

**SIMULACIÓN EN UN ENTORNO DE REALIDAD VIRTUAL DE LA
PREFORMA DE AGARRE EN UNA MANO ROBÓTICA USANDO
VISIÓN ARTIFICIAL**

Lucy Nathalie Cristancho Ovalle

Carlos Andrés Zambrano Dueñas

**Director
Prof. Dr. Oscar Avilés**

**UNIVERSIDAD MILITAR NUEVA GRANADA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA MECATRÓNICA
BOGOTÁ, D.C.**

2010

**SIMULACIÓN EN UN ENTORNO DE REALIDAD VIRTUAL DE LA
PREFORMA DE AGARRE EN UNA MANO ROBÓTICA USANDO
VISIÓN ARTIFICIAL**

Lucy Nathalie Cristancho Ovalle

Carlos Andrés Zambrano Dueñas

**Trabajo de grado para optar al título de
Ingeniero en Mecatrónica**

**Director
Prof. Dr. Oscar Avilés**

**UNIVERSIDAD MILITAR NUEVA GRANADA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA MECATRÓNICA
BOGOTÁ, D.C.**

2010

Nota de aprobación

El Trabajo de grado titulado “**SIMULACIÓN EN UN ENTORNO DE REALIDAD VIRTUAL DE LA PREFORMA DE AGARRE EN UNA MANO ROBÓTICA USANDO VISIÓN ARTIFICIAL**”, elaborado y presentado por los estudiantes *Lucy Nathalie Cristancho Ovalle* y *Carlos Andrés Zambrano Dueñas*, como requisito parcial para optar al título de Ingeniero en Mecatrónica, fue aprobado por el jurado calificador.

Ing. Oscar Fernando Avilés Sánchez M.Sc. Ph.D
DIRECTOR

Ing. Byron Alfonso Pérez Gutiérrez M.Sc.
JURADO

Ing. Wilken Rodríguez Escobar.
JURADO

*A nuestras familias, por la
invaluable paciencia, amor,
sacrificio y apoyo dado durante
nuestra formación profesional.*

Agradecimientos

Los autores expresan sus más sinceros agradecimientos por su orientación y motivación a:

Ing. Oscar Fernando Avilés Sánchez M.Sc. Ph.D., profesor y director del Programa de Ingeniería Mecatrónica de la Universidad Militar Nueva Granada y tutor de este proyecto.

Ing. Byron Alfonso Pérez Gutiérrez M.Sc., profesor y director del Centro de Realidad Virtual de la Universidad Militar Nueva Granada.

Ing. Leonel D Rozo Castañeda M.Sc. Ph.D(e), profesor de Robótica e Inteligencia Artificial de la Universidad Militar Nueva Granada y actualmente estudiante de doctorado en Automática, Robótica y Visión en la Universidad Politécnica de Cataluña.

A los docentes del departamento de Ingeniería Mecatrónica que durante toda nuestra formación nos aportaron tanto al desarrollo profesional como personal con sus conocimientos y experiencias.

Agradecemos también a todas aquellas personas que de una u otra forma nos ayudaron durante el desarrollo del proyecto de grado.

Índice general

1. Introducción	5
1.1. Motivación	5
1.2. Síntesis del trabajo	6
1.3. Objetivos y Metodología de Trabajo	6
1.4. Organización del Trabajo	7
2. Revision de la Literatura	9
2.1. Manos Robóticas	9
2.1.1. UTAH-MIT	10
2.1.2. DLR-HAND	10
2.1.3. TUAT/Karlsruhe	11
2.1.4. UMNG	12
2.1.5. RAPHaEL	13
2.2. Simulación en Ambientes Virtuales	13
2.2.1. Estudio y Desarrollo de Primitivas Motoras para Manipu- lación con Manos Robóticas Antropomorfas	14
2.2.2. DLR Hand II	15
2.2.3. Visualización y Simulación del Agarre de Manos Robóticas	16
2.3. Reconocimiento de formas	17
2.3.1. Detección y Reconocimiento de Señales de Tránsito Uti- lizando Matching de Chamfer.	17

2.3.2.	Reconocimiento del Lenguaje de Gestos Manuales Alfabéticos Mediante Visión Artificial.	18
2.3.3.	Sistema de Reconocimiento de Rostros	18
3.	Fundamentos Teóricos de la Mano	21
3.1.	La mano humana	21
3.1.1.	Mecanismo Extensor de los Dedos	23
3.1.2.	Función de la mano	25
3.1.3.	Modelos prensiles humanos	28
3.2.	Modelo cinemático de la mano humana	30
3.2.1.	Modelo Cadena Cinemática de n articulaciones	32
4.	Reconocimiento de Formas	39
4.1.	Etapas de un problema de reconocimiento de formas	40
4.2.	Métodos para Reconocimiento de Formas	42
4.3.	Postulados De Niemann para el Reconocimiento de Formas	43
4.4.	Aplicaciones de Reconocimiento de formas	44
4.5.	Las Redes Neuronales Artificiales (ANNs)	44
4.5.1.	Modelo de Neurona Artificial	45
4.5.2.	Aprendizaje	47
4.5.3.	Tipos de Aprendizaje	48
4.5.3.1.	Aprendizaje por Corrección de Error.	48
4.5.3.2.	Aprendizaje por Refuerzo	48
4.5.3.3.	Aprendizaje Estocástico	49
4.5.4.	Perceptrón	49
4.5.5.	Adaline	50
4.5.6.	Funciones de Base Radial	51
4.5.7.	Backpropagation	52

5. Validación Experimental	55
5.1. Diseño e Implementación del Programa de identificación de patrones.	55
5.2. Diseño del Entorno Virtual	63
5.3. Diagrama de Flujo del Proceso.	74
6. Conclusiones	77
A. Tutorial basico de VRML	81

Índice de figuras

2.1. Mano Robótica UTAH-MIT[15]	10
2.2. Mano Robótica DLR-Hand.[9]	11
2.3. Mano robótica TUAT/Karlsruhe[8]	11
2.4. Mano Robótica UMNG[10]	12
2.5. Diagrama del Hardware de Actuación de la Mano UMNG 2004[10]	12
2.6. Mano Robótica RAPHaEL[6]	13
2.7. Mano UPC[1]	14
2.8. Mano DLR-Hand II[4]	15
2.9. Simulación de Agarre Mano robótica DLR-Hand II[4]	16
2.10. Visualización y Simulación del Agarre de Manos Robóticas.[2] . . .	16
2.11. Detección y Reconocimiento de Señales de Tránsito[5]	17
2.12. Software de Reconocimiento del Lenguaje de Gestos Manuales.[17]	18
2.13. Interfaz Visual del Sistema de Reconocimiento de Rostros.[14] . . .	19
3.1. Articulaciones y Huesos de la Mano.[3]	22
3.2. Mecanismo Extensor de un Dedo[3]	24
3.3. Posturas de las manos y muñecas (Modelo-Gregori-Blasco-Barrau 2001)	27
3.4. Diagrama de Bloques de las Acciones que Intervienen en la Pren- sión [10]	29
3.5. Modelos Prensiles Humanos a) Cilíndrica Palmar, b) Subterminal Lateral, c) Centralizado, d) Tridigital, e) Posición Subterminal [12] .	29

3.6. Clasificación de los Modelos Prensiles Humanos[13]	30
3.7. Modelo Cinemático de la Mano[10]	31
3.8. Variables Afectadas por la Combinación Articulación – Juntura[10]	33
3.9. Modelo esquemático dedo robot[10]	34
3.10. Modelo cinemático dedo robot, con sus respectivos ejes coordena- dos y orígenes[10]	35
4.1. Etapas de un Problema de Reconocimiento de Formas [16]	41
4.2. Modelo general de una Red Neuronal Artificial	45
4.3. Arquitectura de una Red Neuronal	45
4.4. Ejemplos de Funciones de Salida o Transferencia.	46
4.5. Estructura General de un Sistema de Aprendizaje.	47
4.6. Adaline.	50
4.7. RBF	51
4.8. Representación de Agrupamiento	52
4.9. Red Backpropagation[11]	53
5.1. Diagrama de Flujo Para el Programa de Generación de Datos.	55
5.2. Interfaz Prog. generar datos.	56
5.3. Base de Figuras #1	56
5.4. Base de Figuras #2	57
5.5. Base de Figuras #3	57
5.6. Tablas de parámetros para entranamiento.	58
5.7. Arquitectural general de la adquisición de imágenes y extracción de parámetros.	59
5.8. Entorno Red N	62
5.9. Esquema general de un entorno de RV.	63
5.10. Modelo Original de la Mano con Mecanismos (Vista Superior).	64
5.11. Modelo Original de la Mano con Mecanismos (Vista Frontal).	64

5.12. Modelo de la Falange con Mecanismos.	65
5.13. Modelo de la Mano Modificada sin Mecanismos.	65
5.14. Modelo Modificado de la Falange sin Mecanismos	66
5.15. Ubicación Errónea del Origen.	66
5.16. Corrección de la Ubicación del Origen en el Ensamble.	67
5.17. Exportación de SolidWorks™ a V•Realm™ Builder	68
5.18. Creación de un Escenario en V•Realm™ Builder.	69
5.19. Importación del Modelo CAD a V•Realm™ Builder	69
5.20. Modelo CAD Importado en V•Realm™ Builder	70
5.21. Código en VRMLPAD, El Círculo Rojo Muestra el Objeto Sobre el que se Está Efectuando los Cambios.	72
5.22. Posición, Orientación de Cámaras y Focos de Luz	72
5.23. Vinculación de V•Realm™ Builder con Matlab™ GUI.	73
5.24. Variación de Ángulos de Cada Falange Manualmente.	74
5.25. Diagrama de Flujo del Proceso I de la Simulación.	75
5.26. Diagrama de Flujo del Proceso II de la Simulación.	75
A.1. Insert Box	87
A.2. Rotation	88
A.3. Edit Rotation	88
A.4. Ángulo de Rotación	89
A.5. Translation	89
A.6. Modelo CAD	90
A.7. Exportación	91
A.8. Importación	91
A.9. Mano en VRML	92
A.10. Código	92
A.11. Transformación Falanges	93

Índice de cuadros

3.1. Articulaciones y Límites de la Mano Humana [10]	23
3.2. Movimiento limitado de los dedos[13]	25
3.3. matriz D-H	35
4.1. Aplicaciones de Reconocimiento de formas	44
5.1. Centros de Rotación Dedo 1.	70
5.2. Centros de Rotación Dedo 2.	71
5.3. Centros de Rotación Dedo 3.	71
5.4. Centros de Rotación Dedo 4.	71
5.5. Centros de Rotación Pulgar.	71

Resumen

En la última década se ha construido un considerable número de manos robóticas antropomorfas; sin embargo, muy pocos estudios se han llevado a cabo en relación con la destreza de agarre de dichas manos y sus correspondientes simulaciones en entornos de realidad virtual.

En este trabajo se presenta la simulación de preforma de agarre, por medio de un software de clasificación y la integración de un entorno realidad virtual, en el que se analizan figuras primitivas por medio de visión artificial. Las figuras son clasificadas por medio de una red neuronal, según los parámetros característicos extraídos de cada una de las imágenes capturadas por la cámara. Lo anterior se logra analizando el modelo cinemático de la mano robótica antropomorfa diseñada en CAD, sin tener en cuenta todos los movimientos de la muñeca y los movimientos de abducción y aducción de los dedos, ya que por el modelo mecánico que posee la mano con la que se va a trabajar estos movimientos son imposibles de lograr.

El prototipo final del software de simulación presenta al usuario la posibilidad de seleccionar la cámara que desea utilizar para la captura de la imagen de la figura a analizar, adicionalmente permite modificar algunos parámetros de la red neural implementada para la clasificación de patrones y finalmente la visualización de la preforma de agarre, que toma la mano para cada una de las figuras que reconoce.

Glosario

Abducción: Movimiento de un miembro o un segmento del miembro al separarse de la línea media del cuerpo. Se produce en el hombro, la cadera, la mano y el pie.

Aducción: Movimiento contrario a la abducción, en el cual un miembro o un órgano cualquiera se acercan al plano medio del cuerpo.

Cinemática: Se entiende por cinemática el estudio del movimiento sin considerar las fuerzas que lo producen, por lo tanto se trata de estudiar tanto las propiedades geométricas como las temporales del movimiento.

Cinemática directa: Permite obtener la posición y orientación del final del dedo del manipulador en función de las variables de las articulaciones.

Cinemática inversa: Permite la obtención de las variables articulares que hacen que la posición y orientación del final del dedo sea la deseada.

Movimiento prensil: Movimientos en los cuales un objeto es agarrado y mantenido parcial o totalmente dentro de la mano.

Movimiento no prensil: Movimientos en los cuales no son realizadas acciones de agarre pero los objetos pueden ser manipulados empujándolos o levantándolos con la mano entera o con los dedos individualmente.

Realidad Virtual: Es una representación del mundo a través de medios electrónicos, cuyo objetivo es dar la sensación de estar en una situación real en la que podemos interactuar con lo que nos rodea. Puede ser de dos tipos: inmersiva y no inmersiva.

Realidad Virtual Inmersiva: Utiliza métodos de realidad virtual que con frecuencia se ligan a un ambiente tridimensional creado por computadora el cual se manipula a través de cascos, guantes u otros dispositivos que capturan la posición y rotación de diferentes partes del cuerpo humano.

Realidad Virtual No Inmersiva: Utiliza medios como el que actualmente nos ofrece Internet, en el cual podemos interactuar a tiempo real con diferentes personas en espacios y ambientes que en realidad no existen sin la necesidad de dispositivos adicionales a la computadora.

Red Neuronal Artificial: Consiste en una simulación de las propiedades observadas en los sistemas neuronales biológicos a través de modelos matemáticos recreados mediante mecanismos artificiales (como un circuito integrado, un ordenador o un conjunto de válvulas). El principal objetivo es conseguir que las máquinas presenten respuestas similares a las que es capaz de dar el cerebro que se caracterizan por su generalización y su robustez.

Capítulo 1

Introducción

La mano es una de las herramienta más importante que posee el ser humano, al ser el principal órgano de manipulación física, que permite tomar y sostener objetos con gran precisión, debido a la gran versatilidad de movimientos que es capaz de realizar. Permite llegar a crear lenguajes de señales y gracias a la sensibilidad con la que cuentan se convierten en instrumento de lectura para personas con discapacidades visuales. Las anteriores tareas son solo algunas de las más importantes que la mano humana es capaz de realizar. Por este motivo al ser humano le surge la idea de imitar por medio de mecanismos la estructura de la mano para poder simular la gran cantidad de movimientos con los que cuenta este órgano; ya que poder sujetar y manipular objetos de diferentes formas y tamaños, crea un reto desde el punto de vista mecánico y adicionalmente agrega un problema de aprendizaje, convirtiéndose estos dos temas en el principal enfoque de este trabajo.

El avance de los sistemas de visión artificial aporta mayores ventajas a los robots en la manipulación de objetos, integrando los sistemas de visión y sensores táctiles para proporcionar una mayor precisión en el control de los dedos de una mano robótica.

1.1. Motivación

Sabiendo con anterioridad que en la Universidad Militar Nueva Granada se han desarrollado trabajos enfocados en el diseño y construcción de manos robóticas antropomórficas y poco se ha implementado en la inclusión de estos diseños en

ambientes de realidad virtual, con el fin de analizar los movimientos de agarre dependiendo de una realimentación visual.

En la actualidad hay procesos que requieren ser simulados para un mejor desempeño en la industria o en la realización de tareas de precisión; como es el caso de procesos industriales donde se manipulan sustancias peligrosas o elementos muy costosos, tele operaciones, en el entrenamientos de robots, entre otros. Por este motivo surge la necesidad de desarrollar un software que simule los movimientos de los dedos y vincule la visión artificial, con el objetivo de realizar tareas autónomas a la hora de reconocer formas primitivas y determinar su agarre.

1.2. Síntesis del trabajo

Se pretende implementar una mano antropomorfa que se encuentra en un ambiente de realidad virtual y tendrá una estrategia de percepción motora para que simule el movimiento de agarre de objetos, los cuales serán capturados por una cámara y dependiendo de la forma de dicha figura, el entorno virtual mostrara diferentes formas de agarre para el objeto. Al momento de realizar la implementación de una estrategia de percepción motora en un dispositivo robótico deben probarse diferentes estrategias, hasta conseguir el resultado deseado; usualmente las estrategias de agarre más efectivas se retienen y se va estableciendo un esquema motor preferido para cada forma.

Con dichas pruebas se podrá ver la efectividad del diseño propuesto y se realizaran cambios o mejoras en la propuesta para que este proyecto pueda ser utilizado en el campo de la medicina aportando a la ciencia, en el campo educativo como tema para futuros trabajos de investigación o grado y a nivel industrial con el objetivo de simular procesos donde se emplea el agarre.

1.3. Objetivos y Metodología de Trabajo

Objetivo General

Diseñar e implementar un software de simulación de agarre en una mano robótica antropomórfica que se encuentra en un ambiente de realidad virtual que a partir de visión artificial visualizara figuras primitivas.

Objetivo Especificos

- Implementar un software de adquisición y procesamiento de imágenes.
- Implementar una red neuronal para clasificar figuras.
- Estudiar los movimientos de agarre de una mano humana.
- Implementar una base de conocimiento donde se encuentran las posibles formas de agarre para cada figura.
- Diseñar un entorno virtual donde se representaran los movimientos de una mano robótica.

Metodología de Trabajo

- A. Estudiar las tecnicas existentes que permita implementar el proyecto.
- B. Estudio de procedimientos actuales sobre identificación de patrones.
- C. Diseño e implementación de un programa que identificación de patrones.
- D. Diseño e implementación del modelo de una mano antropomorfa en un entorno de realidad virtual.
- E. Realización de pruebas.
- F. Documentación del trabajo realizado.
- G. Socialización de resultados.

1.4. Organización del Trabajo

El proyecto de grado está organizado en 6 capítulos, así: El primer capítulo se encuentra una introducción del trabajo, los objetivos y metodología de trabajo. El segundo capítulo introduce historia sobre algunos trabajos relacionados con manos robóticas, reconocimientos y simulación en entornos virtuales. El tercer capítulo trata sobre el estudio biomecánico de la mano, que incluye una breve descripción de su funcionamiento, su modelo cinemático y el agarre. El cuarto capítulo expone sobre el reconocimiento de formas, centralizando en el tratamiento de imágenes y las redes neuronales. El quinto capítulo contiene la validación experimental donde se describen los procesos de realización de los programas y simulaciones, junto con algunas pruebas realizadas. Finaliza el proyecto de grado con el capítulos seis, el cual está dedicado a las conclusiones obtenidas en el desarrollo del proyecto.

Capítulo 2

Revision de la Literatura

En el transcurso de los últimos años la incorporación de las manos robóticas como efectores finales de manipuladores y como punto central de investigación, ha producido grandes avances, ya que cada vez las tareas realizadas por estos dispositivos son más precisas y con mayor destreza. Produciendo adelantos basados en el estudio y diseño del agarre humano, gracias a la vinculación de otras ramas de la investigación como lo son la visión artificial o visión por computador y la realidad virtual, generando directamente acciones virtuales mediante realimentación visual.

Varios investigadores han realizado estudios sobre mecanismos, para llevar a cabo la sujeción y manipulación de diferentes objetos, enfocándose en trabajos relacionados con manos robóticas antropomórficas con mecanismos sub-actuados e incluso con un control muy sofisticado, simulación en ambientes virtuales y reconocimiento de formas, de los que se destacan y sirvieron de apoyo para la realización de este trabajo.

2.1. Manos Robóticas

A continuación se presentan una serie de trabajos en donde se han empleado manos robóticas para diferente tipo de tareas, cada una nos muestra una propuesta diferente desde el punto de vista del diseño y funcionamiento mecánico hasta sus diferentes aplicaciones y objetivos de implementación.

2.1.1. UTAH-MIT

En 1984 se desarrolló la mano robótica “UTAH-MIT”, la cual cuenta con características antropomórficas similares a las de la mano. Cuenta con tres dedos y un pulgar, cada uno con cuatro grados de libertad empleando un total de 32 actuadores para proporcionar el movimiento y la fuerza de agarre de 31N en cada dedo. Provee una manipulación precisa de objetos, ya que las articulaciones se mueven por medio de un par de tendones y cada tendón es actuado por un cilindro neumático.

Debido a la compleja estructura cinemática, el control necesitaba ser muy sofisticado, lo que generaba alto costo computacional[15].

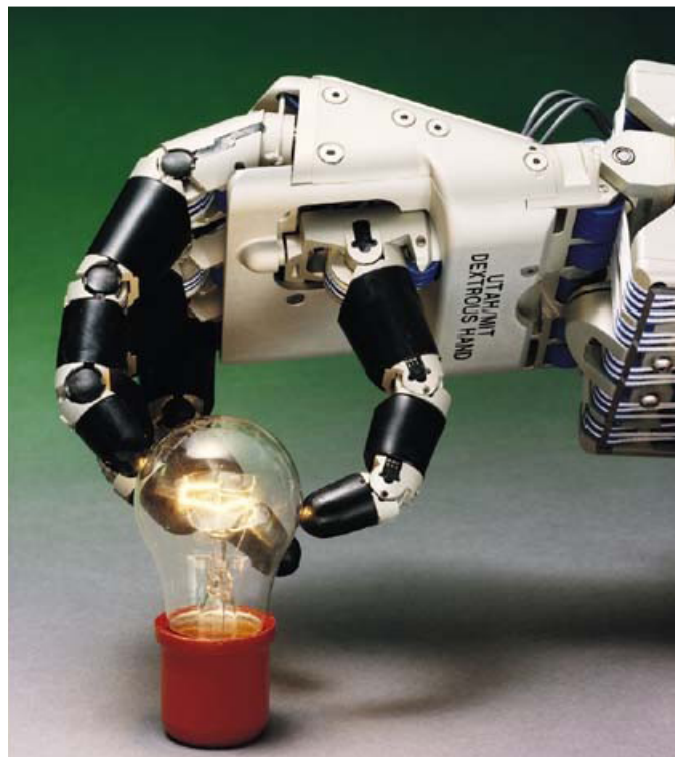


Figura 2.1: Mano Robótica UTAH-MIT[15]

2.1.2. DLR-HAND

En 1998, se presentó el diseño de una mano robótica multi articulada con cuatro dedos, semi-antropomórfica llamada DLR Hand . La mano cuenta con sensores de torque, fuerza y posición en cada dedo y todos los actuadores se encuentran montados directamente en los dedos, como se observa en la figura [9].



Figura 2.2: Mano Robótica DLR-Hand.[9]

2.1.3. TUAT/Karlsruhe

En el 2000 el departamento de tecnología de la Universidad de Tokio Japón, en conjunto con la Universidad de Karlsruhe en Alemania , desarrollaron el diseño de una mano robótica humanoide la cual cuenta con cuatro dedos de 4 grados de libertad cada uno, denominada TUAT/Karlsruhe. Esta mano humanoide, mostrada en la figura 2.3, es capaz de sostener un objeto con los dedos y la palma, la palma se puede adaptar completamente a la forma del objeto que se esta sujetando.

