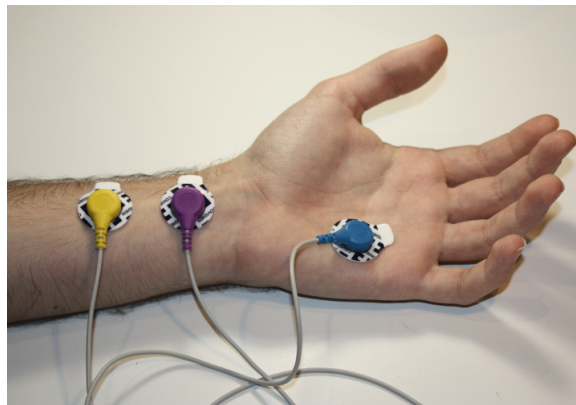


Figura B.6: Ubicación de los electrodos sobre la mano de un usuario.

Como nota adicional, cuando se necesite aumentar la sensibilidad de los electrodos, se procede a girar el potenciómetro de GAIN de la placa E-Health con un destornillador. Ver figura B.7.



Figura B.7: Potenciómetro GAIN de la placa E-Health.

B.2. Lectura de los electrodos EMGs desde Arduino

Lo primero que se debe hacer para leer los electrodos EMGs desde arduino es instalar arduino y posteriormente instalar la librería eHealth.h.

La librería **eHealth.h** es la librería oficial para operar con la placa de sensores, al descargarla se debe descomprimir el archivo .zip en el cual aparecerán dos carpetas: eHealth y PinChangeInt. Se debe mover estas dos carpetas al directorio dónde están ubicadas las librerías de Arduino. No va a funcionar si se mueve sólo los ficheros .cpp y .h que hay dentro, ni si estas dos carpetas están dentro de otra.

Se encontraron algunas incidencias al intentar utilizar esta librería con las últimas versiones de Arduino. La versión de arduino que ha funcionado sin ningún inconveniente es la versión 1.0.5.

A continuación se observará un código sencillo en arduino, el cuál permite leer los sensores EMGs. Ver figura B.8.

```

1  /* ELECTROMIOGRAFIA CON ARDUINO
2  * Muestra las lecturas de los electrodos por Serial
3  *
4  *
5  * Escrito por Francisco Beas Hurtado
6  * www.cafeyprozac.wordpress.com
7  *
8  * y Transductor
9  * www.robotlogs.net
10 * /
11
12 #include <eHealth.h>
13
14
15 void setup()
16 {
17   Serial.begin(9600);
18 }
19
20
21 void loop()
22 {
23   int val = eHealth.getEMG();
24
25   Serial.println(val);
26 }
27
28 //Final de linea

```

Figura B.8: Código en arduino para leer sensores EMGs.

El programa anterior escribe las lecturas de los electrodos por Serial. La función `eHealth.getEMG()` mide la actividad muscular que detectan los sensores y devuelve un número entero entre 0 y 1024. Lo normal es recibir valores entre 80 y 400, que varían en función de la persona.