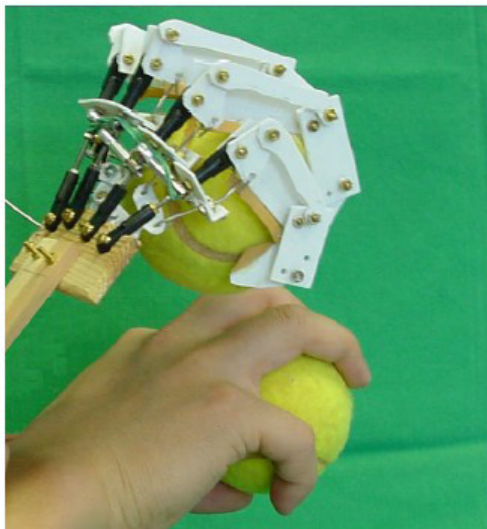


Figura 2.3: Mano robótica TUAT/Karlsruhe[8]

2.1.4. UMNG

En el año 2004 la Universidad Militar Nueva Granada presenta un trabajo de grado titulado “Diseño y Construcción de una Mano Robot de Cuatro Dedos que Imita los Modelos Prensiles Humanos”, en donde se realizó el diseño mecatrónico de una mano robot para el agarre de objetos de forma regular; involucrando el estudio y desarrollo mecánico, electrónico y software de control elaborado para el funcionamiento del proyecto, realizando énfasis en el diseño mecánico y construcción de la mano robot. El resultado de este trabajo fue la elaboración de una mano robot que cumple con los modelos básicos de la presión humana, permitiendo una mejor interacción entre el robot manipulador que la porta y su ambiente, logrando así mayor capacidad y eficiencia en labores que antes solo podían ser realizadas por manos humanas, o por grippers específicos para cada fin [10]. En las figuras 2.4 y 2.5 se observa la mano artificial mencionada anteriormente.

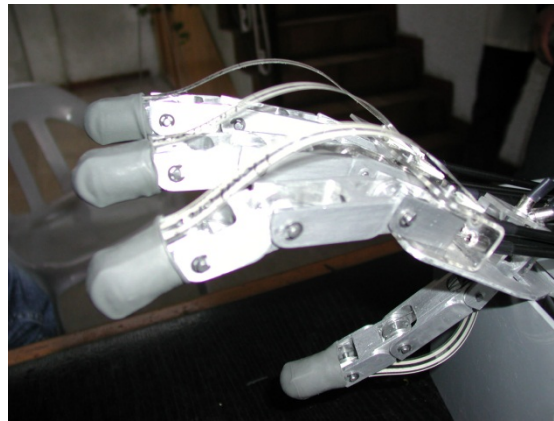


Figura 2.4: Mano Robótica UMNG[10]

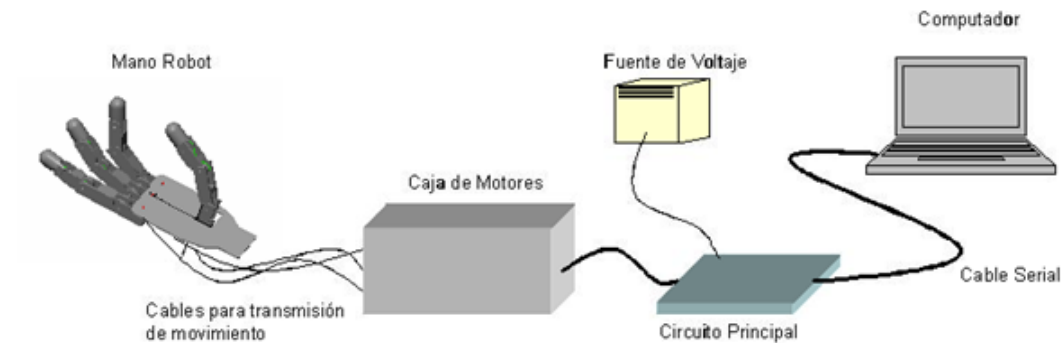


Figura 2.5: Diagrama del Hardware de Actuación de la Mano UMNG 2004[10]

2.1.5. RAPHaEL

Esta mano robótica mostrada en la figura 2.6, fue desarrollada en el Laboratorio de Robótica y Mecánica (RoMeLa) de la Facultad de Ingeniería de Virginia Tech en el 2009. La mano es capaz de sujetar desde una pesada lata hasta un delicado foco de una manera firme. Es articulada por un compresor de aire de 60 psi que mediante un microcontrolador coordina los movimientos de los dedos sustituyendo así el uso de voluminosos motores o servomecanismos gracias a el diseño de los impulsores de aire comprimido, permitiendo que esta mano robot haga movimientos suaves, naturales, con precisión y mucha delicadeza, ya que la fuerza de sus dedos puede ser ajustada con sólo con cambiar la presión del aire. El ajuste de la potencia del aire comprimido, junto a la presencia de los ligamentos elásticos, hacen que los dedos sigan el contorno del objeto que sostienen con total naturalidad[6].

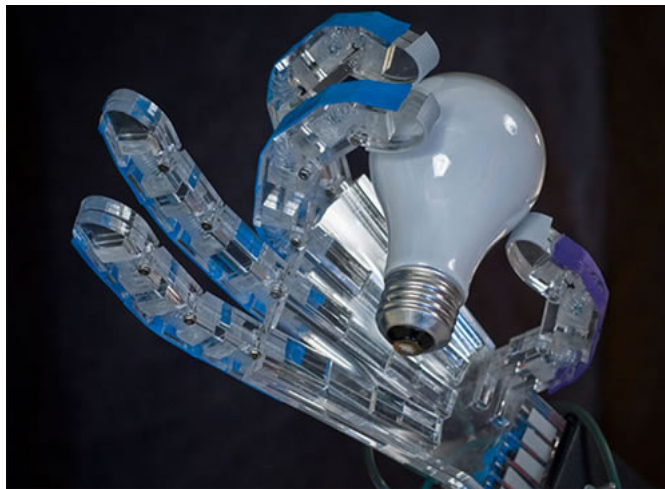


Figura 2.6: Mano Robótica RAPHaEL[6]

Robotic Air Powered Hand with Elastic Ligaments o RAPHaEL forma parte del proyecto CHARLI (Cognitive Humanoid Robot with Learning Intelligence) que es un robot humanoide bípedo de 5 pies de altura.

2.2. Simulación en Ambientes Virtuales

En la actualidad la simulación de procesos en ambientes de realidad virtual ha ido creciendo paulatinamente junto con el avance de los computadores y lengua-

jes de programación; a continuación presentamos algunos trabajos de simulación de manos en ambientes de realidad virtual.

2.2.1. Estudio y Desarrollo de Primitivas Motoras para Manipulación con Manos Robóticas Antropomorfas

A nivel internacional se encuentra en España el artículo “Estudio y Desarrollo de Primitivas Motoras para Manipulación con Manos Robóticas Antropomorfas”, realizado en el año 2001 en la Universidad Politécnica de Cartagena, en donde se presentan una serie de estudios relacionados con la manipulación de precisión por parte de manos robóticas antropomorfas. Se entiende manipulación de precisión al control cinemático de la posición del objeto agarrado, utilizando únicamente los contactos de las puntas de los dedos con el objeto. La aproximación utilizada, define un conjunto de primitivas de manipulación básicas que la mano puede llevar a cabo, como la mostrada en la figura 2.7. Estas funciones primitivas son flexibles ya que incluyen una serie de parámetros asociados a la geometría y tamaño de los objetos a manipular y además, permiten su implementación partiendo de distintas topologías de agarre iniciales. En este trabajo también se describe como estas primitivas de manipulación se pueden emplear de forma secuencial para dar lugar a tareas más complejas [1].

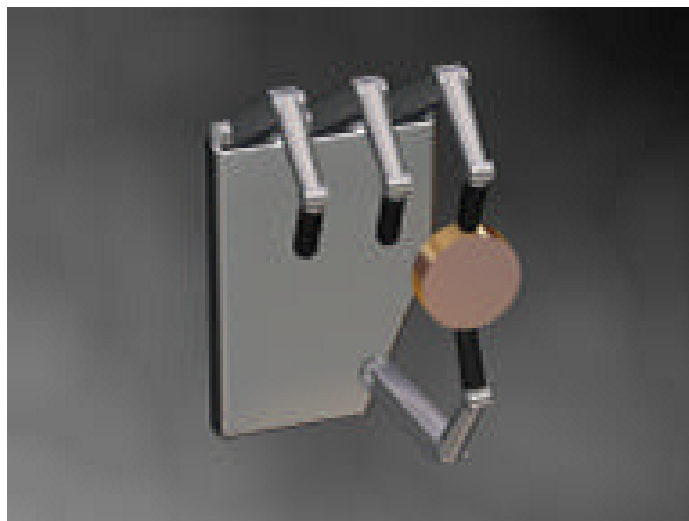


Figura 2.7: Mano UPC[1]

2.2.2. DLR Hand II

En el año 2001, Butterfass, presenta las mejoras hechas al diseño de la mano multiarticulada DLR-HAND . En éste nuevo diseño se presenta una estructura denominada “Esqueleto Abierto”, la cual proporciona una mayor facilidad para acceder a los circuitos, mecanismos y sensores de la mano. La palma de la mano esta equipada con actuadores más poderosos y con sensores de fuerza y torque de 6 grados de libertad en cada dedo, como se muestra en la figura 2.8XX. Para obtener una mayor precisión en la manipulación de objetos mediante manipuladores robóticos, es necesario determinar la técnica de agarre más adecuada a la estructura del efector final . Por medio del análisis y simulación de las técnicas de agarre que se programarán en los efectores finales, como puede observarse en la figura 2.9 .

Se hace el análisis de la configuración más adecuada de la mano DLR-HAND II necesaria para realizar trabajos que requieren de una gran destreza , como lo es el manejo de desarmadores y herramienta especializada, obteniendo una aproximación de la posición de la mano robótica y los dedos en la tarea asignada[4].



Figura 2.8: Mano DLR-Hand II[4]

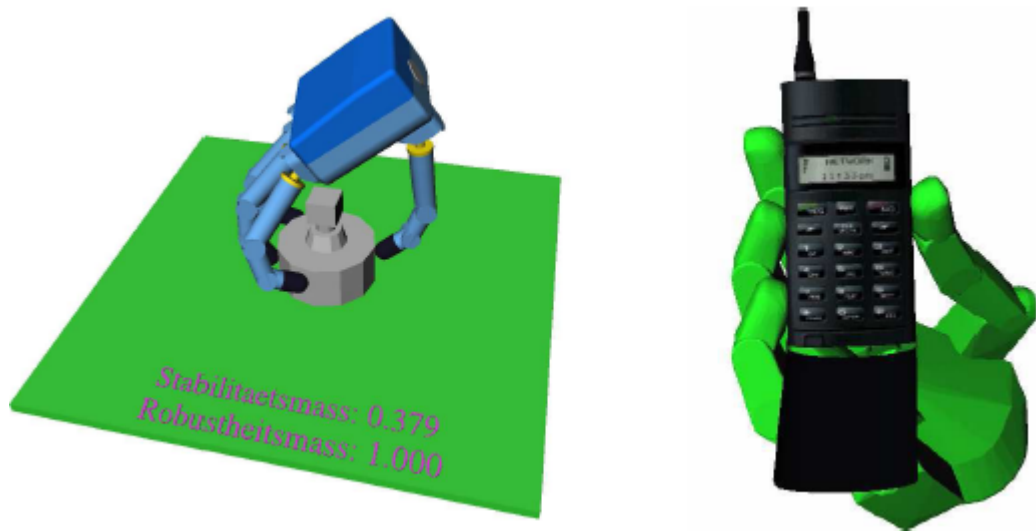


Figura 2.9: Simulación de Agarre Mano robótica DLR-Hand II[4]

2.2.3. Visualización y Simulación del Agarre de Manos Robóticas

En el año 2003 se presentan un simulador interactivo que puede ser usado con una gran variedad de manos robóticas y manipuladores con la finalidad de obtener información acerca del tipo de agarre fuerzas y posibles colisiones con el entorno. El simulador puede detectar los puntos de contacto que se producen entre el objeto a manipular y la mano robótica. El simulador, mostrado en la figura 2.10, provee un ambiente ideal para la planeación de agarres y puede servir como método de prueba en el análisis y síntesis del agarre y control [2].

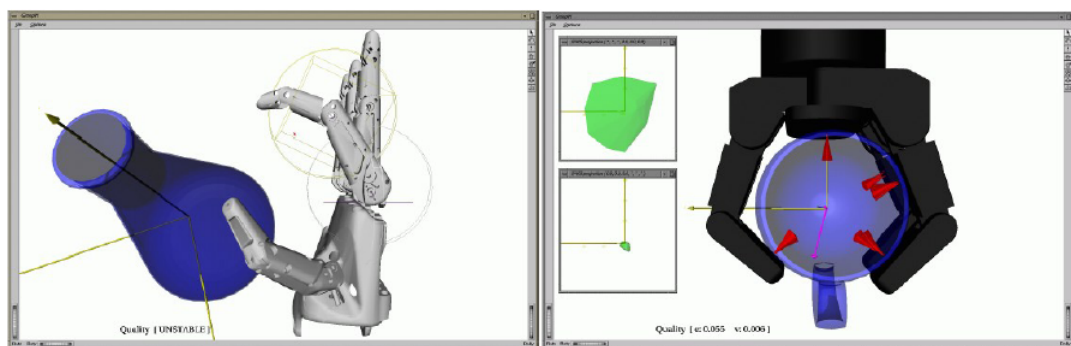


Figura 2.10: Visualización y Simulación del Agarre de Manos Robóticas.[2]

2.3. Reconocimiento de formas

El reconocimiento de formas y clasificación de patrones, es uno de los temas que están en un constante avance, ya que cada día se mejoran las cámaras, los equipos de procesamiento y los métodos tanto matemáticos, estadísticos y de inteligencia artificial, utilizados para realizar este tipo de tareas. A continuación se muestran algunos trabajos enfocados en diferentes aplicaciones de la vida cotidiana.

2.3.1. Detección y Reconocimiento de Señales de Tránsito Utilizando Matching de Chamfer.

En este proyecto se presenta un prototipo que permite ayudar a un conductor de un vehículo a colocar atención a las señales del tránsito que estén postadas en la vía, pretendiendo asistir al conductor, y por ende evitar infracciones o accidentes. El prototipo desarrollado con técnicas de visión computacional permite detectar y reconocer las señaléticas presentes en la vía e informar su naturaleza al conductor mediante una señal audible o proyección visual. La investigación se centró principalmente en las fases de detección inicial; con el fin de contar con una heurística rápida, aprovechando la segmentación por color, con sus características de invariabilidad del sistema HSV (Brillo, Saturación, Valor), y/o detección inicial por bordes, aprovechando el algoritmo de Chamfer optimizado. Para finalmente detectar y reconocer los símbolos de la señal, utilizando técnicas de transformación distancia y matching jerárquico de Chamfer, acondicionadas para este tipo de aplicación. El prototipo en su fase de prueba fue implementado en Matlab como se puede ver en la figura 2.11, con el propósito inicial de comprobar la efectividad de los métodos empleados. Una vez comprobados se utilizó OpenCV para probar su funcionamiento en tiempo real[5].



Figura 2.11: Detección y Reconocimiento de Señales de Tránsito[5]

2.3.2. Reconocimiento del Lenguaje de Gestos Manuales Alfabéticos Mediante Visión Artificial.

En el año 2009, en la Universidad Distrital Francisco José De Caldas se diseñó e implementó un sistema con la capacidad de interpretar los signos alfabéticos manuales haciendo uso de herramientas como visión artificial y redes neuronales artificiales, titulado “Reconocimiento del Lenguaje de Gestos Manuales Alfabéticos Mediante Visión Artificial”. Una forma de comunicación, que no requiere el empleo del lenguaje hablado, se puede efectuar usando las manos, con las cuales se puede simbolizar el alfabeto completo ejecutando símbolos y movimientos estándar de comunicación no audible a partir de gesturas visuales.

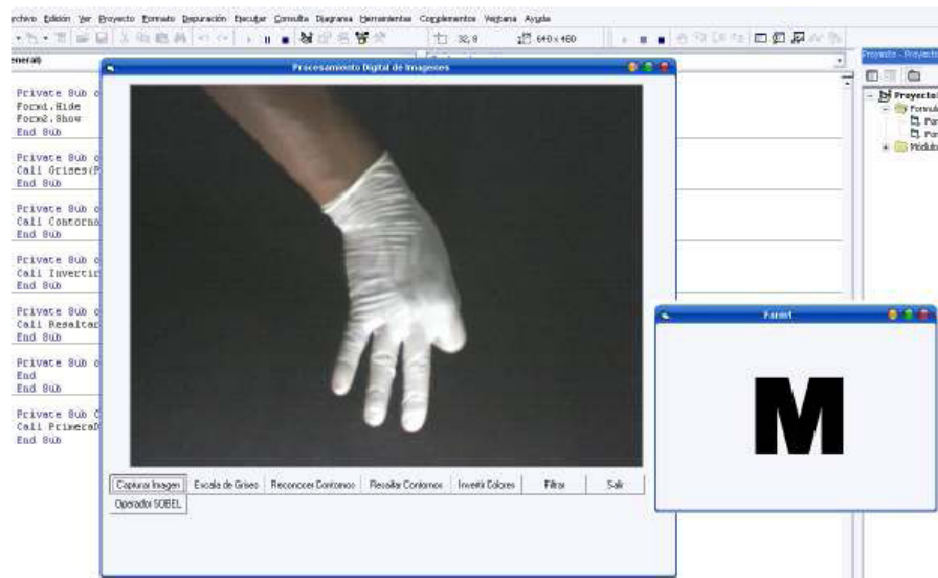


Figura 2.12: Software de Reconocimiento del Lenguaje de Gestos Manuales.[17]

Con el fin de reconocer estas mismas, se implementó un sistema que capta el símbolo que está representando la mano como se observa en la figura 2.12, para ser clasificado como una letra del alfabeto ante el usuario que desconoce el código de gesturas[17].

2.3.3. Sistema de Reconocimiento de Rostros

Este proyecto propone un sistema computacional capaz de reconocer rostros a partir de imágenes faciales capturadas a través de una cámara web. El sistema compara paramétricamente la imagen adquirida con aquellas almacenadas en una base

de datos (usuarios registrados). Para ello, se desarrolló una aplicación en MATLAB, utilizando la interfaz visual Guide. En esta aplicación se pueden controlar los parámetros utilizados para el proceso de reconocimiento del rostro, ya que este tiene características únicas como distancia entre los ojos, anchura de la nariz, forma de la barbilla, pómulos, forma de la boca, etc [14].

El método de reconocimiento empleado es basado en el procesamiento de imágenes “eigenfaces”. Actualmente el sistema se encuentra en la fase de ajuste y calibración de parámetros.

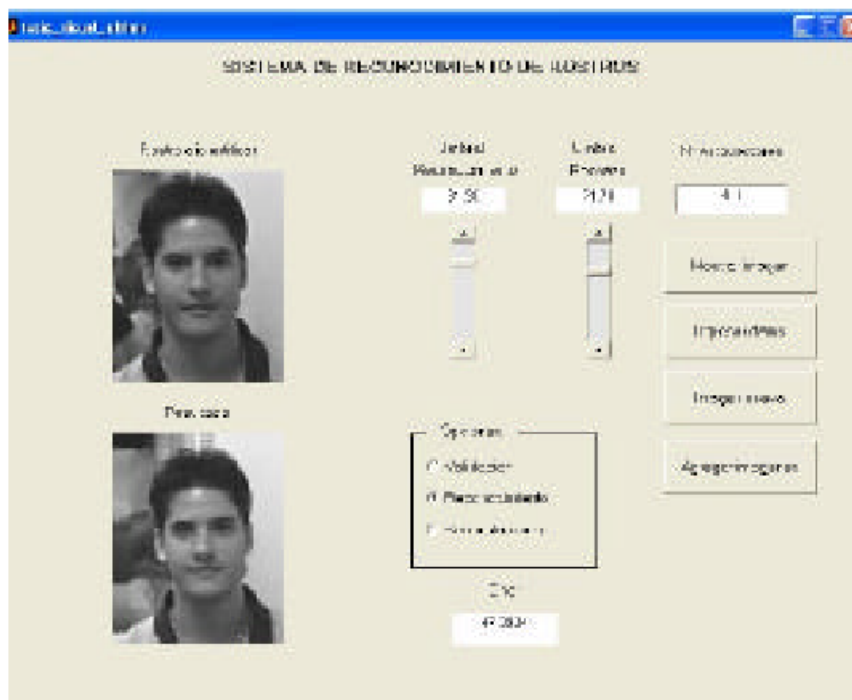


Figura 2.13: Interfaz Visual del Sistema de Reconocimiento de Rostros.[14]

Capítulo 3

Fundamentos Teóricos de la Mano

A continuación se presentan las bases necesarias para comprender el estudio de la anatomía de la Mano, la cinemática relacionada con los efectores finales de dedos Multi-articulados, así como criterios correspondientes a los diferentes tipos de agarre que se presentan durante la sujeción de diversos objetos.

3.1. La mano humana

La mano es uno de los mecanismos más complejos del cuerpo humano con un poco más de 25 grados de libertad. Las partes esqueléticas de la mano humana incluyen los huesos de la muñeca, los huesos de la palma y los huesos de los dedos.

La mano cuenta con 27 huesos incluyendo los de la muñeca, 14 falanges en los dedos, 5 metacarpianos y 8 carpianos, estos huesos constituyen los 4 grupos principales de articulaciones de la mano, denominadas:

- Articulación Distal Interfalángica (DIF).
- Proximal Interfalángica (PIF).
- Metacarpofalángica (MCF).
- Carpometacarpiana (CMC)

Como se muestra en la figura [3].

Cada uno de los dedos cuenta con tres falanges, falange distal, media y proximal, mientras que el dedo pulgar solo cuenta con falange distal y proximal, la longitud

de las falanges incrementa conforme se acercan a los metacarpianos, es decir, la falange distal es de menor longitud mientras que la falange proximal es la más larga de las tres.

La cabeza de la falange proximal y media es bicondiliar, los metacarpianos están ubicados atrás de la falange proximal de cada uno de los dedos, agrupándose en un eje arqueado para formar la curvatura de la palma. En los dedos, la falange distal y media forman la articulación distal interfalángica (DIF), la falange media y proximal forman la articulación proximal interfalángica (PIF) y la falange proximal y el metacarpiano forman la articulación metacarpofalángica (MCF).

Los carpianos adquieren su nombre dependiendo de la forma del hueso, se encuentran en un arreglo de dos filas que están unidas estrechamente a los ligamentos ínter óseo, en la fila distal se encuentran el trapecio, el trapezoide, el unciforme, y el encorvado, desde la parte lateral hasta la parte central. Los huesos de toda la fila proximal son el escafoides, el ulnar, piramidal y el pisiforme, la articulación entre la fila proximal y distal se llama Mediocarpal, mientras que la articulación entre los huesos adyacentes se llama Intercarpales[3].

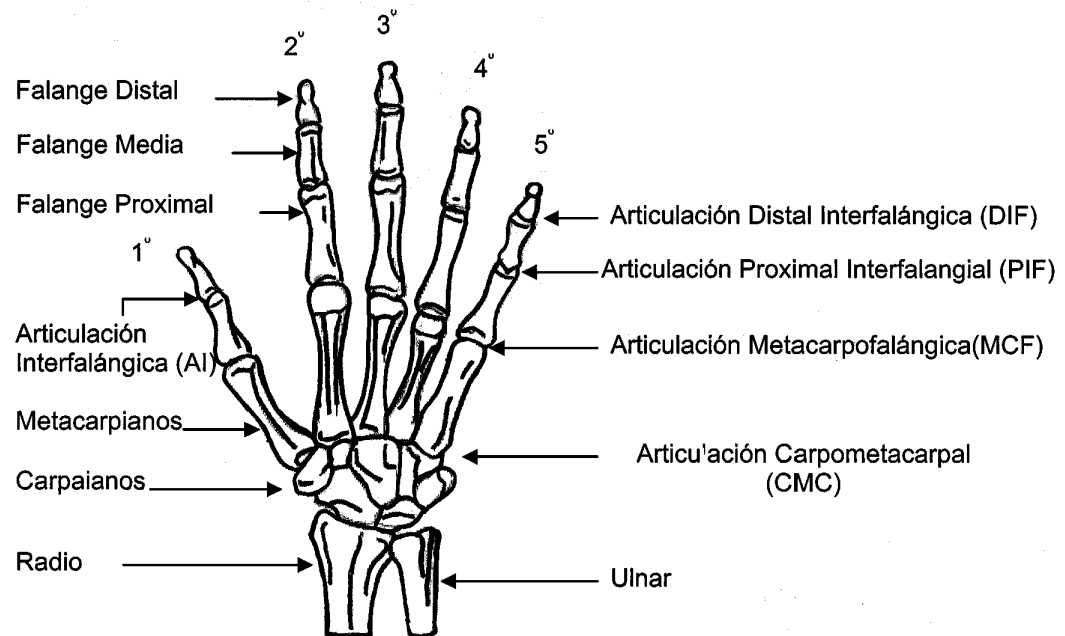


Figura 3.1: Articulaciones y Huesos de la Mano.[3]

La articulación MCF de los dedos, es un ejemplo de articulaciones condiliares, donde el cóndilo oval del hueso metacarpiano se articula con la cavidad de las

falanges proximal, ésta estructura permite movimientos angulares en dos direcciones por lo que se considera una articulación biaxial, proporcionando la capacidad de realizar movimientos de flexión-extensión y abducción-aducción.

Las articulaciones que se encuentran entre las falanges poseen una estructura parecida a la de una bisagra que permite únicamente movimiento en un solo eje (mono axial), cabe destacar que la cara de los lados proximales de la falange media y distal son cóncavas y la cara distal son convexas[3].

En la tabla 3.1, se observan los grados de libertad de cada junta y el rango de movimiento aproximado posible para cada uno de los dedos. A pesar de tener más o menos 25 grados de libertad, algunos de estos grados de libertad son independientes. Estas interdependencias surgen de las interconexiones musculares de la mano.

ARTICULACIONES DE LA MANO				
Parte	Juntura	DOF	Angulo Grados Flexión-Extensión	Angulo Grados Abducción-Aducción
Dedos	DIP	1	60	-
“	PIP	1	100	-
“	MCP	2	90	60
Pulgar	IP	1	85	-
“	MCP	2	50	30
“	CMC	2	120	45

Cuadro 3.1: Articulaciones y Límites de la Mano Humana [10]

3.1.1. Mecanismo Extensor de los Dedos

El mecanismo extensor de los dedos esta compuesto por los tendones intrínseco y extrínseco. A través del mecanismo la inserción intrínseca, controla el movimiento de la articulación PIF (proximal interfalángica) y MCF (metacropofalángica) para proporcionar un movimiento fino en cada uno de los dedos . La estructura del tendón extensor largo extrínseco esta formada por 4 componentes principalmente:

1. Bandas Sagitales que circunscriben la cabeza de los metacarpianos.
2. La Corredera Proximal que se inserta en la base de la falange proximal.
3. La Corredera Central que se inserta en la base de la falange intermedia.
4. Las Correderas Laterales que se insertan en la base de la falange distal.

En la figura 3.2 se muestra el mecanismo extensor de un dedo.

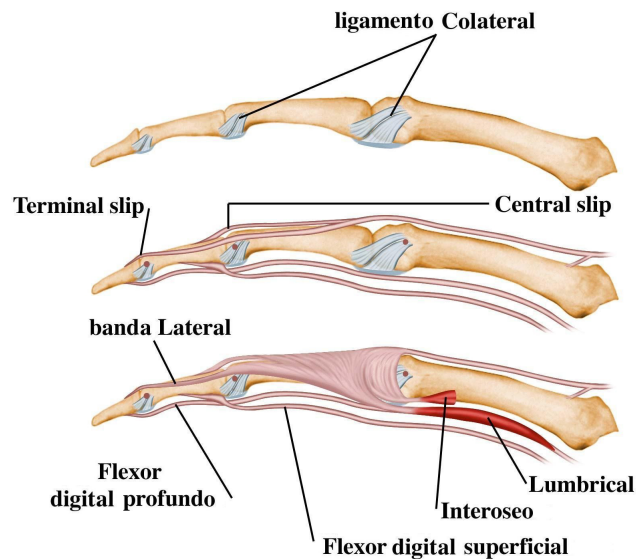


Figura 3.2: Mecanismo Extensor de un Dedo[3]

Al flexionar la articulación MCF, las bandas sagitales se deslizan hacia adelante para aplicar fuerza extensora a lo largo de la corredera central, extendiendo la falange intermedia. Al flexionar la articulación interfalángica proximal (PIF), las correderas laterales se deslizan palmares al eje de rotación para aflojar y permitir la flexión de la articulación interfalángica distal (DIF).

El mecanismo extensor de la articulación distal interfalángica (DIF) esta conformado por los siguientes tendones y ligamentos:

1. Fibras Arqueadas oblicuas que atraviesan transversalmente la corredera central y las bandas laterales.
2. Ligamento transverso de Landsmeer, que atraviesa el dorso de la articulación sobre la línea de montaje.
3. El ligamento triangular que se extiende entre las inserciones de las bandas laterales y la falange distal.
4. El ligamento retinacular oblicuo de Landsmeer, que se extiende desde la inserción de los extensores en la falange distal sobre el eje de la articulación PIF (interfalángica proximal) y sobre la línea de la articulación.

El mecanismo extensor del dedo pulgar es básicamente una forma modificada del mecanismo de la articulación DIF de los demás dedos, ya que no hay falange intermedia. La porción extrínseca esta integrada por el extensor corto del pulgar que corre paralelo al centro del de la articulación MCF para insertarse sobre la falange proximal y al extensor largo del pulgar que corre sobre la cara cubital de la articulación MCF para insertarse en la base de la falange distal [3].

El movimiento limitado de los dedos como consecuencia de la anatomía de la mano, observada en la Tabla3.2 .

MANO HUMANA	MUC-I
$-90^{\circ} \leq \theta_1 \leq 90^{\circ}$	$\theta_1 = 0^{\circ}$
$-15^{\circ} \leq \theta_2 \leq 15^{\circ}$	$\theta_2 = 0^{\circ}$
$-15^{\circ} \leq \theta_3 \leq 15^{\circ}$	$\theta_3 = 0^{\circ}$
$-15^{\circ} \leq \theta_{4m} \leq 15^{\circ}$	$\theta_{4m} = 0^{\circ}$
$0^{\circ} \leq \theta_{5m} \leq 90^{\circ}$	$0^{\circ} \leq \theta_{5m} \leq 80^{\circ}$
$0^{\circ} \leq \theta_{6m} \leq 110^{\circ}$	$\theta_{6m} = f(\theta_{5m})$
$0^{\circ} \leq \theta_{7m} \leq 90^{\circ}$	$\theta_{7m} = f(\theta_{6m})$

Cuadro 3.2: Movimiento limitado de los dedos[13]

3.1.2. Función de la mano

Para poder entender los alcances de la funcionalidad de la mano es necesario describir algunos movimientos importantes que esta puede realizar facilmente ya que estos son básicos pero vitales para los diferentes movimientos que esta puede realizar.

Posición de referencia, es aquella a partir de la cuál se miden los movimientos articulares.

Flexión, se denomina así al movimiento consistente en doblarse o disminuir el ángulo entre dos partes del cuerpo, podemos decir que es un movimiento en el cual un segmento del cuerpo se desplaza en un plano sagital con respecto a un eje transversal, aproximándose al segmento adyacente.

Extensión, esta consiste en enderezarse, o aumentar el ángulo entre dos segmentos del cuerpo, es un movimiento sagital respecto a un eje transversal tal que, desde una posición de flexión, retorna a la posición del cuerpo de referencia o la sobrepasa.

Abducción, este movimiento consiste en acercar a la línea media del cuerpo, el movimiento se efectúa en el plano frontal, en torno de un eje antero-posterior, que

aproxima el segmento corporal comprometido a la línea media.

Abducción de la mano, consiste en separa los dedos, uno de otro, en un plano Aducción, en este caso el movimiento consiste en alejarse de la línea media del cuerpo, movimiento que se efectúa en un plano frontal, en torno de un eje antero-posterior, el segmento corporal se aleja de la línea media. **Aducción del pulgar**, es extenderlo o flexionarlo en torno a la palma de la mano.

Aducción de la mano, consiste en cerrar los dedos uno contra otro, en un plano.

Pronación, el movimiento consiste en hacer girar el antebrazo de tal modo que la palma de la mano quede hacia abajo.

Supinación, consiste en hacer girar el antebrazo de tal modo que la palma de la mano quede hacia arriba Circunducción, este movimiento consiste en que una parte del cuerpo describe un cono cuyo vértice está en la articulación y su base en la extremidad distal de esa parte y no necesita rotación. **Prehensión**, acción de tomar envolviendo un objeto, los dedos se cierran en torno al objeto envolviéndolo.

Pinza, acción de tomar con las puntas de los dedos opuestos.

Hiperextensión de los dedos, empujar con los dedos estando la mano en posición neutra.

Pinza palpar, tomar un objeto con los dedos índice, mayor, anular y meñique, (los cuales se flexionan cuando se esta sujetando un objeto). También se define así, la toma por oposición entre el pulgar y otro dedo opuesto solamente.

Compresión digital, es la acción de presionar en forma plana con los dedos.

Compresión pulpar, es la acción de presionar un objeto con la palma de la mano.

en la figura3.3 podemos ver las pocisiones anteriormente descritas para una mejor comprensión además una descripción de algunos movimientos en donde permite únicamente unos grados máximos de apertura o cierre.

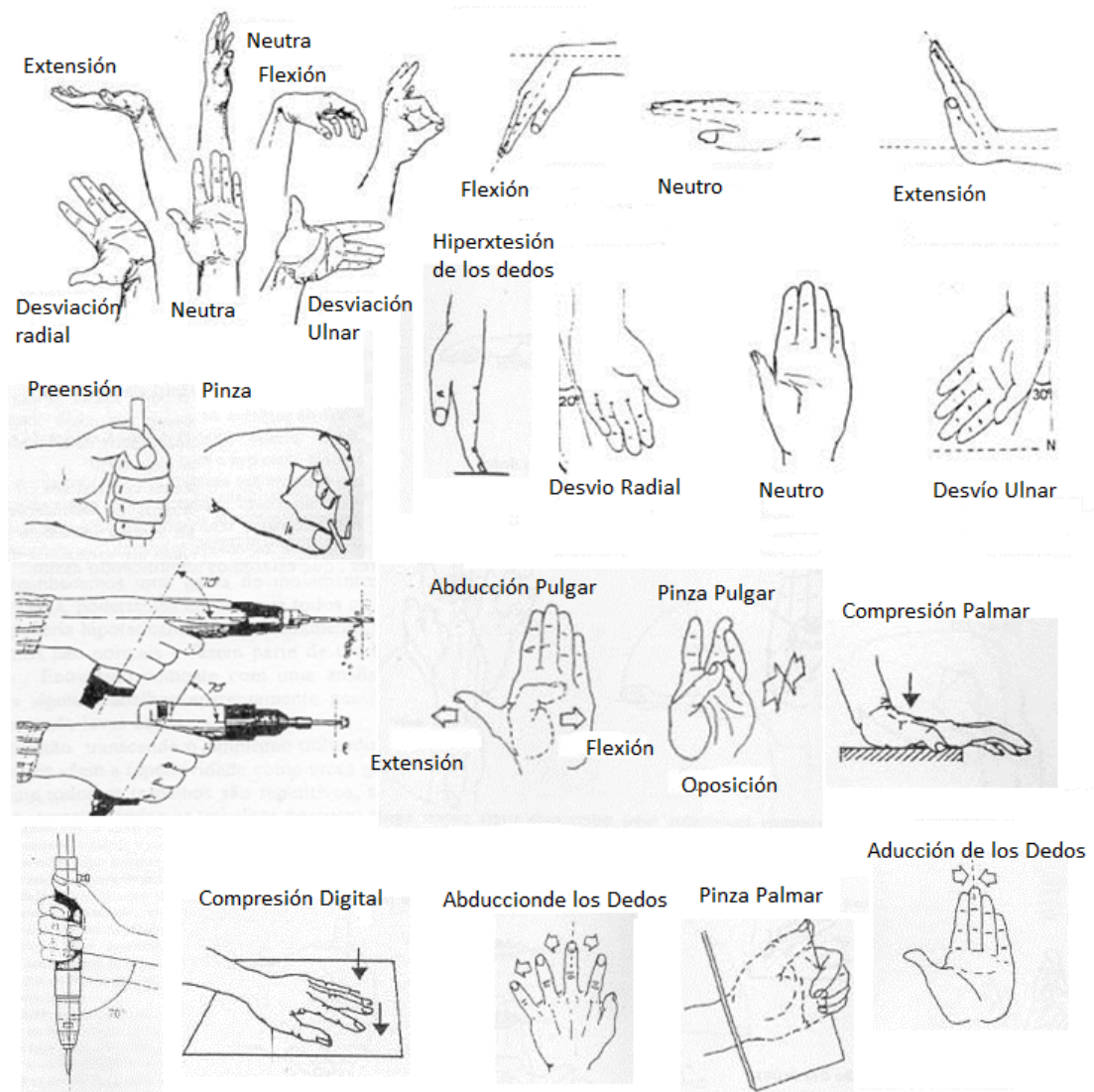


Figura 3.3: Posturas de las manos y muñecas (Modelo-Gregori-Blasco-Barrau 2001)

En las actividades prensiles es donde la mano ha permitido al hombre desarrollar la mayor habilidad. Y es que la mano es primordialmente un órgano prensil. La variada actividad prensil de la mano puede ser reducida a una *prensión de fuerza*, una *prensión de precisión* y una combinación de ambas o *presa en garra* que en algunos aspectos es diferente de las otras formas.

En una *prensión de fuerza* el objeto es asido fuertemente, bien para ser utilizado como herramienta o proyectil, bien para poder trabajar sobre él con la otra mano. Una vez hecha la presa, la mano puede mantenerse fija o ser movida como un todo por el resto del miembro gracias a la intervención de articulaciones de otros segmentos

del mismo. La destreza del movimiento es debida lógicamente a todo el miembro, no participando los dedos en la consecución de esa destreza.

En la *preensión de precisión*, no sólo la forma de sostén es más precisa, sino que los pequeños movimientos de los dedos, son esenciales para las habilidades llevadas a cabo. El objeto es cogido entre las juntas de los dedos y el pulgar, a veces por todos los dedos trifalángicos, más a menudo sólo por el pulgar y el índice, con el dedo medio frecuentemente involucrado, como cuando se utiliza un lápiz u otro utensilio pequeño. La colocación del instrumento puede ser realizada por la muñeca.

La *preensión en garra* suele ser utilizada para sostener o llevar objetos o para agarrarse y sostener el peso del cuerpo, como en la acción de trepar. En estos casos los dedos forman como unas asas alrededor del objeto en cuestión, como cuerdas, ramas, etc., pudiendo intervenir o no el pulgar en esta preensión. En conjunto se trata de una preensión para transmisión de fuerzas y no para una manipulación habilidosa.

En resumen se puede decir que por los movimientos posibles de la muñeca y de los dedos se puede realizar un repertorio relativamente reducido de las actividades manuales, pero la amplitud de los movimientos básicos y la precisión del control con que se pueden llevar a cabo, especialmente con la práctica, son adquisiciones humanas que no tienen rival.

3.1.3. Modelos prensiles humanos

La preensión humana consiste en la forma de cómo se agarra un objeto, el acto de sostenerlo y levantarlo, y el control del mismo usando la mano. Un deseo, ya sea generado interna o externamente, crea una respuesta en el ojo y en la mente. Esta manda una orden y un instante después la mano se posiciona sobre el objeto, se acomoda para agarrarlo y allí empieza con su trabajo. El diagrama de bloques de la figura 3.4 muestra en forma simplificada las relaciones incluidas entre las diferentes actividades incluidas en la preensión.

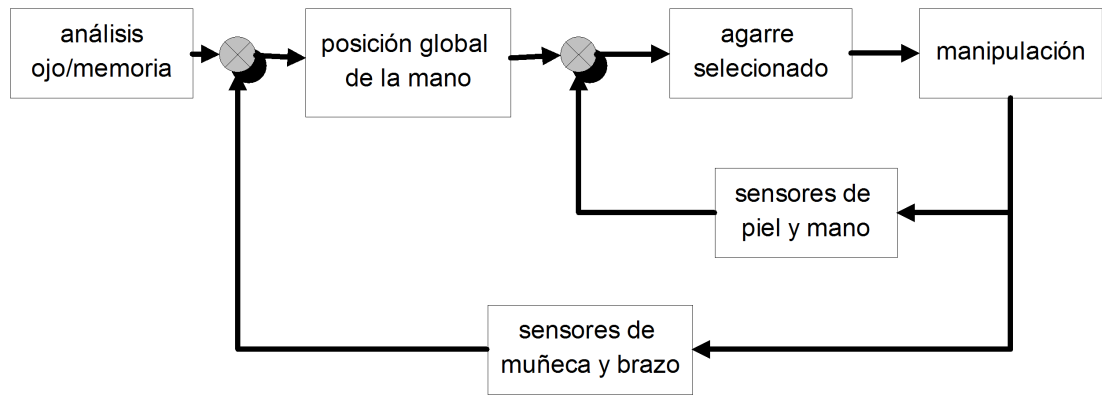


Figura 3.4: Diagrama de Bloques de las Acciones que Intervienen en la Prensaión [10]

Existen 6 patrones básicos de modelos prensiles humanos los cuales se observan en la figura 3.5

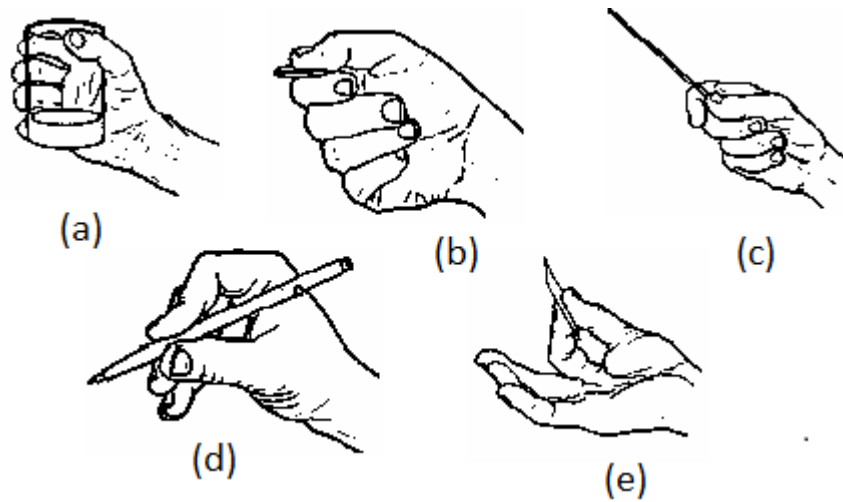


Figura 3.5: Modelos Prensiles Humanos a) Cilíndrica Palmar, b) Subterminal Lateral, c) Centralizado, d) Tridigital, e) Posición Subterminal [12]

En la figura 3.6 se muestra una clasificación taxonómica sobre las habilidades de la mano desarrollada por Schlesinger.

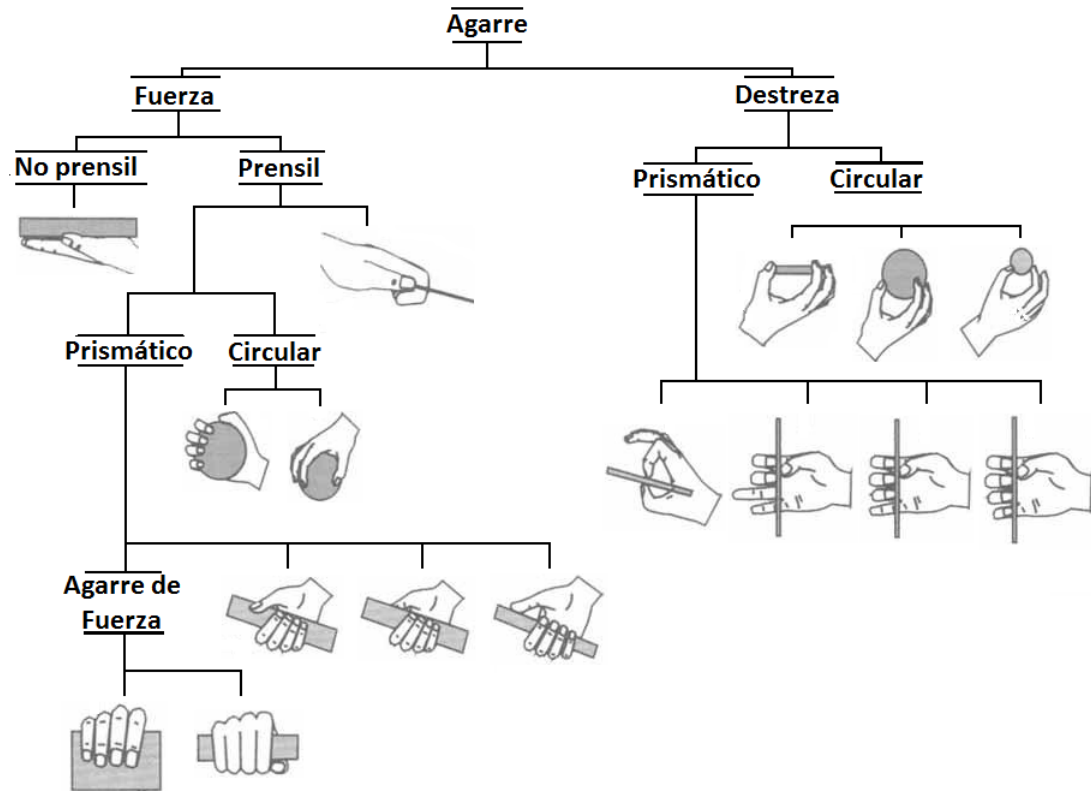


Figura 3.6: Clasificación de los Modelos Prensiles Humanos[13]

3.2. Modelo cinemático de la mano humana

Para una correcta y ágil manipulación, se debe formular como un objeto es agarrado y manipulado, y que fuerzas deben aplicarse sobre él. Este desarrollo requiere conocimiento en las relaciones geométricas del sistema manipulador-objeto, por lo tanto existen dos principales modelos cinemáticos en un sistema de manipulación multidedos, los cuales son:

- Cinemática de la mano
- Cinemática del contacto.