Bibliografía

- [1] D. H. Myszka, *Machines & Mechanisms: Appueda kinematic analysis*. México: Pearson Education, 2012.
- [2] J. J. Uicker, “Teoría de Máquinas y Mecanismos,” no. January, 2016.
- [3] M. I. M. Juan Felipe Solarte Paredes, “APLICACIÓN DE LA HIPÓTESIS VIRTUAL SPRING DAMPER EN EL CONTROL DE POSICIÓN DE UN DEDO DE PRÓTESIS DE MANO ROBÓTICA SUB-ACTUADA: ESTUDIO DE SIMULACIÓN,” 2017.
- [4] O. A. V. Albán, *DISEÑO Y CONTROL DE ROBOTS INDUSTRIALES: TEORÍA Y PRÁCTICA*, 2010.
- [5] M. Tarokh, “[T + l p,” pp. 1192–1197, 1988.
- [6] I. Aaron and C. González, “Diseño Y Construcción De Un Sistema Para La Detección De Señales Electromiográficas,” *Universidad Autónoma de Yucatán*, p. 110, 2010.
- [7] Q. U. E. Para and O. El, ““ Implementación de señales mioeléctricas en un sistema de control para generar movimientos en una prótesis personalizada de mano ”,” 2015.
- [8] P. Konrad, “Noraxon: The ABC of EMG,” *Signal Processing*, no. April, pp. 1–60, 2005. [Online]. Available: <http://demotu.org/aulas/control/ABCofEMG.pdf>
- [9] K. E. Buitrago, “Protocolo de Pruebas,” pp. 1–19, 2015.
- [10] F. Cordella, A. L. Ciancio, R. Sacchetti, A. Davalli, A. G. Cutti, E. Guglielmini, and L. Zollo, “Literature review on needs of upper limb prosthesis users,” *Frontiers in Neuroscience*, vol. 10, no. MAY, pp. 1–14, 2016.
- [11] C. Pylatiuk, S. Schulz, and L. Döderlein, “Results of an internet survey of myoelectric prosthetic hand users,” *Prosthetics and Orthotics International*, vol. 31, no. 4, pp. 362–370, 2007.
- [12] M. Tavakoli, B. Enes, J. Santos, L. Marques, and A. T. de Almeida, “Underactuated anthropomorphic hands: Actuation strategies for a better functionality,” *Robotics and Autonomous Systems*, vol. 74, pp. 267–282, 2015. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0921889015001888>

- [13] D. S. V. Bandara, R. A. R. C. Gopura, G. Kajanathan, M. Brunthavan, and H. I. M. M. Abeynayake, "An under-actuated mechanism for a robotic finger," *4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, IEEE-CYBER 2014*, no. c, pp. 407–412, 2014.
- [14] V. Niola, C. Rossi, and S. Savino, "Tendons Grasping Tests by WM 2D™," pp. 73–77, 2016.
- [15] M. Tavakoli and A. T. De Almeida, "Adaptive under-actuated anthropomorphic hand: ISR-SoftHand," *IEEE International Conference on Intelligent Robots and Systems*, no. Iros, pp. 1629–1634, 2014.
- [16] R. Gajendran, D. C. Tkach, and L. J. Hargrove, "Crosspoint switching of EMG signals to increase number of channels for pattern recognition myoelectric control," *International IEEE/EMBS Conference on Neural Engineering, NER*, pp. 259–262, 2013.
- [17] R. G. E. Clement, K. E. Bugler, and C. W. Oliver, "Bionic prosthetic hands: A review of present technology and future aspirations," *Surgeon*, vol. 9, no. 6, pp. 336–340, 2011. [Online]. Available: <http://dx.doi.org/10.1016/j.surge.2011.06.001>
- [18] M. S. Imbachi, "Ventajas para el gobierno de una prótesis activa de mano mediante señales EMG, derivadas de un mecanismo para agarre auto adaptativo de objetos - Anteproyecto."
- [19] Iván Gustavo Páez Chingal, "SÍNTESIS DE UN MECANISMO DE BARRAS PARA ACTUACIÓN DE UNA PRÓTESIS DE MANO SUBACTUADA ANTROPOMÓRFICA CON CAPACIDAD DE AUTOADAPTACIÓN," p. 29, 2018.
- [20] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in Manipulation Research: The YCB Object and Model Set and Benchmarking Protocols," *arXiv e-prints*, Feb. 2015.
- [21] F. D. Davis, "Information Technology Introduction," vol. 13, no. 3, pp. 319–340, 2013.
- [22] L. Resnik, "Development and testing of new upper-limb prosthetic devices: Research designs for usability testing," vol. 48, no. 6, pp. 697–706, 2011.
- [23] C. A. C. Escudero, *PROPUESTA METODOLÓGICA DE EVALUACIÓN DE EXPERIENCIA DE USUARIO EN AGENCIAS VIRTUALES DE VIAJES*, Chile, 2017.
- [24] J. Sauro, "Measuring usability with the system usability scale (sus)," <https://measuringu.com/sus/>, 2011, accedido 10-12-2018.