Para proveer una fácil descripción analítica de sus movimientos la mano humana puede ser considerada como una estructura articulada. Una estructura articulada se encuentra compuesta por segmentos rígidos conectados por articulaciones rotacionales. Un modelo jerárquico de la mano por dedo puede ser construido modelando juntas de a tres nodos (tres nodos por dedo). Cada junta (o nodo) posee su propio sistema coordinado local o su correspondiente matriz de transformación.

De esta forma, la matriz de transformación global se consigue atravesando cada división (dedo) y multiplicando las matrices de cada nodo. Se introducen restricciones en el movimiento de cada segmento rígido, porque para cualquier estructura articulada, los segmentos rígidos que se encuentran conectados poseen siempre restricciones por sus interconexiones con el sistema.

La junta o articulación inicial de la mano es el nodo de la muñeca (wrist joint) y sus juntas siguientes son los nodos MCP de los cuatro dedos y el nodo CMC del dedo pulgar (gordo). Para cada dedo el nodo PIP es el nodo hijo del nodo MCP pero es el padre del nodo DIP, esto se puede observar en la figura 3.7[10].

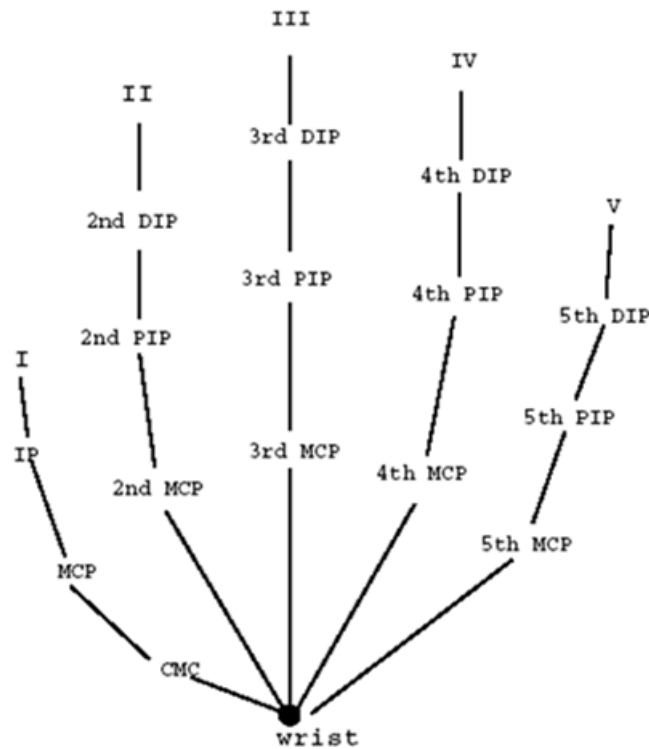


Figura 3.7: Modelo Cinemático de la Mano[10]

Con el objeto de desarrollar un plan para controlar el movimiento de cada dedo de la mano (donde cada dedo es tratado como un manipulador, o en su defecto una cadena cinemática), es necesario desarrollar técnicas para representar la posición del dedo en relación del tiempo.

Los modelos geométricos y cinemáticos del manipulador se emplean tanto para la simulación como para el control. Estos modelos se basan en el empleo de las transformaciones entre sistemas de referencia. En estos modelos se involucra esen-

cialmente el estudio de las relaciones existentes entre el espacio de las variables articulares y el espacio de trabajo, que suele ser un espacio cartesiano.

Se define cada dedo del manipulador como un todo, es decir cada dedo es independiente pero su análisis es el mismo para cada uno, además se considera cada dedo como una cadena cinemática o como un manipulador independiente de 3 grados de libertad. Cada dedo posee dos elementos básicos como articulaciones y enlaces o juntas (joints) que son estructuras rígidas que conectan las articulaciones. Cada articulación representa un grado de libertad. Se sabe que las articulaciones pueden traer consigo un movimiento lineal (articulación tipo prismática) aunque este no es nuestro caso, o un movimiento rotacional (articulaciones de rotación) entre los enlaces adyacentes.

3.2.1. Modelo Cadena Cinemática de n articulaciones

Un robot articulado puede describirse definiendo cuatro magnitudes asociadas a cada articulación. Una de estas magnitudes es la variable de la articulación y las restantes son parámetros fijos para cada robot. Estos parámetros permiten definir una representación de las relaciones de traslación y rotación entre los enlaces adyacentes. A esta relación se le denomina representación de Denavit - Hartenberg (1955).

La posición y la orientación del extremo del dedo y de cada articulación del mismo se puede describir como el producto de n transformaciones homogéneas, una por cada una de las n articulaciones del manipulador. En la figura 3.8 se observan las cuatro variables afectadas por la combinación articulación – junta.

La longitud de la junta, a_n , se define como la distancia a lo largo de la línea que es mutuamente perpendicular a los ejes de las dos articulaciones adyacentes. La torsión entre los ejes de las articulaciones en un plano perpendicular a a_n es el ángulo α_n . La distancia d_n es la distancia entre las normales a n y a $n - 1$ de los dos enlaces. Esto se refiere, a veces, como el desplazamiento de la articulación. En el caso de una articulación lineal, ésta es la variable de articulación.

El ángulo θ_n es el ángulo entre los enlaces medido como el ángulo entre las normales a n y a $n - 1$ en el plano normal al eje de la articulación. La posición y la orientación de la articulación n se define completamente con respecto a la articulación $n - 1$ por los cuatro parámetros a_n , α_n , θ_n y d_n . Se puede asignar un sistema

de referencia de coordenadas a cada enlace para establecer las relaciones entre sucesivos enlaces utilizando las transformaciones para cada una de las cuatro variables.

El origen del sistema de referencia de coordenadas para el enlace n se incluye en la intersección del eje, z_n , para la articulación $n - 1$ y la normal común, a_n , desde la articulación n a la articulación $n + 1$. Es decir, el sistema de referencia se incorpora, en el extremo de un enlace, en la articulación sucesiva. Esto se realiza utilizando: Una rotación de ángulo θ_n alrededor del eje z_{n-1} . Una traslación de distancia d_n a lo largo de z_{n-1} . Una traslación a lo largo de x_n en una longitud a_n . Una rotación de ángulo α_n alrededor de x_n .

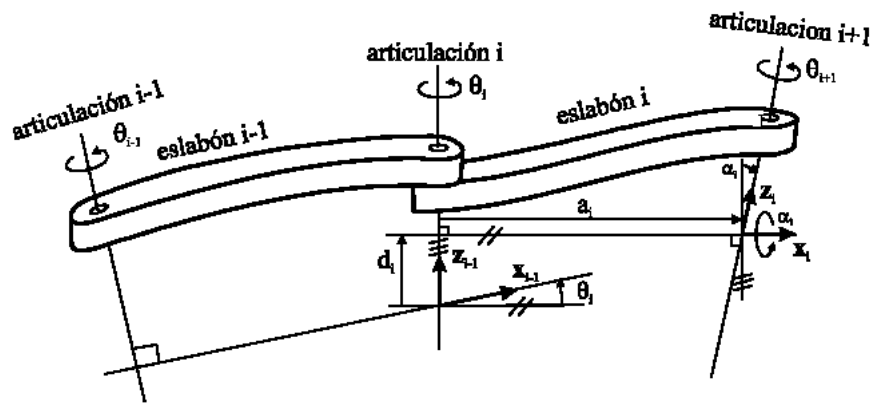


Figura 3.8: Variables Afectadas por la Combinación Articulación – Juntura[10]

Utilizando las transformaciones definidas con anterioridad para la rotación y la traslación, se puede entonces desarrollar la transformación que relaciona el sistema de referencia de coordenadas del enlace n con el enlace $n - 1$. Por ejemplo, la transformación homogénea A que describe la posición y orientación del sistema de referencia de coordenadas del primer enlace se escribe:

$$A_n = Rot(z_1, \Theta_2) Trans(0, 0, d_2) Trans(a_2, 0, 0) Rot(x_2, \alpha_2) \quad (3.1)$$

Al componer estas transformaciones aplicando las matrices de transformación elementales para las rotaciones y las traslaciones, se obtiene la siguiente forma general asociada a la articulación. Para una articulación rotacional, esta transformación se reduce a:

$$A_n = \begin{bmatrix} \cos\theta & -\sin\theta\cos\alpha & \sin\theta\sin\alpha & a\cos\theta \\ \sin\theta & \cos\theta\cos\alpha & -\cos\theta\sin\alpha & a\sin\theta \\ 0 & \sin\alpha & \cos\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

Para una articulación lineal, el valor de a es 0 ($a = 0$). Esta matriz es conocida como la matriz de Denavit – Hartenberg, o matriz D-H. Para construir el modelo directo de un robot con n articulaciones es necesario definir un sistema de referencia solidario a cada segmento y elegir sus parámetros.

A continuación puede procederse a la obtención de matrices de transformación de cada articulación, y a partir de las matrices de cada articulación puede hallarse la transformación compuesta que relaciona la localización (n) con la (0). Es decir, la descripción de un enlace con respecto al sistema de referencia del robot (espacio universal) se puede realizar como un sistema de referencia de coordenadas de enlace, T_n , donde:

$$T_n = A_1A_2A_3A_4A_5\dots A_n \quad (3.3)$$

Para concluir, con el objeto de calcular la matriz T , se deben encontrar primero las n matrices A , y este proceso se realiza mediante la sustitución de los valores a_n , α_n , θ_n y d_n en la matriz D-H.

Se Considera el modelo del dedo robot ilustrado en la figura 3.9, se debe empezar entonces por definir los sistemas coordenados en cada articulación para así luego encontrar los parámetros estructurales.

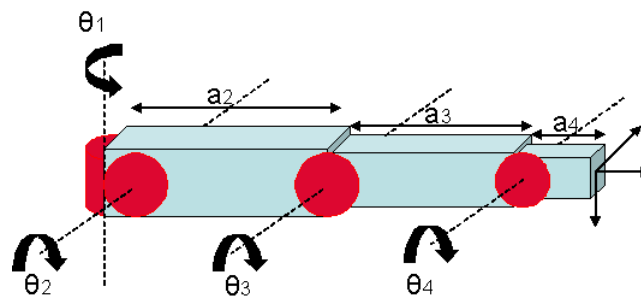


Figura 3.9: Modelo esquemático dedo robot[10]

El procedimiento entonces es el siguiente:

1. Enumere las articulaciones, donde la primera es la más cercana a la base.
2. El sistema de coordenadas i , mide el movimiento de las articulaciones $i + 1$. (Articulaciones; $i = 1 \dots n$).
3. El sistema de coordenadas de la base (X_0, Y_0, Z_0), donde Z_0 va alineado con el eje de movimiento articulación 1.
4. Z_i alineado con eje de movimiento articulación $i + 1$.
5. Origen O_i es la intersección entre Z_{i-1} y Z_i . Si son paralelos, se toma la intersección de la normal común entre Z_{i-1} , Z_i y Z_i .
6. X_i , regla de la mano derecha Z_{i-1} y Z_i . Si no se interceptan, normal común entre Z_{i-1} y Z_i .
7. Y_i completa el sistema de coordenadas en cada articulación.

Ahora bien al ubicar en la figura 3.10 del modelo cinemático los sistemas coordenados, y los orígenes se pueden encontrar los parámetros estructurales.

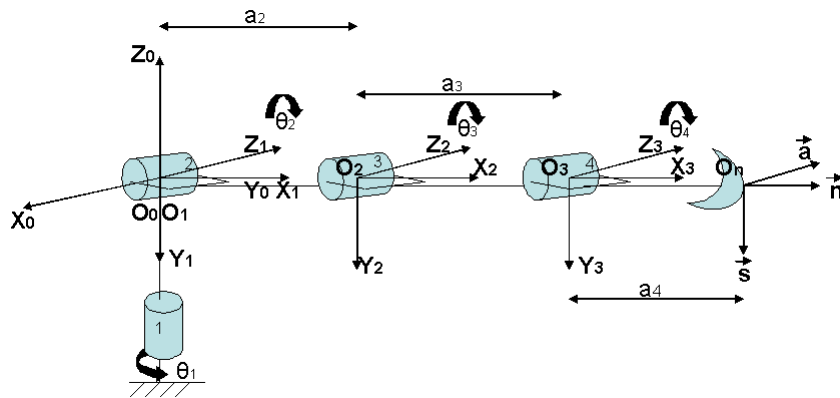


Figura 3.10: Modelo cinemático dedo robot, con sus respectivos ejes coordenados y orígenes[10]

Link	a_i	d_i	α_i	θ_i
1	L_1	0	0	θ_1
2	L_2	0	0	θ_2
3	L_3	0	0	θ_3

Table 3.3: matriz D-H

$$A_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

$$A_3^0 = A_1^0 \cdot A_2^1 \cdot A_3^2 \quad (3.5)$$

$$A_1^0 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & L_1\cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & 0 & L_1\sin\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

$$A_1^0 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & L_1\cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & 0 & L_1\sin\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.7)$$

$$A_2^1 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & L_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & L_2\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.8)$$

$$A_3^2 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & L_3\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & L_3\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.9)$$

$$A_2^0 = A_1^0 * A_2^1 \quad (3.10)$$

$$A_2^0 = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) & 0 & L_2 \cos(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 & L_3 \sin(\theta_1 + \theta_2) + L_1 \sin\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.11)$$

$$A_3^0 = A_2^0 * A_3^2 \quad (3.12)$$

$$A_3^0 = \begin{bmatrix} \cos(\theta_1 + \theta_2 + \theta_3) & -\sin(\theta_1 + \theta_2 + \theta_3) & 0 & L_3 \cos(\theta_1 + \theta_2 + \theta_3) + L_2 \cos(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2 + \theta_3) & \cos(\theta_1 + \theta_2 + \theta_3) & 0 & L_3 \sin(\theta_1 + \theta_2 + \theta_3) + L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.13)$$

Capítulo 4

Reconocimiento de Formas

Es el estudio de como las máquinas observan el medio ambiente, aprenden a distinguir formas interesantes de lo que observan y llegan a conclusiones acerca de las categorías o clases de las formas. En 40 años de investigación todavía no se ha podido diseñar un reconocedor de patrones para máquinas de uso general.

El mejor reconocedor de patrones es el humano, pero aún so se sabe completamente como un humano reconoce formas. Mientras mas formas importantes tenga a su disposición un humano mejor será la decision que tome.

Se pueden enseñar a las computadoras a reconocer formas. Se pueden escribir programas de computadoras para ayudar a un banco a decidir si dar o no dar un préstamo, para ayudar a un doctor a diagnosticar una enfermedad y para ayudar a un piloto a aterrizar su avión.

El reconocimiento de formas puede ser:

- **Reconocimiento de formas Supervisado (análisis discriminante):**

La forma de entrada es identificada como un miembro de una clase predefinida. Por ejemplo, se sabe de antemano que una persona tiene un tumor canceroso de acuerdo a la intensidad de color de una imagen de resonancia magnética que se le toma.

- **Reconocimiento de formas No supervisado (análisis de conglomerados):**

La forma de entrada es asignada a una clase que aún no es conocida. Las clases son estimadas basadas en la similaridad de las formas.

4.1. Etapas de un problema de reconocimiento de formas

1. Formulación del problema
2. Recolección de datos
3. Análisis de datos
4. Selección y extracción de variables
5. Reconocimiento de formas no supervisado (conglomerados)
6. Reconocimiento de formas supervisado (discriminación)
7. Evaluación de resultados
8. Interpretación

La percepción de objetos por un sistema es el objetivo declarado de la disciplina del Reconocimiento de Formas, que se engloba en un conjunto de técnicas mucho más ambiciosas como lo son la Inteligencia Artificial. El reconocimiento de formas, sin embargo, ha ido adquiriendo entidad de por sí mismo, hasta el punto de que en la actualidad constituye un campo de investigación que evoluciona con dinámica propia, a menudo independiente de la inteligencia artificial.

Reconocer un objeto consiste en asociarle o identificarlo con un mensaje semántico, es decir un significado. En el caso más general, este mensaje semántico es simplemente un punto (también llamado forma) de un universo semántico (a veces llamado universo interno, en oposición al universo externo captado por los sensores). Normalmente, a una forma (p.e.: silla) le corresponden muchos posibles objetos (p.e.: todas las posibles sillas). Dicho de otro modo, una forma es un conjunto de objetos que se caracterizan por estar etiquetados por un mismo mensaje semántico.

Un sistema de reconocimiento de formas, en su versión más simple, estará constituido de la manera esquematizada en la figura 000000, es decir, por dos módulos:

- Un módulo de representación, que obtiene, a partir de cada objeto, (usualmente) captado del mundo real mediante una serie de sensores, una representación conveniente para su utilización por el módulo de interpretación.

- Un módulo de interpretación, que proporciona, a partir del conjunto de formas, también representado convenientemente, y de la representación del objeto proporcionada por el módulo de representación, la forma a la que pertenece el objeto. Lleva a cabo pues un proceso de comparación ("pattern matching") entre el objeto y el conjunto de formas.

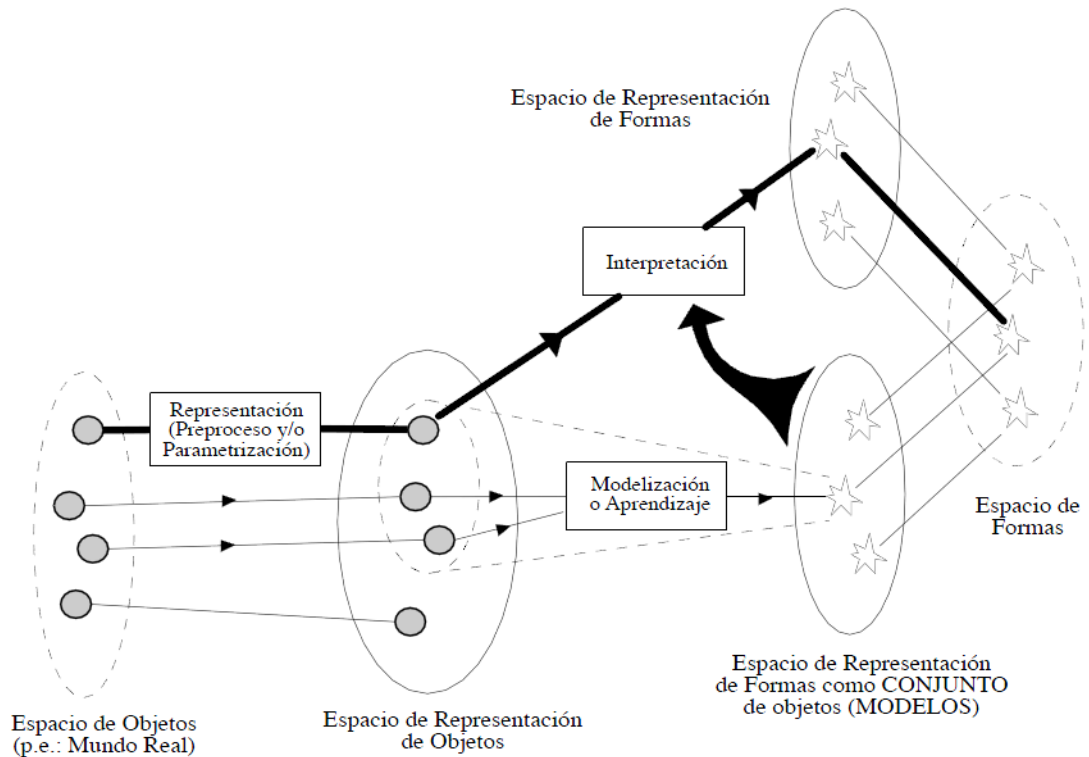


Figura 4.1: Etapas de un Problema de Reconocimiento de Formas [16]

El módulo de representación puede no ser necesario, si se da la circunstancia de que los objetos ya vienen dados de manera conveniente para el módulo de interpretación. En muchos otros casos (como los presentados en la parte experimental de este trabajo) el módulo de representación está Compuesto por varias etapas, algunas de las cuales afectan un pre procesó y otras una parametrización:

- Las etapas de pre proceso efectúan en general transformaciones que, o bien no cambian el dominio de representación, o bien llevan a subconjuntos de éste (p.e.: filtrado de la señal, supresión de grises por umbral...).
- Las etapas de parametrización cambian el dominio de representación y a menudo reducen drásticamente la cantidad de información, suprimiendo aquella que resulta redundante y/o inútil. En este último caso, se puede considerar

la operación como equivalente a extraer las características más significativas del objeto, para representarlo por un conjunto de parámetros o descriptores (p.e.: transformada discreta de Fourier, cadena de símbolos,...).

El conjunto de formas utilizado por el módulo de interpretación representa el conocimiento que tiene el sistema de su entorno. Muy a menudo la representación del mismo es simplemente un conjunto de representaciones de formas (p.e.: un conjunto de gramáticas), pero puede estar representado por una estructura rica y compleja (p.e.: una rica red o gramática). El proceso de obtención de este conjunto de formas se conoce como modelización o aprendizaje del entorno (de sus objetos) a reconocer y es efectuado por el módulo de aprendizaje.

4.2. Métodos para Reconocimiento de Formas

1. Template matching (“pareo de plantillas”). La forma a ser reconocida es comparada con un conjunto de plantillas (formas en dos dimensiones), tomando en cuenta que pueden haber habido traslaciones, rotaciones o cambio de escala.
2. Método estadístico. Aquí cada forma es representada como un conjunto de mediciones de p características y puede considerarse como un punto en el espacio p dimensional. Primero hay que elegir aquellas características que permiten a los vectores de formas de distintas clases ocupar regiones disjuntas (separabilidad) en el espacio p -dimensional. Dado un conjunto de formas el objetivo es establecer fronteras en el espacio de características que separen las formas que pertenecen a distintas clases. Hay dos metodologías: la basada en teoría de decisión y la basada en análisis discriminante. En la primera las fronteras son determinadas por las distribuciones de probabilidad de las formas que pertenecen a cada clase. En el Segundo caso se especifica una forma paramétrica (lineal, cuadrática, etc) de la frontera y luego se encuentra la mejor frontera basándose en la clasificación de las formas disponibles en la muestra.
3. Syntactic or structural matching (“pareo estructural o sintáctico”). En este caso una forma es vista como una composición de subformas simples que a su vez son construidas de subformas más simples. Las subformas más simples a ser

reconocidas son llamadas primitivos y una forma compleja es representada en términos de interrelaciones entre estos primitivos. Este método tiene analogía con la sintaxis de un lenguaje. Hay problemas para separar en sus primitivos una forma que tiene mucho ruido. Requiere bastante esfuerzo computacional. Este método ha sido aplicados en clasificación de imagines de textura y de ondas acústicas.

4. Redes Neuronales. Consiste de una red en los cuales los nodos son neuronas artificiales que se conectan mediante enlaces que tienen distintas ponderaciones. Las redes neuronales tienen la habilidad de aprender complejas relaciones no lineales de input-output usando procedimientos secuenciales de entrenamiento y además se adaptan ellas mismas a los datos. Las redes neuronales más usadas para efectos de clasificación supervisada son: Feed-forward networks (redes de alimentación hacia adelante) que incluye a MLP (Multilayer perceptron) y las redes de funciones de base radial (RBF). Para clasificación no supervisada se usa las redes de Kohonen conocidas como Self-Organizing Maps (SOM).

4.3. Postulados De Niemann para el Reconocimiento de Formas

Cualquier aproximación al diseño de sistemas de reconocimiento de formas, se debe plantear teniendo en cuenta lo posible, los siguientes postulados:

1. Representatividad de el diseño de un proceso de Reconocimiento de Formas en un campo dado requiere de una muestra representativa de formas.
2. Discriminabilidad. Una forma simple tiene siempre características que permiten determinar su grado de pertenencia a una clase, esto es, siempre es posible encontrar un conjunto de características que discriminen una forma simple en algún tipo de espacio.
3. Compacidad y Separación. Las características de las formas de una clase ocupan un dominio compacto de la representación, y los dominios ocupados por clases diferentes están separados. El que se verifique este postulado es condición necesaria para que pueda tener lugar un proceso de RF. En la práctica resulta de extrema importancia la elección de características que lo cumplan[7]

4.4. Aplicaciones de Reconocimiento de formas

Las principales aplicaciones del reconocimiento de formas se muestran en el siguiente cuadro:

Area	Aplicación	Formas de entrada	Clases de las Formas
Bioinformática	Biología Molecular	Secuencia de DNA o proteínas	Tumor maligno o benigno
Data Mining	Busqueda de importantes formas	Mediciones de varias características	Conglomerados bien separados
Reconocimiento de Personas	Reconocimiento de caras, habla, huellas, de escritura.	Imágenes digitales, ondas acústicas	Usuarios autorizados para entrar
Remote Sensing	Reconocimiento de suelos	Imágenes multiespectrales	Tipos de terrenos, tipos de cultivos
Medicina	Diagnóstico de enfermedades	Mediciones de temp, presión, edad, peso, etc	Persona sana o enferma

Cuadro 4.1: Aplicaciones de Reconocimiento de formas

4.5. Las Redes Neuronales Artificiales (ANNs)

Las Redes Neuronales Artificiales son una simulación abstracta de un sistema nervioso real que está formado por un conjunto de unidades neuronales conectadas unas con otras por medio de conexiones del tipo axón. Estas conexiones tienen una gran semejanza con las dendritas y los axones en los sistemas nerviosos biológicos.

Los modelos de Redes Neuronales Artificiales se pueden clasificar como:

- Modelos de tipo Biológico:

Redes que tratan de simular los sistemas neuronales biológicos así como las funciones auditivas o algunas funciones básicas de la visión

- Modelos dirigidos a Aplicaciones:

Modelos menos dependientes de los sistemas biológicos. Modelos en donde sus arquitecturas están fuertemente ligadas a las necesidades de las aplicaciones

Una Red Neuronal Artificial consiste en un conjunto de elementos de procesamiento simples conectados entre sí y entre los que se envían información a través de las conexiones.

Conjunto de Unidades de procesamiento (neuronas).

Conexiones entre unidades (Asociando a cada conexión un peso o valor)

Funciones de Salida y Activación para cada unidad de procesamiento.

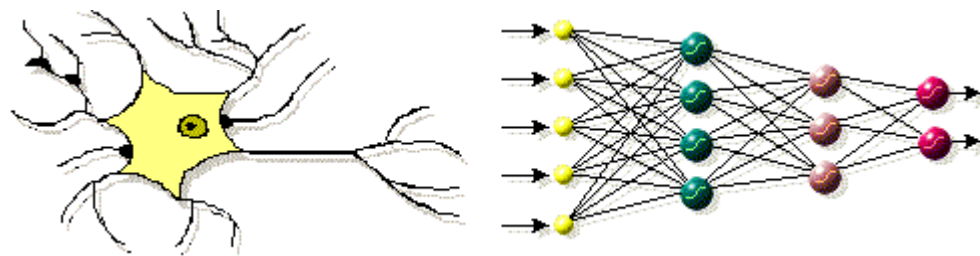


Figura 4.2: Modelo general de una Red Neuronal Artificial

4.5.1. Modelo de Neurona Artificial

El modelo de Neurona y la arquitectura de una Red Neuronal, describen como la Red transforma sus entradas en las salidas. Todo esto, puede ser visto simplemente como una "Computación".

Los elementos individuales de cálculo que forman la mayoría de los modelos de sistemas neuronales artificiales, reciben el nombre de Elemento de Procesamiento o Neurona Artificial.

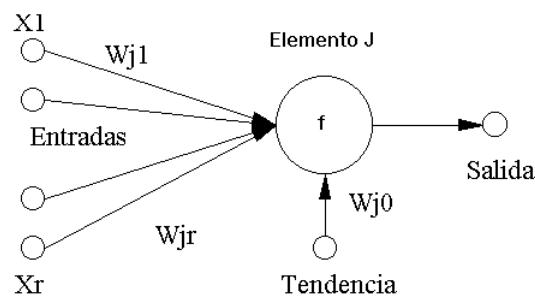


Figura 4.3: Arquitectura de una Red Neuronal

- Entradas

Cada EP puede tener múltiples entradas, asociadas a propiedades diferentes.

Opcional: Entrada de Tendencia o Bias (valor constante).

- Conexiones (Pesos)

Cada entrada tiene asociada un Peso o Intensidad de Conexión. W_{ji} : conexión entre la entrada i y el EP j .

Resaltan la importancia de la entrada al PE.

- Función de Activación

Cada EP determina un valor de entrada neto (basándose en las entradas y en las fuerzas de conexión asociadas a cada una de ellas).

$$E_j.neta_i = \sum_{j=0} (x_j * w_{ij})$$

Con el valor de entrada neta, se calcula el valor de activación del EP: donde $(t - 1)$ representa el paso temporal anterior al instante t .

- Función de Salida

Una vez calculada la activación de un EP, se determina el valor de salida. Este se obtiene aplicando la función de salida (Transferencia) sobre la activación del EP. $y_i = f_i(a_i)$.

La función de transferencia actúa como un Limitador de Rango Dinámico.

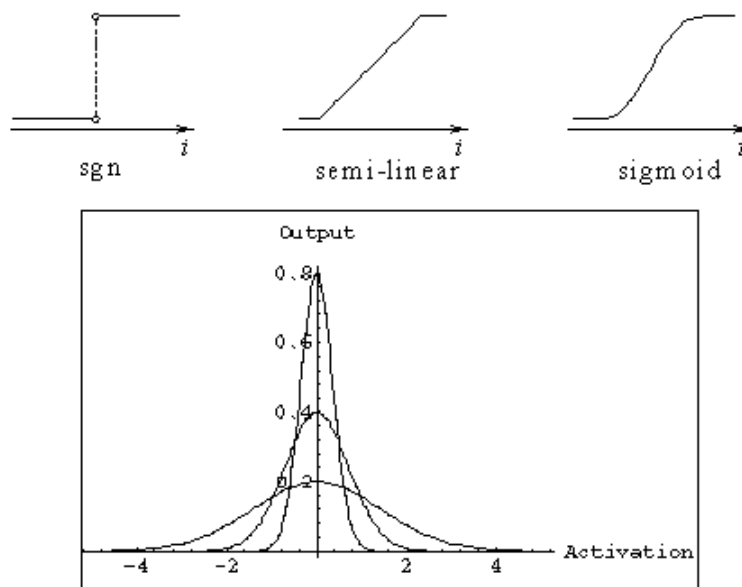


Figura 4.4: Ejemplos de Funciones de Salida o Transferencia.

4.5.2. Aprendizaje

El aprendizaje es un proceso por el cual los parámetros libres de una red neuronal son ajustados a través de un proceso continuo de estimulación por parte del entorno en donde se sitúa el sistema.

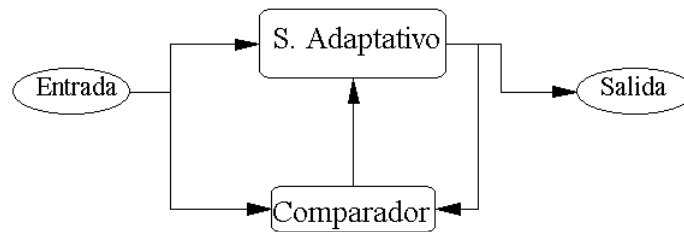


Figura 4.5: Estructura General de un Sistema de Aprendizaje.

El tipo de aprendizaje es determinado por la forma o manera que tienen lugar dichos cambios. Esta definición implica la siguiente secuencia de hechos:

- La red neuronal se encuentra estimulada por el entorno.
- La red neuronal cambia como consecuencia de dicho estímulo.
- La red neuronal responde de manera diferente al entorno a causa de los cambios que se han producido en su estructura interna.
- La ecuación que especifica como cambian los pesos, recibe el nombre de Ley de Aprendizaje.

El aprendizaje se puede clasificar en dos categorías generales:

- Aprendizaje Supervisado

Necesita un profesor que mida el funcionamiento del sistema.

- Aprendizaje no Supervisado

No se necesita profesor. El sistema debe organizarse a si mismo y por si sólo.

4.5.3. Tipos de Aprendizaje

- Aprendizaje por Corrección de Error.
- Aprendizaje por Refuerzo.
- Aprendizaje Estocástico.

4.5.3.1. Aprendizaje por Corrección de Error.

El entrenamiento consiste en presentar al sistema un conjunto de pares de datos, representado la entrada y la salida deseada para dicha entrada. Este conjunto recibe el nombre de conjunto de entrenamiento. Trata de minimizar el Error entre la Salida Deseada y la Actual. Aprendizaje OFF Line.

Método:

1. Inicializar aleatoriamente los pesos
2. Presentación del conjunto de entrenamiento (CE)
3. Obtención de las salidas para el CE
4. Comparación de salidas deseadas con actuales.
5. Si se verifica el criterio de finalización ir al siguiente paso, sino ir al paso 2.
6. Fín.

4.5.3.2. Aprendizaje por Refuerzo

Aprendizaje más lento que el anterior. No se dispone de un ejemplo completo del comportamiento deseado. No se conoce la salida deseada exacta para cada entrada. Se conoce como debería de ser el comportamiento de manera general ante diferentes entradas. Es un aprendizaje ON Line. Relación de entrada-salida a través de un proceso de éxito o fracaso, produciendo una señal (Señal de Refuerzo) que mide el buen funcionamiento del sistema.

Esta seña "señal de refuerzo" está caracterizada por el hecho de que es menos informativa que en el caso de aprendizaje supervisado mediante ejemplos. Barto, Sutton y Anderson han formulado el "aprendizaje por Refuerzo" como una estrategia de

aprendizaje en donde no se necesita un conjunto de ejemplos evaluados por un profesor. El sistema descrito por Barto explora el espacio entrada-salida y usa una señal de refuerzo (feedback) sobre las consecuencias de la señal de control (salida de la red) sobre el entorno.

Los pesos se ajustan en base a la señal de refuerzo basándose en un mecanismo de probabilidades. "Si una acción tomada por el sistema de aprendizaje es seguida por un estado satisfactorio, entonces la tendencia del sistema a producir esa particular acción es reforzada. En otro caso, la tendencia del sistema a producir dicha acción es disminuida". La función del supervisor es más la de un crítico que la de un maestro.

4.5.3.3. Aprendizaje Estocástico

Este tipo de aprendizaje consiste básicamente en realizar cambios aleatorios en los valores de los pesos y evaluar su efecto a partir del objetivo deseado y de distribuciones de probabilidad.

Método:

1. Se realiza un cambio aleatorio en los Pesos.
2. Se determina la nueva energía de la red, si la energía no decrece: se aceptaría el cambio en función de una determinada y preestablecida distribución de probabilidades.

4.5.4. Perceptrón

El perceptron es la base de la mayor parte de las arquitectura de las RNA que se interconectan entre sí. Las neuronas emplean funciones de activación diferentes según la aplicación, algunas veces son funciones lineales, otras funciones sigmoidales (p.ej. la tanh), y otras funciones de umbral de disparo. La eficiencia sináptica se representa por factores de peso de interconexión w_{ij} , desde la neurona i , hasta la neurona j .

Los pesos pueden ser positivos (excitación) o negativos (inhibición). Los pesos junto con las funciones $f(z)$ dictan la operación de la red neuronal. Normalmente las funciones no se modifican de tal forma que el estado de la red neuronal depende del valor de los factores de peso (sinápsis) que se aplica a los estímulos de la neurona.

En un perceptrón, cada entrada es multiplicada por el peso W correspondiente, y los resultados son sumados, siendo evaluados contra el valor de umbral, si el resultado es mayor al mismo, el perceptrón se activa.

4.5.5. Adaline

El Adaline fue la primera red neural lineal (Widrof & Ho, 1960). El nombre significa ADAPtive LINear Element, y fue desarrollada como un circuito electronico adaptivo para uso en telefona mucho antes de que se concibiera el termino “red neural”. Se entrena por medio de un proceso de minimizacion de errores que garantiza la convergencia a una solucion. La funcion de error mas frecuentemente usada es el error cuadratico.

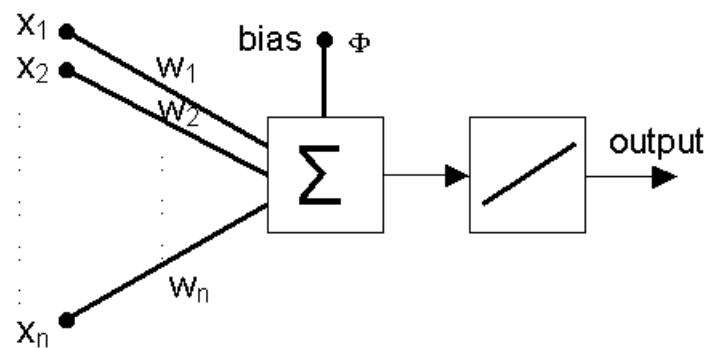


Figura 4.6: Adaline.

El ADALINE, al igual que el Perceptron, se entrena por medio de la Regla Delta:

- Inicializar los w_i en valores aleatorios pequeños.
- Repetir
 - Inicializar el error E en cero.
 - Para cada w_i inicializar Δw_i en cero.
 - Para cada $(\vec{x}; y) \in D$:
 - Computar $\delta = y - o(\vec{x})$
 - Actualizar $E \leftarrow E + \delta^2$
 - Para cada w_i actualizar $w_i \leftarrow \Delta w_i + \eta \delta x_i$

- Para cada w_i actualizar $w_i \leftarrow w_i + \Delta w_i$
- hasta que el error E sea suficientemente pequeño.

El algoritmo anterior actualiza los w_i al final de cada época en vez de continuamente con cada presentación de los vectores de entrenamiento. A esta forma de actualizar los pesos se le denomina actualización en lotes (batch update). Cuando los pesos se actualizan con cada presentación estamos realizando actualización continua (continuous update). Ambos métodos funcionan en la práctica, pero solo el primero puede demostrarse matemáticamente que garantiza la convergencia del algoritmo. La actualización continua puede presentar inestabilidades a menos que se usen valores de muy pequeños (que causan una convergencia más lenta). Sin embargo, la actualización continua es algo menos propensa a quedar atrapada en mínimos locales de la función de error.

4.5.6. Funciones de Base Radial

Son funciones cuya salida depende de la distancia a un punto denominado Centro.

Características:

- Simétricas respecto $x = 0$
- Se definen con al menos dos parámetros:
 - Centro: Punto donde la función posee un extremo.
 - Anchura: Magnitud de la variación de la función según se aleja del centro.

Función de Gauss 1D - 2D

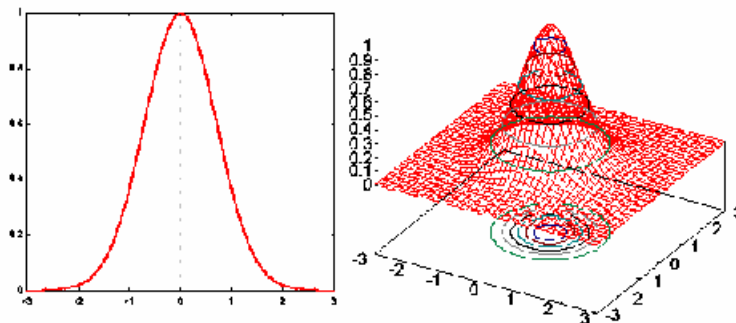


Figura 4.7: RBF

Las funciones de base radial tienen todas ellas un carácter Local pues son funciones que alcanzan un nivel cercano al máximo de su recorrido cuando el patrón de entrada $X(n)$ está próximo al centro de la neurona. A medida que el patrón se aleja del centro, el valor de la función va tendiendo al valor mínimo de su recorrido.

Las salidas de las redes de neuronas de base radial son, por tanto una combinación lineal de gaussianas, cada una de las cuales se activa para una determinada porción del espacio definido por los patrones de entrada.

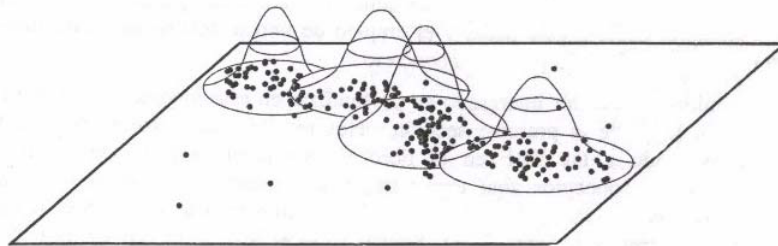


Figura 4.8: Representación de Agrupamiento

4.5.7. Backpropagation

Backpropagation es un tipo de red de aprendizaje supervisado, que emplea un ciclo de propagación (adaptación de dos fases). Una vez que se ha aplicado un patrón a la entrada de la red como estímulo, éste se propaga desde la primera capa a través de las capas superiores de la red, hasta generar una salida. La señal de salida se compara con la salida deseada y se calcula una señal de error para cada una de las salidas.

Las salidas de error se propagan hacia atrás, partiendo de la capa de salida, hacia todas las neuronas de la capa oculta que contribuyen directamente a la salida. Sin embargo las neuronas de la capa oculta sólo reciben una fracción de la señal total del error, basándose aproximadamente en la contribución relativa que haya aportado cada neurona a la salida original. Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido una señal de error que describa su contribución relativa al error total. Basándose en la señal de error percibida, se actualizan los pesos de conexión de cada neurona, para hacer que la red converja hacia un estado que permita clasificar correctamente todos los patrones de entrenamiento.

La importancia de este proceso consiste en que, a medida que se entrena la red, las neuronas de las capas intermedias se organizan a sí mismas de tal modo que las distintas neuronas aprenden a reconocer distintas características del espacio total de entrada.[11]

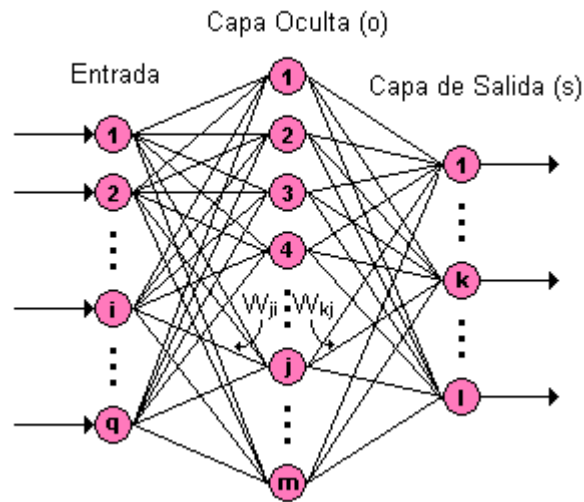


Figura 4.9: Red Backpropagation[11]

Después del entrenamiento, cuando se les presente un patrón arbitrario de entrada que contenga ruido o que esté incompleto, las neuronas de la capa oculta de la red responderán con una salida activa si la nueva entrada contiene un patrón que se asemeje a aquella característica que las neuronas individuales hayan aprendido a reconocer durante su entrenamiento.

Capítulo 5

Validación Experimental

5.1. Diseño e Implementación del Programa de identificación de patrones.

Inicialmente se planteó un programa para la extracción de los parámetros de cada una de las figuras a analizar en Matlab®, con el cual se generaron una serie de tablas donde se almacenan los parámetros con los cuales se entrenó la red neuronal para que realizara un reconocimiento óptimo de la formas.

El diagrama de flujo del programa de generación de datos de entrenamiento es el mostrado en la figura5.1

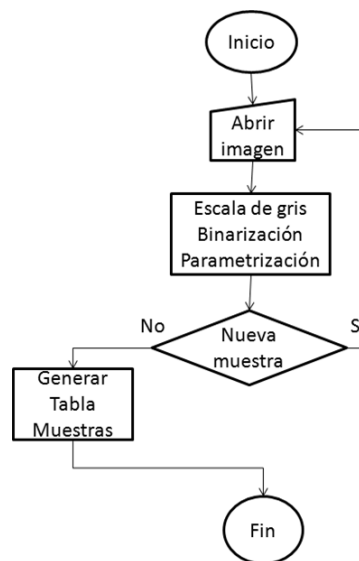


Figura 5.1: Diagrama de Flujo Para el Programa de Generación de Datos.

A continuación en las figuras 5.2, 5.3, 5.4 y 5.5 se muestran la interfaz del programa y las base de datos de figuras que se utilizaron para obtener las tablas con las cuales se entreno a la red neuronal.

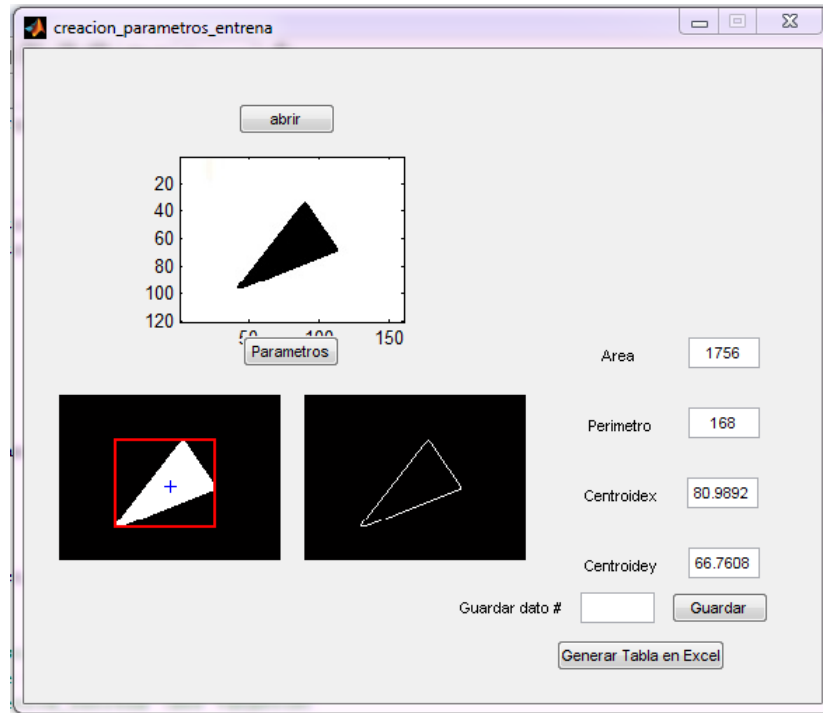


Figura 5.2: Interfaz Prog. generar datos.

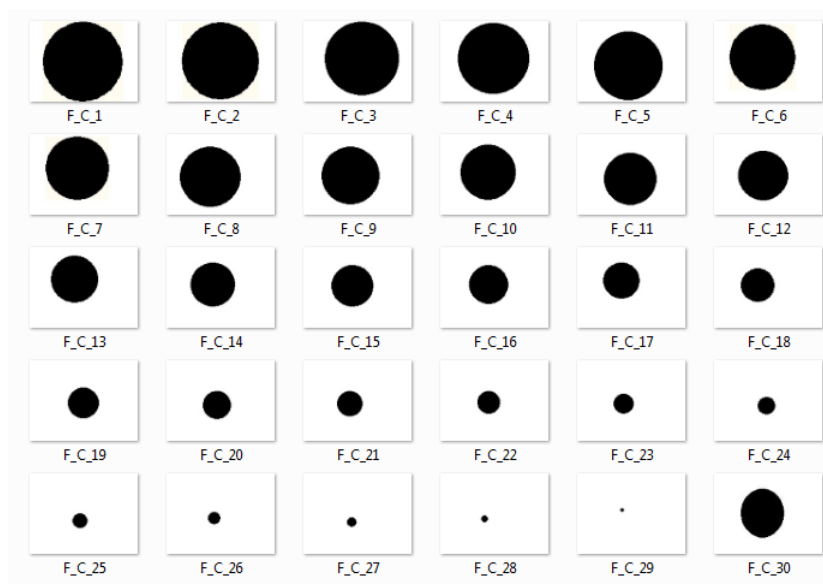


Figura 5.3: Base de Figuras #1

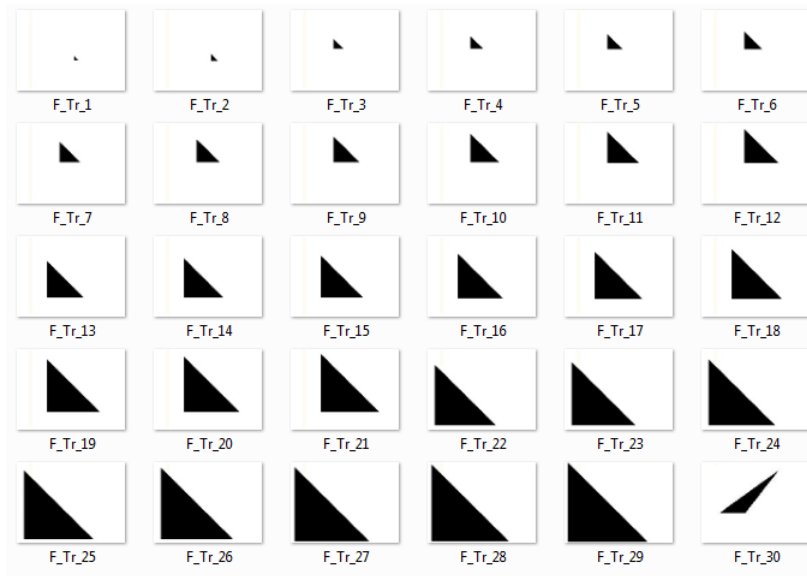


Figura 5.4: Base de Figuras #2

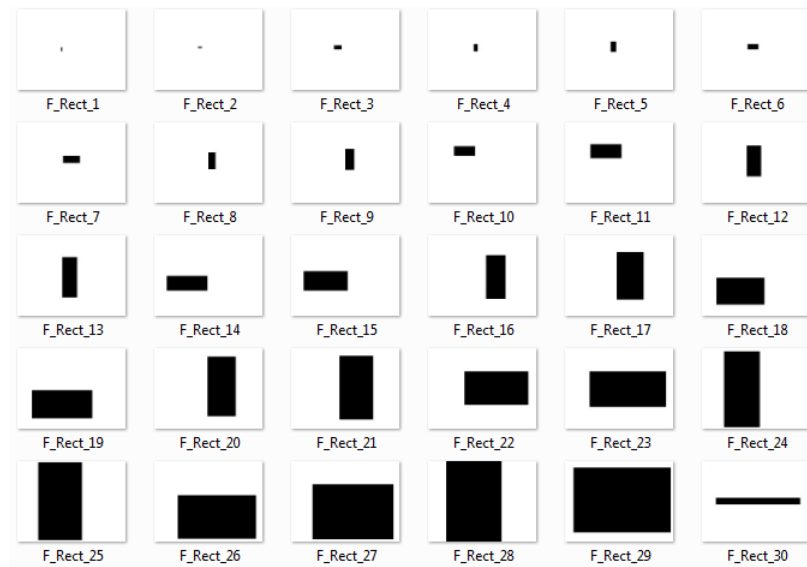


Figura 5.5: Base de Figuras #3

Para la extracción de parámetros de cada una de las figuras inicialmente se captura la imagen o se abre alguna que se tenga en el disco y que sea de 120 x 160 de tamaño, ya que la matriz en donde se almacena la imagen es de estas dimensiones. A continuación esta imagen se pasa a escala de grises, este proceso sustrae toda la información de color que contiene cada pixel y nos deja una separación de 255 niveles entre el blanco y el negro.

Luego se procede a calcular un valor de umbral, para poder convertir la imagen a blanco y negro. El umbral es un valor límite (o bien un intervalo) a partir del cual todos los valores de intensidades mayores serán codificados como 1 mientras que los que estén por debajo serán codificados a 0. Para este proceso se utiliza la función graythresh de matlab, la cual permite obtener un valor óptimo para el umbral, necesario en la binarización; que es el paso que sigue. Posteriormente se procede a calcular los parámetros de área, perímetro, centroide en X y en Y.

A continuación se muestran las tablas obtenidas con los parámetros de cada una de las figuras.

#	A	B	C	D
1	Area	Perimetro	Centroide x	Centroide en Y
2	10944	336	0,5000263	0,5009875
3	10227	328	0,4998975	0,5010084
4	9480	308	0,5000527	0,5000527
5	8794	296	0,5000783	0,5000783
6	8144	284	0,5	0,5
7	7473	279	0,4997836	0,5012788
8	6936	267	0,5002055	0,5001112
9	6328	252	0,5	0,5
10	5774	240	0,5001168	0,5001168
11	5252	228	0,5	0,5
12	4748	216	0,5	0,5
13	4270	208	0,4999019	0,4999019
14	3822	196	0,5000486	0,5000486
15	3388	184	0,499356	0,499356
16	3002	172	0,4995756	0,4995756
17	2622	160	0,4994279	0,4994279
18	2270	149	0,4988416	0,4988416
19	1942	140	0,4989907	0,4989907
20	1656	128	0,5	0,5
21	1382	116	0,4994142	0,4994142
22	1124	104	0,5	0,5
23	902	92	0,4998044	0,4998044
24	692	80	0,4986031	0,4986031
25	526	72	0,4997806	0,4997806
26	374	60	0,4997569	0,4997569
27	252	48	0,4962522	0,4962522
28	152	36	0,493891	0,493891
29	76	24	0,5	0,5
30	24	12	0,5	0,5
31	3644	190	0,4994897	0,4997744

#	A	B	C	D
1	Area	Perimetro	Centroide x	Centroide en Y
2	36	20	0,5	0,5
3	100	36	0,5	0,5
4	196	52	0,5	0,5
5	324	68	0,5	0,5
6	484	84	0,5	0,5
7	676	100	0,5	0,5
8	900	116	0,5	0,5
9	1156	132	0,5	0,5
10	1444	148	0,5	0,5
11	1764	164	0,5	0,5
12	2116	180	0,5	0,5
13	2500	196	0,5	0,5
14	2916	212	0,5	0,5
15	3364	228	0,5	0,5
16	3844	244	0,5	0,5
17	4356	260	0,5	0,5
18	4900	276	0,5	0,5
19	5476	292	0,5	0,5
20	6084	308	0,5	0,5
21	6724	324	0,5	0,5
22	7396	340	0,5	0,5
23	8100	356	0,5	0,5
24	8836	372	0,5	0,5
25	9604	388	0,5	0,5
26	10404	404	0,5	0,5
27	11236	420	0,5	0,5
28	12100	436	0,5	0,5
29	12996	452	0,5	0,5
30	13924	468	0,5	0,5
31	2227	169	0,4977782	0,4980963

#	A	B	C	D
1	Area	Perimetro	Centroide x	Centroide en Y
2	21	15	0,3611111	0,6388889
3	55	27	0,35	0,65
4	105	39	0,3452381	0,6547619
5	171	51	0,3425926	0,6574074
6	253	63	0,3409091	0,6590909
7	351	75	0,3397436	0,6602564
8	465	87	0,3388889	0,6611111
9	595	99	0,3382353	0,6617647
10	741	111	0,3377193	0,6622807
11	903	123	0,3373016	0,6626984
12	1081	135	0,3369565	0,6630435
13	1275	147	0,3366667	0,6633333
14	1485	159	0,3364198	0,6635802
15	1711	171	0,3362069	0,6637931
16	1953	183	0,3360215	0,6639785
17	2211	195	0,3358586	0,6641414
18	2485	207	0,3357143	0,6642857
19	2775	219	0,3355856	0,6644144
20	3081	231	0,3354701	0,6645299
21	3403	243	0,3353659	0,6646341
22	3741	255	0,3352713	0,6647287
23	4095	267	0,3351852	0,6648148
24	4465	279	0,3351064	0,6648936
25	4851	291	0,3350334	0,6649666
26	5253	303	0,3349673	0,6650327
27	5671	315	0,3349057	0,6650943
28	6105	327	0,3348485	0,6651515
29	6555	339	0,3347953	0,6652047
30	7021	351	0,3347458	0,6652542
31	1213	181	0,481931	0,6596349

Figura 5.6: Tablas de parámetros para entranamiento.

En general la adquisición, procesamiento y extracción de parámetros puede observarse en la figura 5.7:

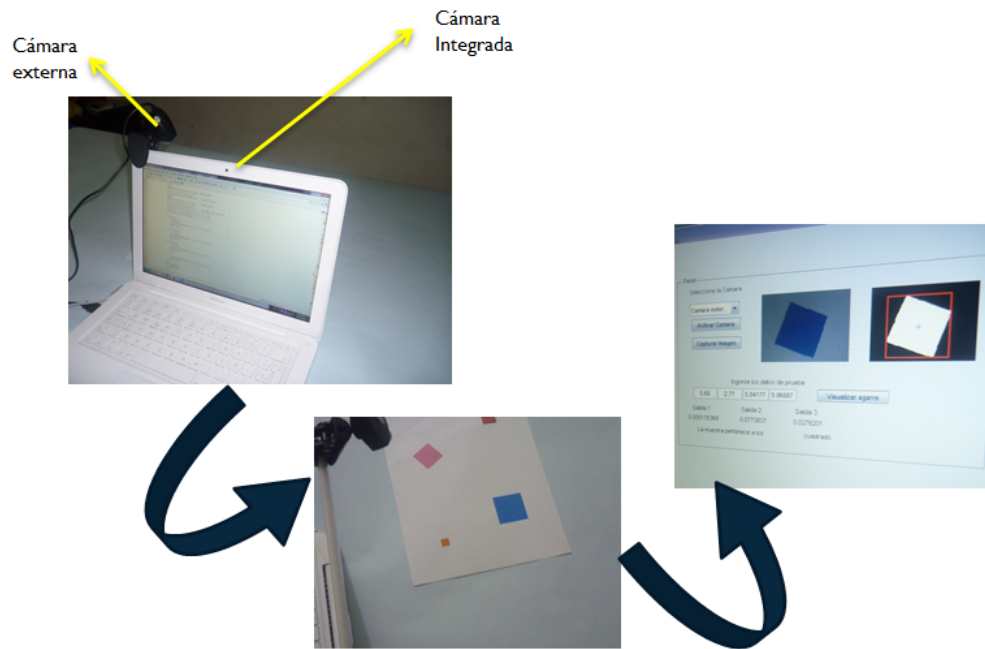


Figura 5.7: Arquitectural general de la adquisición de imágenes y extracción de parámetros.

El reconocimiento de formas se realizó en Matlab por el método de redes neuronales, ya que estas tienen la habilidad de aprender complejas relaciones no lineales de input-output usando procedimientos secuenciales de entrenamiento y además se adaptan ellas mismas a los datos.

El algoritmo implementado fue el de Backpropagation, donde una vez que se ha aplicado un patrón a la entrada de la red como estímulo, este se propaga desde la primera capa a través de las capas superiores de la red, hasta generar una salida. La señal de salida se compara con la salida deseada y se calcula una señal de error para cada una de las salidas.

Las salidas de error se propagan hacia atrás, partiendo de la capa de salida, hacia todas las neuronas de la capa oculta que contribuyen directamente a la salida. Sin embargo, las neuronas de la capa oculta solo reciben una fracción de la señal total del error, basándose aproximadamente en la contribución relativa que haya aportado cada neurona a la salida original. Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido una señal de error que describa su contribución relativa al error total. Basándose en la señal de error percibida, se actualizan los pesos de conexión de cada neurona, para hacer que la red converja hacia un estado que permita clasificar correctamente todos los patrones de entrenamiento.

La importancia de este proceso consiste en que, a medida que se entrena la red, las neuronas de las capas intermedias se organizan a sí mismas de tal modo que las distintas neuronas aprenden a reconocer distintas características del espacio total de entrada. Después del entrenamiento, cuando se les presente un patrón arbitrario de entrada que contenga ruido o que esté incompleto, las neuronas de la capa oculta de la red responderán con una salida activa si la nueva entrada contiene un patrón que se asemeje a aquella característica que las neuronas individuales hayan aprendido a reconocer durante su entrenamiento. Y a la inversa, las unidades de las capas ocultas tienen una tendencia a inhibir su salida si el patrón de entrada no contiene la característica para reconocer, para la cual han sido entrenadas. cuyo algoritmo es el siguiente:

- El sistema de entrenamiento mediante RBF consiste en:
- Empezar con unos pesos sinápticos cualquiera (generalmente elegidos al azar).
- Introducir unos datos de entrada (en la capa de entradas) elegidos al azar entre los datos de entrada que se van a usar para el entrenamiento.
- Dejar que la red genere un vector de datos de salida (propagación hacia delante).
- Comparar la salida generada por la red con la salida deseada.
- La diferencia obtenida entre la salida generada y la deseada (denominada error) se usa para ajustar los pesos sinápticos de las neuronas de la capa de salidas.
- El error se propaga hacia atrás, hacia la capa de neuronas anterior, y se usa para ajustar los pesos sinápticos en esta capa.
- Se continúa propagando el error hacia atrás y ajustando los pesos hasta que se alcance la capa de entradas.
- Este proceso se repetirá con los diferentes datos de entrenamiento.

En general podemos definir el error total cometido por la red como la suma de los cuadrados de los errores cometidos y cuyo objetivo es minimizar dicho error.

Principios básicos:

- Calcula el error en la salida para cada patrón.
- Ajusta los pesos en la capa de salida para reducir el error.
- Propaga los errores hacia la capa de entrada, ajustando los pesos de las capas ocultas.
- Repite este proceso de forma iterativa.
- Los pesos se pueden actualizar de dos formas:
 - Tras presentar cada patrón.
 - Tras presentar el conjunto de entrenamiento total.

Características principales:

- El algoritmo busca el mínimo de la función error a partir de un conjunto de patrones de entrenamiento.
- El algoritmo precisa que la función de activación sea diferenciable (fácilmente).
- Entrenar consiste en modificar los pesos de la red.
- Los pesos se modifican hacia la dirección descendente de la función error.
- La red entrenada es capaz de generalizar, clasificando correctamente patrones ruidosos e incompletos.

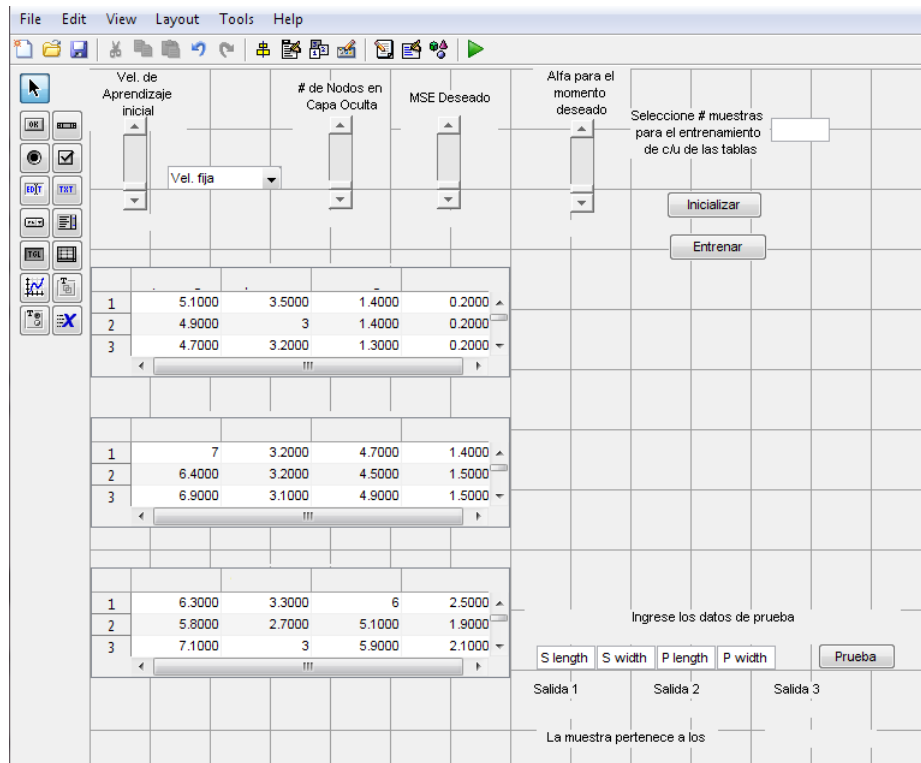


Figura 5.8: Entorno Red N

El entorno de la red se visualiza en la figura 5.8. Donde se puede observar que se dejaron modificables los siguientes parametros:

- Velocidad de aprendizaje.
- Velocidad fija o variable.
- Numero de nodos de la capa oculta.
- MSE deseado.
- Alfa para el momento deseado.
- Numero de muestras para realizar el entrenamiento.

La red consta de cuatro entradas, que corresponden a los cuatro parametros que se extraen en el procesamiento de imagenes y tres salidas, que corresponden a:

- Si el vector de salida es aproximadamente 0.99 0.1 0.1 es circulo.
- Si el vector de salida es aproximadamente 0.1 0.99 0.1 es cuadrado.
- Si el vector de salida es aproximadamente 0.1 0.1 0.99 es triangulo.

5.2. Diseño del Entorno Virtual

El entorno virtual es básicamente una representación del mundo a través de medios electrónicos, cuyo objetivo es dar la sensación de estar en una situación real en la que podemos interactuar con lo que nos rodea.

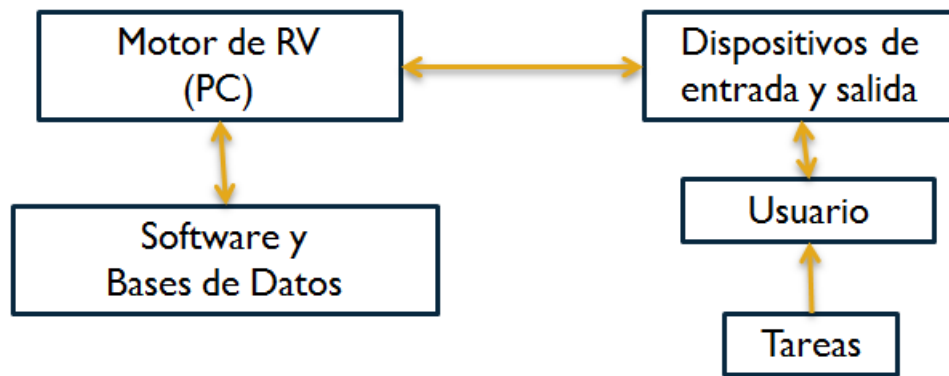


Figura 5.9: Esquema general de un entorno de RV.

En el trabajo se utilizó un modelo CAD creado en SolidWorks™, que fue facilitado por el director de tesis. Para empezar a trabajar sobre dicho modelo se decidió que tiene muchos elementos que pueden influir negativamente sobre la simulación, la razón principal es el manejo de superficies de los mecanismos en los movimientos de cada falange por esto fue necesario quitar los mecanismos que facilitan el movimiento de las falanges en la mano.

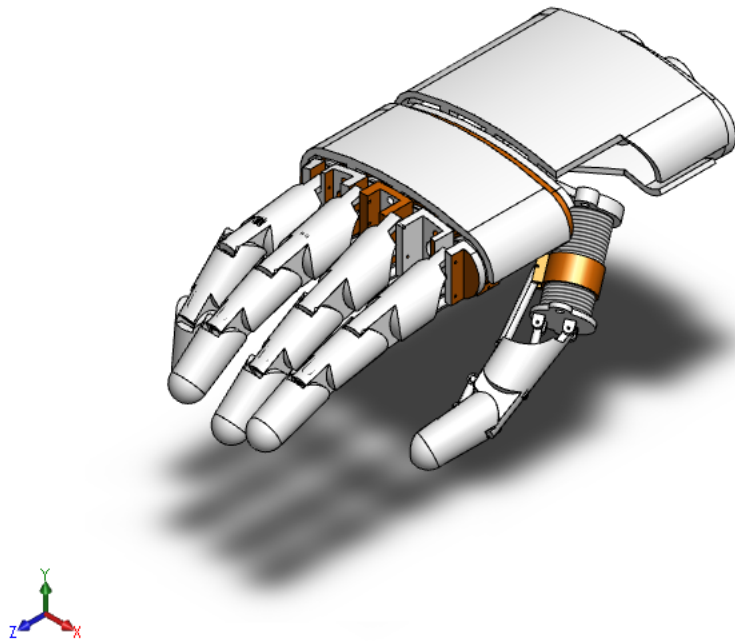


Figura 5.10: Modelo Original de la Mano con Mecanismos (Vista Superior).

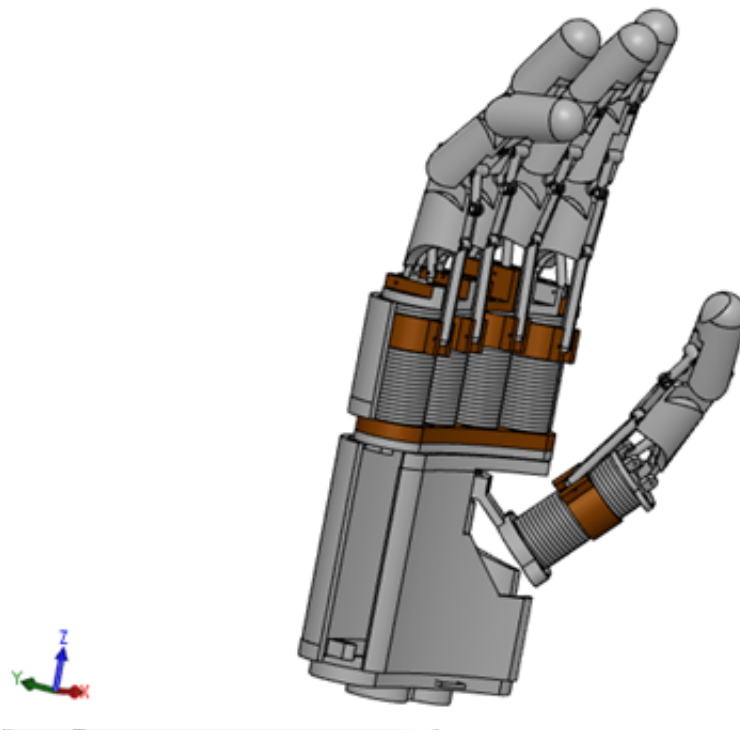


Figura 5.11: Modelo Original de la Mano con Mecanismos (Vista Frontal).

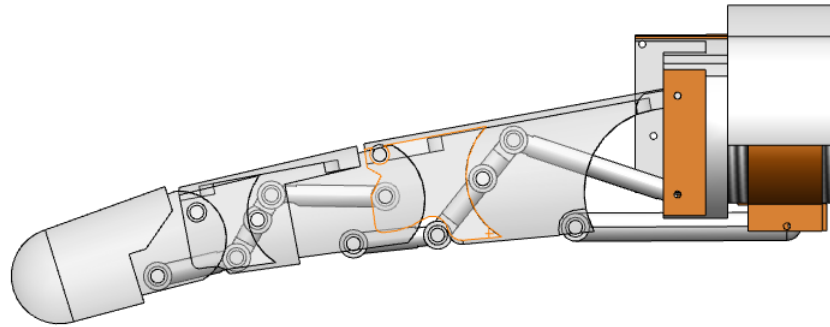


Figura 5.12: Modelo de la Falange con Mecanismos.

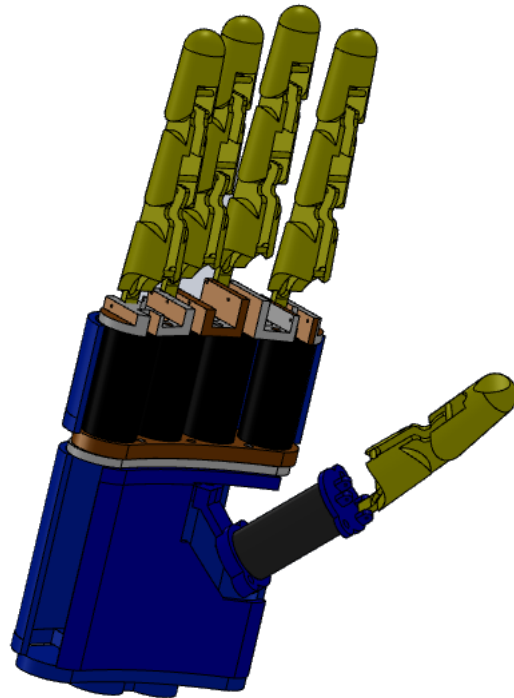


Figura 5.13: Modelo de la Mano Modificada sin Mecanismos.

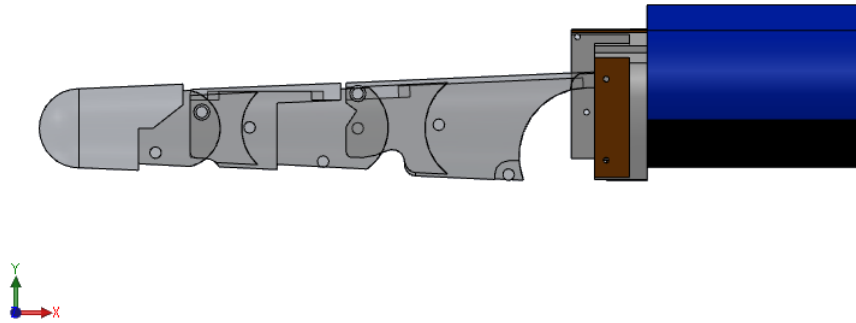


Figura 5.14: Modelo Modificado de la Falange sin Mecanismos

Al quitar los mecanismos del modelo es necesario observar que cada pieza del ensamble tenga un origen coherente respecto a su ubicación en la mano, para que en el momento de ensamblar todas las piezas no ocurran errores en la posición del origen como se muestra en la figura 5.15.

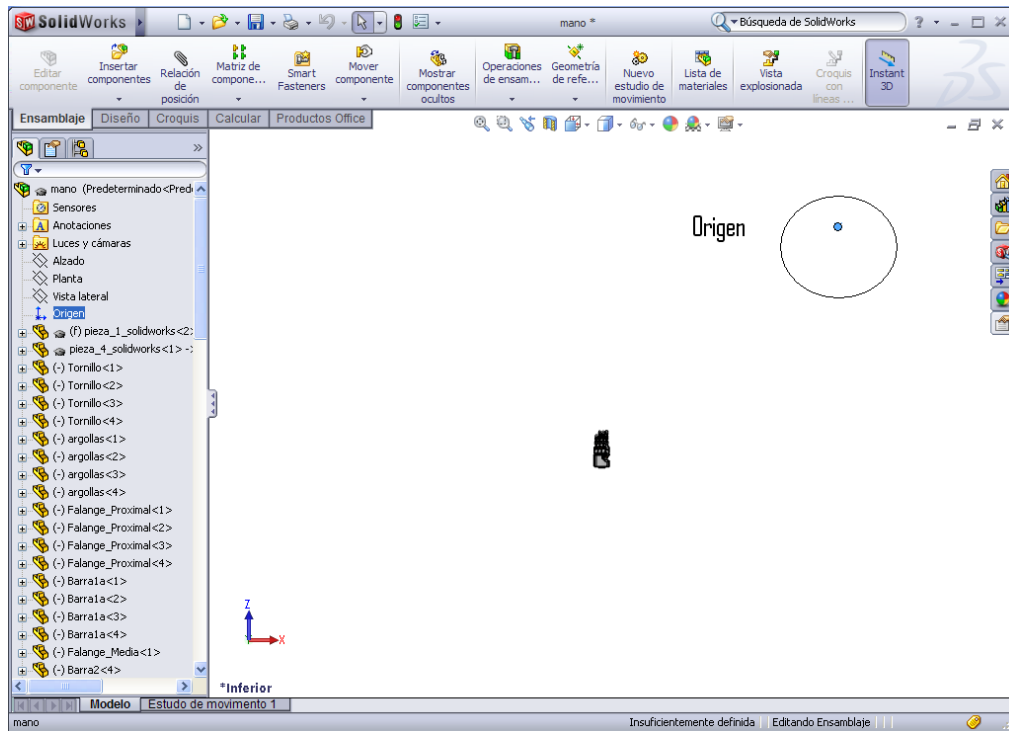


Figura 5.15: Ubicación Errónea del Origen.

El origen inicial de todo un ensamble está ubicado en un sitio que no es el apropiado, ya que al importar este modelo con este error en un programa de desarrollo 3D, ya que es posible que las piezas giren alrededor de este punto.

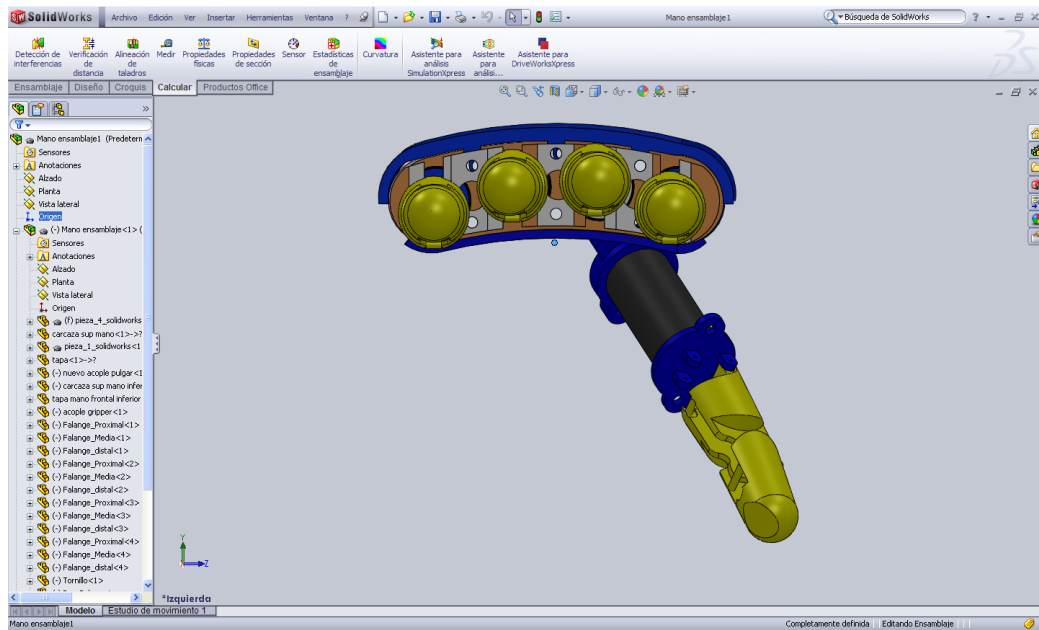


Figura 5.16: Corrección de la Ubicación del Origen en el Ensamble.

Al realizar corregir la ubicación del origen es importante que esté en el centro de la mano, con el fin que los dedos giren en alrededor de este punto, obteniendo como resultado el origen entre los dedos y la palma como se observa en la figura5.16.

El siguiente paso a seguir es la exportación del CAD a .WRL (Web Rule Language), como se puede ver en la figura 5.17. Se exporta a este formato con el fin de ser importado por un programa suministrado en Matlab™ llamado V•Realm™ Builder donde se han realizado simulaciones en asociación con Simulink™ como es el caso de la modelización matemática de un auto, vinculado a un entorno de realidad virtual.

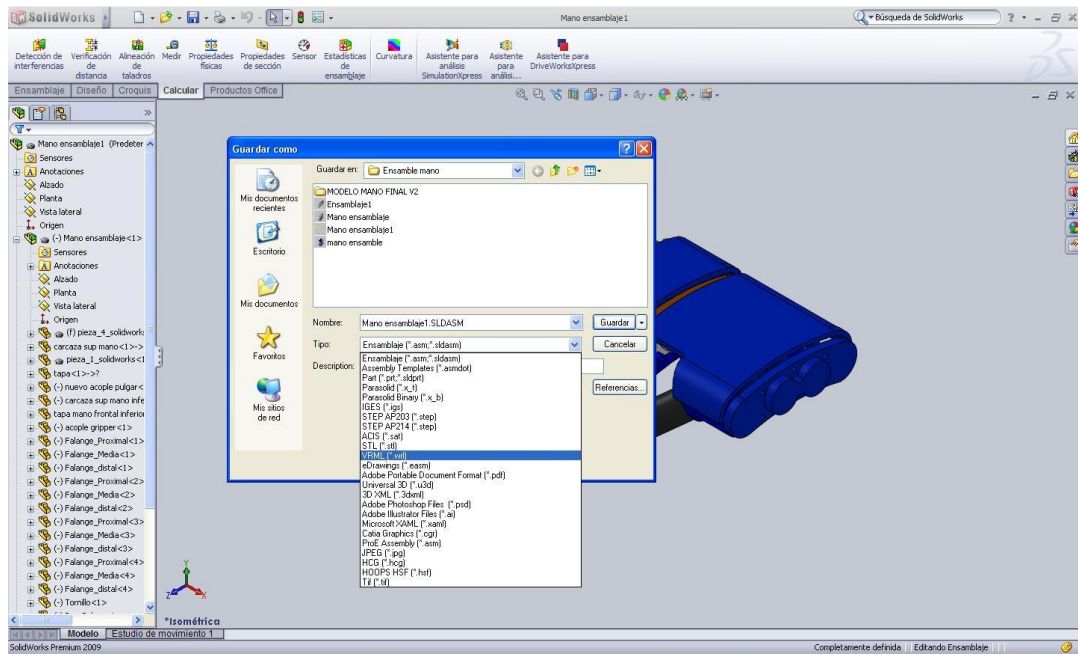


Figura 5.17: Exportación de SolidWorks™ a V•Realm™ Builder

El VRB (V•Realm™ Builder) puede importar el modelo como ensamble o pieza en el caso de este trabajo se importo todo el ensamble con el origen correctamente ubicado siendo este el punto de partida para distribuir centros de rotación en cada falange en el V•Realm™ Builder. Para implementar el modelo CAD en un entorno 3D es necesario crear un escenario donde se realizaran los movimientos básicos de los dedos.

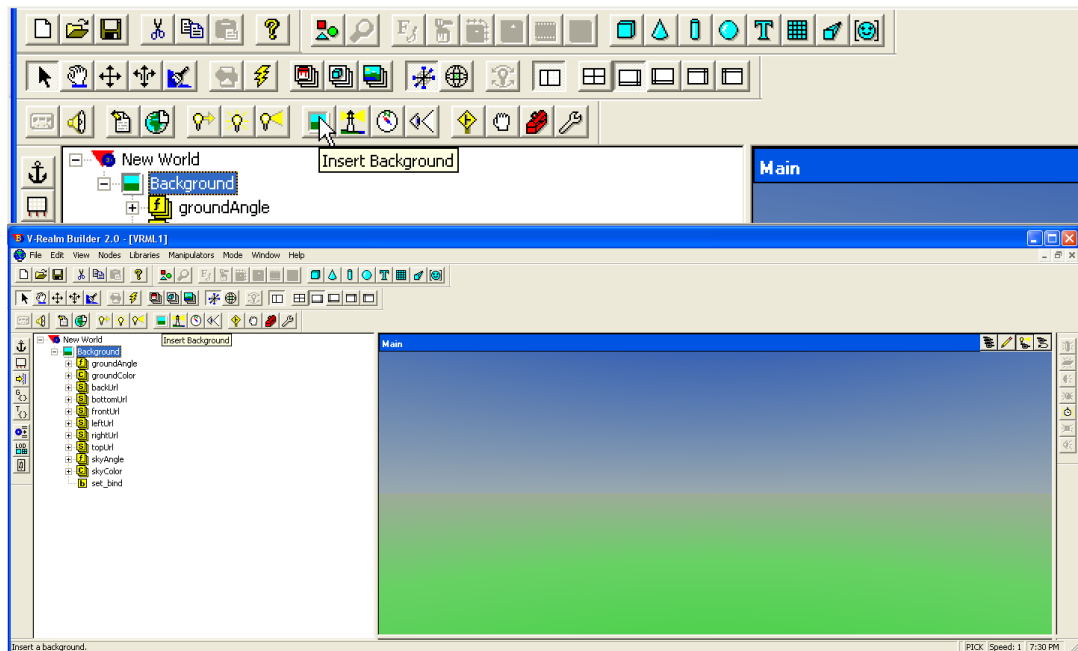


Figura 5.18: Creación de un Escenario en V•Realm™ Builder.

En la figura 5.18 se crea un escenario predeterminado en la que cambiando los parámetros de la opción “Background” puede variar color, ángulo del cielo y terreno.

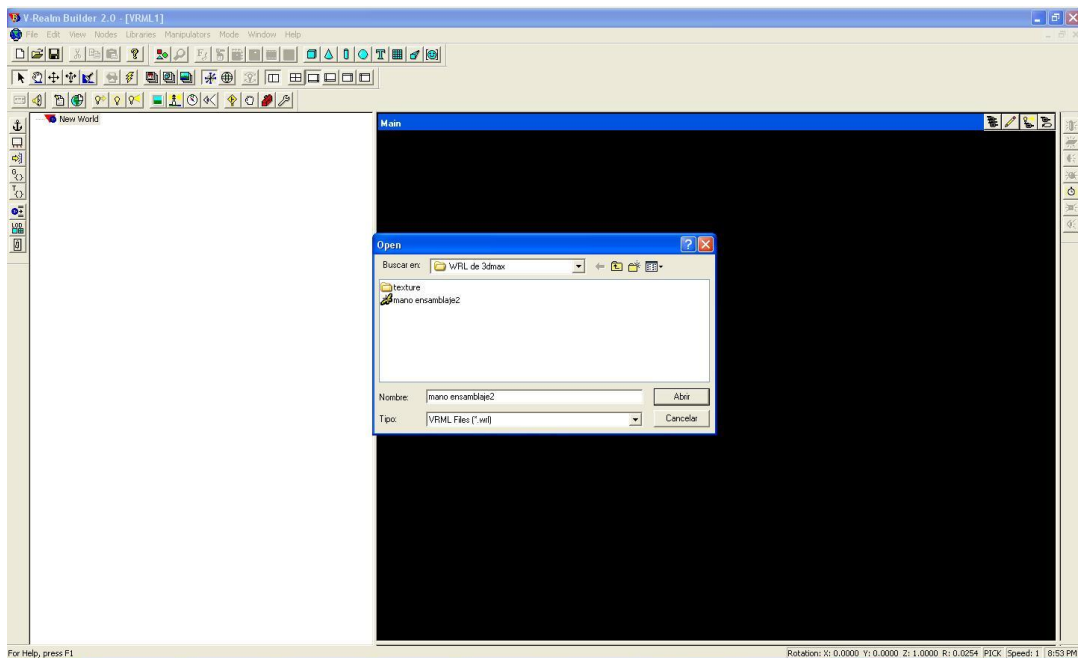


Figura 5.19: Importación del Modelo CAD a V•Realm™ Builder

En la figura 5.19 se muestra el modo de importación en el V•Realm™ Builder muestra una lista de los objetos que se pueden manejar independiente dentro del entorno 3D, como se observa en la 5.20.

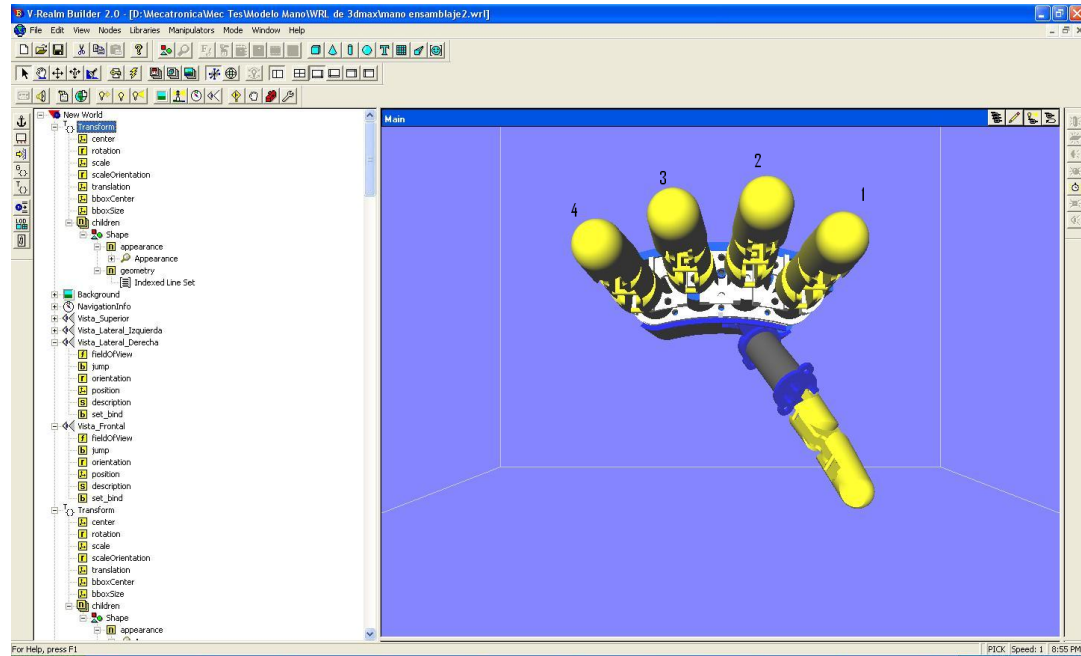


Figura 5.20: Modelo CAD Importado en V•Realm™ Builder

Al tener el ensamble en V•Realm™ Builder el software brinda opciones para manejar cada pieza del ensamble como “IndexedFace”, esto quiere decir que permite representar figuras con una forma no común por medio de la unión de coordenadas mediante caras poligonales. Cada pieza del ensamble tiene una transformación que permite rotar, trasladar y escalar. Para obtener el centro de rotación de cada pieza es necesario recurrir a la herramienta “medir” en el programa SolidWorks™, esta ayuda a medir distancias y coordenadas del punto en la medición respecto al origen de la pieza ó ensamble.

A continuación se presentan las tablas de centros para cada uno de los dedos de la mano:

	x	y	z
Falange prox1	-0.93	17.89	27.68
Falange media1	-43.93	15.39	26.58
Falange distal1	-70.93	12.19	26.98

Cuadro 5.1: Centros de Rotación Dedo 1.

	x	y	z
Falange prox2	-5.93	25.18	8.08
Falange media2	-48.93	22.68	6.98
Falange distal2	-75.93	19.48	7.38

Cuadro 5.2: Centros de Rotación Dedo 2.

	x	y	z
Falange prox3	-2.07	22.95	-13.50
Falange media3	-45.07	20.45	-7.40
Falange Fdistal3	-72.07	17.26	-7.80

Cuadro 5.3: Centros de Rotación Dedo 3.

	x	y	z
Falange prox4	0.93	15.98	-34.1
Falange media4	-42.07	13.49	-27.20
Falange distal4	-69.07	10.29	-27.6

Cuadro 5.4: Centros de Rotación Dedo 4.

	x	y	z
Falange prox	51.64	-36.41	43.45
Falange media	16.08	-56.62	58.27

Cuadro 5.5: Centros de Rotación Pulgar.

Con los centros de rotación de cada falange es necesario introducirlos en el V•Realm™ Builder en la opción “center” o directamente por código en el VRMLPAD, este software muestra el código de todo el entorno. El siguiente paso es determinar el eje en el que va a rotar cada uno de los dedos en la opción “Rotation”, en este trabajo se utilizó el eje z para los dedos (Índice, Medio, Anular, Meñique) y en el caso del dedo pulgar es una combinación de ejes.

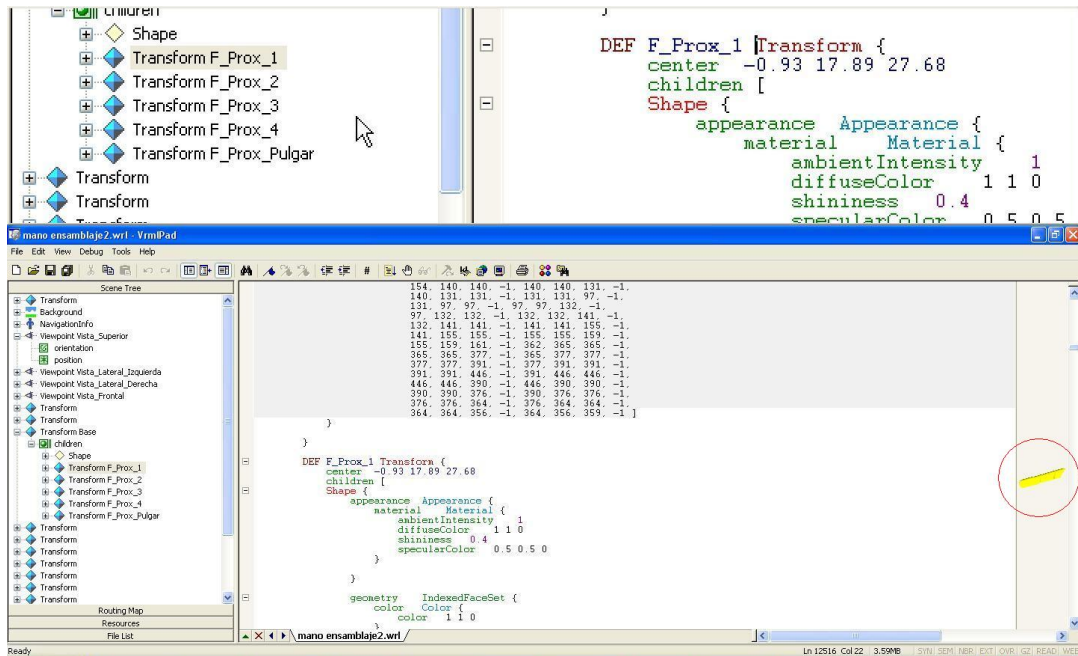


Figura 5.21: Código en VRMLPAD, El Círculo Rojo Muestra el Objeto Sobre el que se Está Efectuando los Cambios.

Código en VRMLPAD, el círculo rojo muestra el objeto sobre el que se está efectuando los cambios.

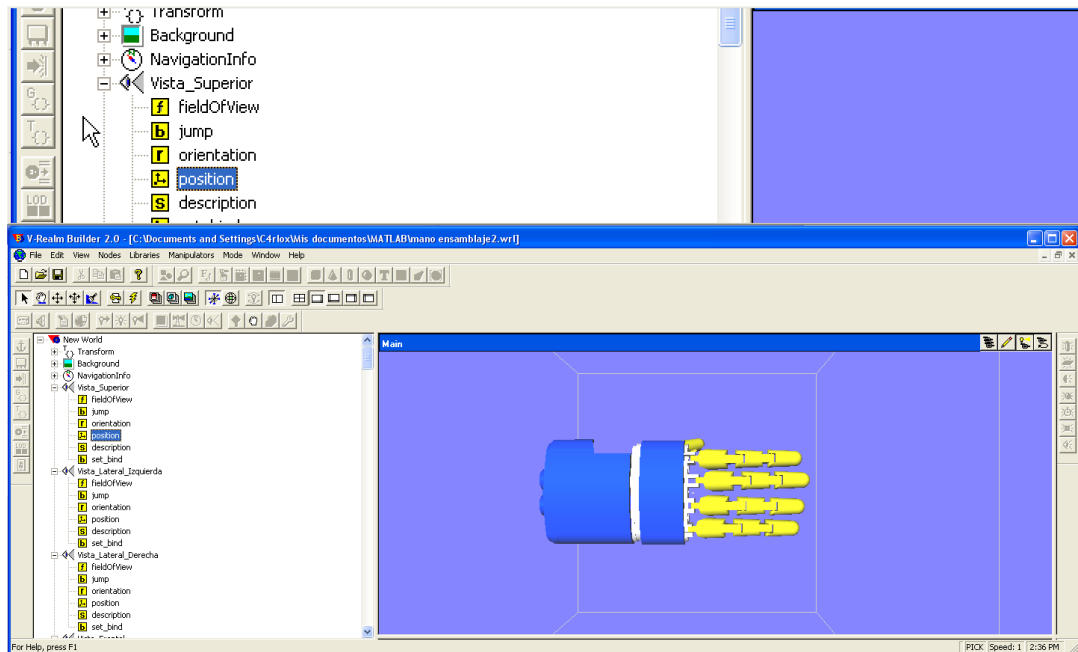


Figura 5.22: Posición, Orientación de Cámaras y Focos de Luz

Es indispensable tener configurados los centros y rotaciones en los dedos de la mano, para proceder a crear un escenario, un foco de luz y puntos vistas ó cámaras. En la opción “Position” y “Orientation” determinan la ubicación de la cámara para las diferentes vistas de la mano, de igual manera se configura los focos de luz. Para continuar en el desarrollo del software es necesario incorporar el entorno con un GUI (Graphical user interface) de Matlab™ para pruebas practicas de esta simulación se creó un programa para mover manualmente cada falange seleccionando el dedo, con el fin de utilizar el movimiento en tiempo real realizado a través de un trackbar o slider.

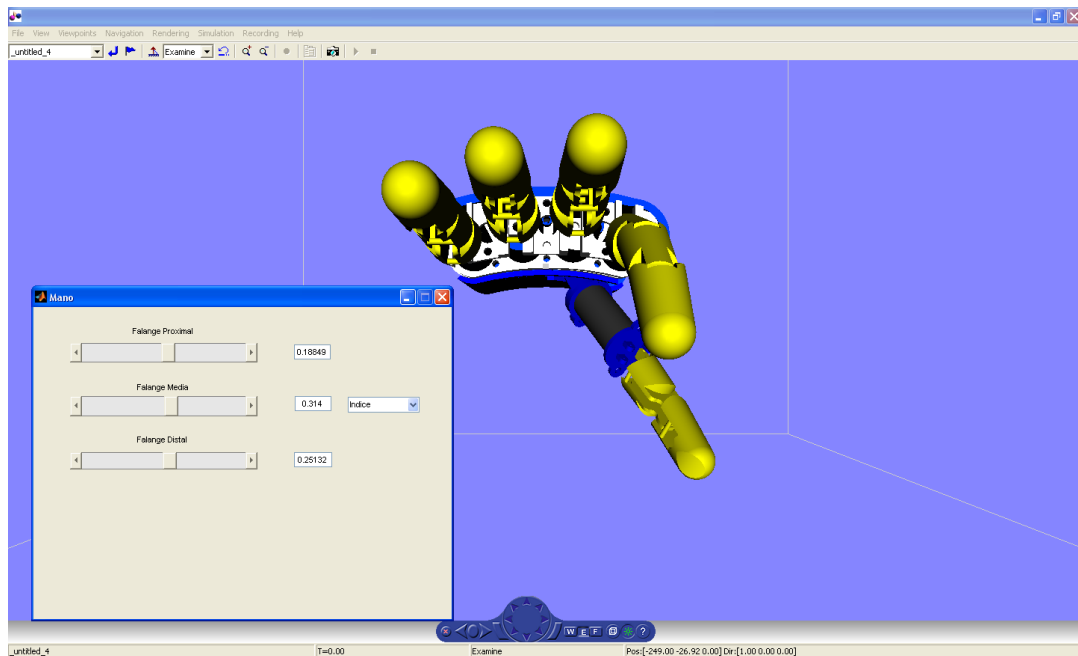


Figura 5.23: Vinculación de V•Realm™ Builder con Matlab™ GUI.

Para unir el escenario creado en V•Realm™ Builder con la interfaz grafica de Matlab™ es necesario recurrir a un grupo de funciones donde es importante saber cómo se llama cada objeto en el programa de V•Realm™ para eso se puede cambiar desde este programa los nombres y así poderlo directamente en el código de la aplicación. Para el caso de los demás es necesario alternar los números dependiendo del dedo.

```
wh = vrworld('mano ensamblaje2.wrl');
open(wh);
view(wh);
handles.F_Prox_1 = vnode(wh,'F_Prox_1');
```

```
handles.F_Media_1 = vrnnode(wh,'F_Media_1');
```

```
handles.F_Distal_1 = vrnnode(wh,'F_Distal_1');
```

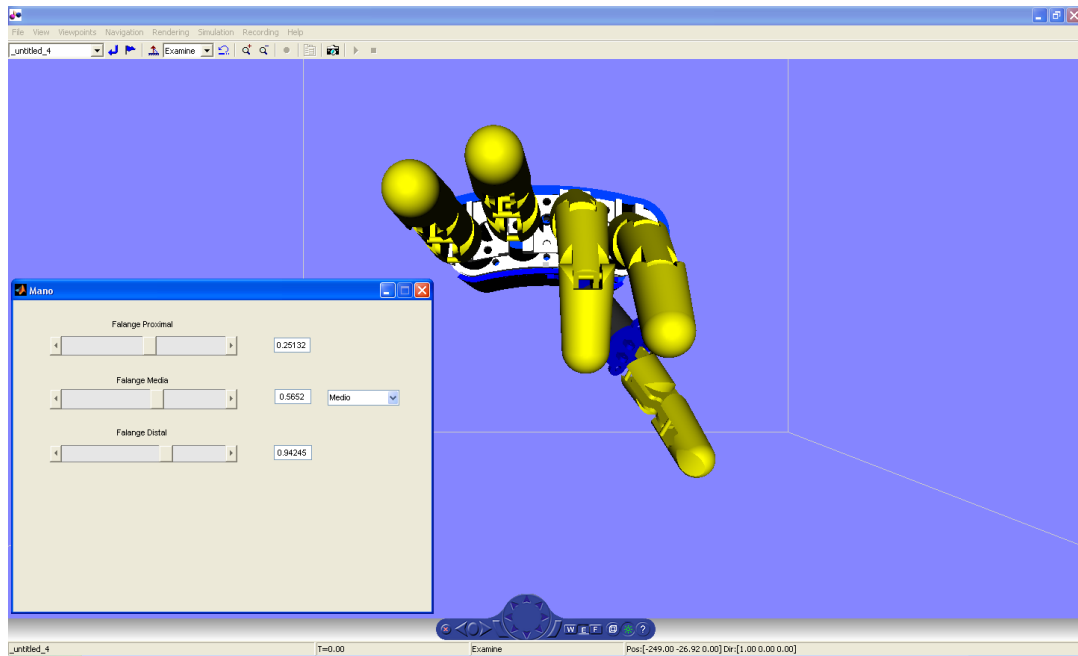


Figura 5.24: Variación de Ángulos de Cada Falange Manualmente.

5.3. Diagrama de Flujo del Proceso.

A continuación en la figura 5.25 y 5.26 se muestra el diagrama de flujo del proceso de la simulación.

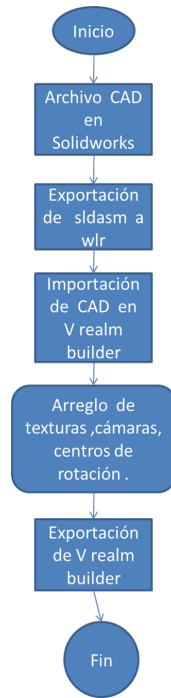


Figura 5.25: Diagrama de Flujo del Proceso I de la Simulación.

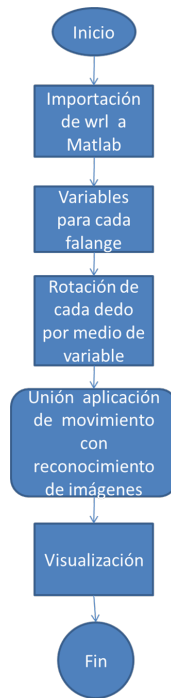


Figura 5.26: Diagrama de Flujo del Proceso II de la Simulación.

Capítulo 6

Conclusiones

El desarrollo de manos Robóticas Antropomórficas, es parte de una serie de investigaciones que se están realizando alrededor del mundo con la finalidad de desarrollar robots en un conjunto que sean capaces de imitar los movimientos y habilidades del ser humano, o lo que se conoce como androides.

Actualmente se están realizando varios estudios con diferentes tipos de agarre, caracterizando los movimientos del dedo y capturando cada una de las posiciones angulares.

En el anterior documento se analizaron las preformas de agarre para un conjunto de formas primitivas en una mano robótica antropomorfa, implementando las herramientas de reconocimiento de formas por medio de redes neuronales.

La red neuronal utilizada para la detección de formas tiene la capacidad de variar parámetros importantes como los son la velocidad de aprendizaje, el número de clústers y el error deseado, aumentando la confiabilidad de detección de esta. Aunque se pudo implementar algún tipo de toolbox de redes neuronales se decidió construirla ya que muchos de estos no tienen la flexibilidad de variar ciertos parámetros.

La simulación es una herramienta para verificar el agarre de una mano robótica Antropomórfica con el fin de planificar trayectorias de grasping o de agarre para una implementación física del proyecto.

En futuros trabajos se espera expandir las experiencias conseguidas con este software para la implementación física de una plataforma en la que se implementen técnicas avanzadas de control en el agarre de objetos con realimentación visual y tal

vez implementar un sistema de visión que sea capaz de distinguir las dimensiones de un objeto en 3D.

Bibliografía

- [1] F. García Córdova J.J Feliu Batlle J. López Coronado A. Martínez Ramírez, J. Molina Vilaplana. Estudio y desarrollo de primitivas motoras para manipulación con manos robóticas antropomorfas. Master's thesis, Departamento de Ingeniería de Sistemas y Automática. Universidad Politécnica de Cartagena., 2001.
- [2] V. Santos F. Valero-Cuevas A. Miller., P. Allen.. *From Robotic Hands to Human Hands: A Visualization and Simulation Engine for Grasping Research*. Dept. of Computer Science, Columbia University, NY, USA, Neuromuscular Biomechanics Laboratory, Cornell University, NY, USA ,The Hospital for Special Surgery, NY, USA, 2003.
- [3] Velázquez Sánchez Alejandro. *Caracterización Cinemática e Implementación de una Mano Robótica Multiarticulada*. PhD thesis, Instituto Politécnico Nacional de Mexico, 2007.
- [4] S. Haidacher H. Liu-G. Hirzinger Ch. Borst, M. Fischer. *Experiments and Experiences with an Anthropomorphic Hand*. German Aerospace Center - DLR. Institute for Robotics and Mechatronics 82230 Wessling, Germany, 2000.
- [5] Diego Aracena Pizarro. Cristián Arriagada García. Detección y reconocimiento de señales de tránsito utilizando matching de chamfer. *Ingeniare. Revista chilena de ingeniería*, 15 N° 2:174–184, 2007.
- [6] Steven Mackay Dennis Hong. Raphael (robotic air powered hand with elastic ligaments). *Virginia Tech College of Engineering*, page <http://www.eng.vt.edu/news/article.php?niid=1686>, 2009.
- [7] José Lorenzo N. Francisco Hembdez T. *Reconocimiento de Formas - Clasificación y Aprendizaje*. Universidad de Las Palmas de Gran Canaria, mayo 2002.

- [8] Asfour T. Dillman R. Fukaya N., Toyama S. Design of the tuat/karlsruhe humanoid hand. Master's thesis, Department of Engineering of Mechanic System, University Tokio Japan, 1999.
- [9] S. Knoch J. Butterfass, G. Hirzinger and H. Liu. Dlr's multisensory articulated hand. Master's thesis, In Proc. of the 1998 IEEE Int. Conf. on Robotics and Automation, Leuven, Belgica, , Mayo 1998.
- [10] Ocampo Gustavo A. Rondón Julián. Diseño y construcción de una mano robot de cuatro dedos que imite los modelos prensiles humanos. Master's thesis, Universidad Militar Nueva Granada, 2004.
- [11] Harold Salazar Maria Isabel Acosta, Camilo Zuluaga. Tutorial de redes neu. Master's thesis, Universidad Tecnológica de Pereira, 2000.
- [12] Paola A. Niño Oscar F. Aviles, Pedro L. Simanca. Diseño y construcción de un dedo para grippers roboticos. *MEMORIAS CONGRESO PUCP*, 283:6, 2007.
- [13] Álvaro Joffre Paola Niño Oscar F Avilés S, João Mauricio Rosário. Diseño de un sistema mecatrónicoantropomórfico de cinco dedos. page 7, 2008.
- [14] Sandra María Villa Palacios. *Sistema de Reconocimiento de Rostros*. Universidad Peruana de Ciencias Aplicadas, 2007.
- [15] D. F. Knutti S. C. Jacobsen, J. E. Wood and K. B. Biggers. The utah/mit dextrous hand: Work in progress. Master's thesis, In Robot, Grippers, Springer-Verlag, Berlin., 1986.
- [16] Héctor Miguel Rulot Segovia. *ECGI - Un algoritmo de Inferencia Gramatical mediante Corrección de Errores*. Universitat de València, 1992.
- [17] Giraldo Ramos Frank Nixon. Serrato Panqueba Beatriz Nathalia. *Reconocimiento del Lenguaje de Gestos Manuales Alfabéticos Mediante Visión Artificial*. Universidad Distrital Francisco José De Caldas, 2009.

Apéndice A

Tutorial basico de VRML

VRML es un lenguaje de definición de escenas cuyo objetivo es la descripción de entornos virtuales 3D que puedan transmitirse e inter-relacionarse a través del WWW. La idea surgió en la primera World Wide Web Conference, en Ginebra en 1994. La implementación del VRML en la creación del entorno virtual se desarrollo por medio de diferentes herramientas como lo son:

- Solidworks
- Vrealm Builder 2.0
- VrmlPad 3.0

Construcción de formas primitivas en VRML

Las formas son los elementos que permiten visualizar los objetos en los mundos VRML.

La sintaxis del nodo Shape es la siguiente:

```
Shape{  
appearance ...  
geometry ... }
```

El campo appearance especifica las propiedades en cuanto a textura, material, etc del objeto que se describe en el campo geometry.

Se llaman formas primitivas a los nodos geométricos primitivos que usa el VRML como son:

- Box
- Cone
- Cylinder
- Sphere

Para entender un poco más las características de esta herramienta se explica cada una a continuación:

Nodo primitivo Box:

Sintaxis:

```
Box{  
size anchura altura profundidad  
}
```

Ejemplo:

```
Box{  
size 2.0 0.5 3.0  
}
```

Nodo primitivo Cone:

Sintaxis:

```
Cone{  
height altura bottomRadius radio_de_la_base bottom valor_lógico side valor_lógico  
}
```

Mediante los campos bottom y side se indica si se desea dibujar la base y la superficie lateral respectivamente. Por defecto estos campos toman el valor TRUE, lo cual indica que se dibuja el cono completo.

Ejemplo:

```
Cone{  
height 3.0 bottomRadius .75  
}
```

Nodo primitivo Cylinder:

Sintaxis:

```
Cylinder{  
height altura radius radio bottom valor_lógico side valor_lógico top valor_lógico  
}
```

Mediante los campos bottom, side y top se indica si se desea dibujar la base inferior, la superficie lateral y la base superior del cilindro. Por defecto estos campos toman el valor TRUE, lo cual indica que se dibuja el cilindro completo.

Ejemplo:

```
Cylinder{  
height 2.0 radius 1.5  
}
```

Nodo primitivo Sphere:

Sintaxis:

```
Sphere{  
radius radio  
}
```

Ejemplo:

```
Sphere{  
radius 1.0  
}
```

Sin embargo, la definición de un nodo primitivo implica la definición de un objeto, pero no su visualización. Es por ello por lo que se han de englobar dentro de un nodo Shape, el cual determina la apariencia de estos objetos.

Ejemplo:

```
#VRML V2.0 utf8  
  
Shape{  
  
appearance Appearance{  
  
material Material {}  
  
}  
  
geometry Cylinder{  
  
height 2.0  
  
radius 1.5  
  
}  
  
}
```

Agrupacion de nodos

Ya teniendo las figuras construidas es necesario agruparlas en nodos característicos que determinan la función y jerarquía de las figuras en el entorno virtual.

Nodo Group:

El nodo Group permite unir un conjunto de nodos de forma que actúen como una entidad única, pero sin efectuar ninguna transformación en ellos. La principal característica de este tipo de grupo es que los objetos son creados todos en el mismo punto es decir en el origen del escenario de realidad virtual.

Sintaxis:

```
Group {  
  
children [ ... ]  
  
}
```

El campo `children` contiene la lista de los objetos que se quieren agrupar, representados por sus nodos `Shape` respectivos:

Ejemplo:

```
Group {  
  children [  
    Shape { ... },  
    Shape { ... },  
    ...  
  ]  
}
```

Nodo Transform:

Por defecto todos los objetos se construyen en el centro del escenario virtual. El nodo `transform` nos va a permitir evitar esto, indicando la posición, orientación y tamaño de los diferentes objetos que va a crear.

Sintaxis:

```
Transform{  
  translation Eje_X Eje_Y Eje_Z  
  rotation Eje_X Eje_Y Eje_Z Ángulo(radian)  
  scale Eje_X Eje_Y Eje_Z  
  children[...]  
}
```

Cada grupo creado mediante el nodo `Transform` va a poseer su propio sistema de coordenadas, cuyos atributos se determinan a través de los campos `translation`, `rotation` y `scale`, los cuales son optativos.

El campo `translation` permite indicar la posición del origen del nuevo sistema de coordenadas perteneciente al grupo dentro del sistema de coordenadas de nodo que lo engloba (nodo padre). A través del siguiente ejemplo esta idea quedará más clara:

Ejemplo:

```

Transform{
# Ejes: X Y Z
translation 2.0 0.0 0.0
children [...]
}

```

Nodo Switch:

La principal característica de un nodo Switch es la de mostrar únicamente uno de los nodos hijos del nodo, el cual ha debido ser seleccionado previamente. Se pueden utilizar para conectar o desconectar los efectos de una determinada propiedad o para alternar entre propiedades diferentes.

El campo *whichChild* especifica el hijo choice que se va activar, siendo el 0 el del primer hijo. Su valor por defecto es -1, lo cual indica que ninguno de los hijos está seleccionado.

El campo *whichChild* es una entrada y por lo tanto puede ser modificado por otro nodo.

Sintaxis:

```

Switch{
whichChoice 0
choice[...]
...
choice[...]
}

```

Nodo Billboard:

El nodo Billboard permite crear un grupo con un sistema de coordenadas especiales, ya que a través del campo *axisOfRotation* (eje de rotación) indicamos el eje sobre el que a de girar el objeto, de forma que, siempre esté de cara al espectador:

Sintaxis:

Billboard{

axisOfRotation Eje_X Eje_Y Eje_Z

children[...]

}

Ejemplo 1:

1. Creacion de un escenario y un cubo en VRealm Builder

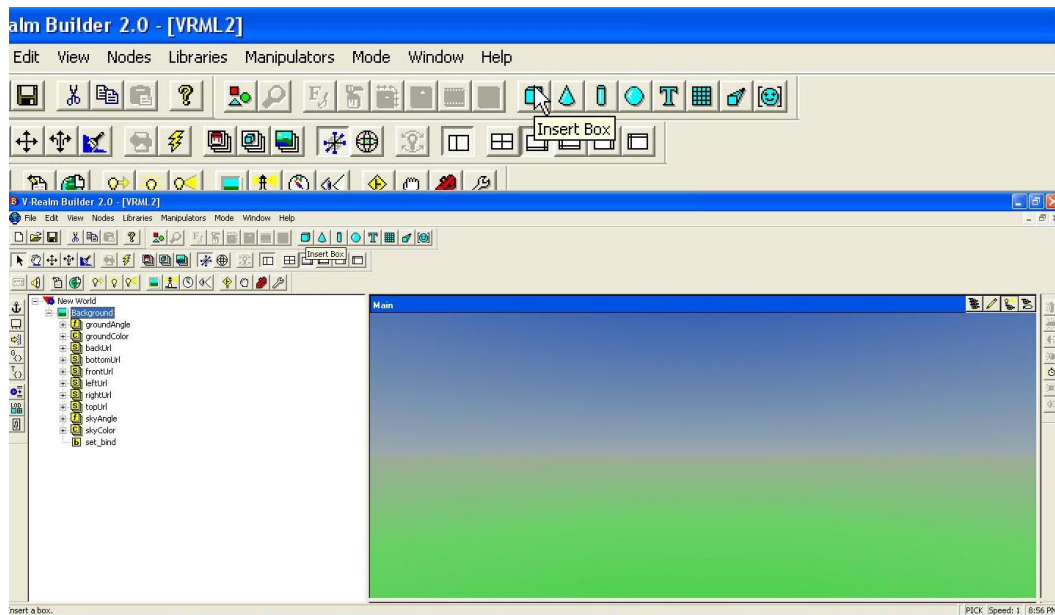


Figura A.1: Insert Box

2. Rotación de una transformación para el cubo

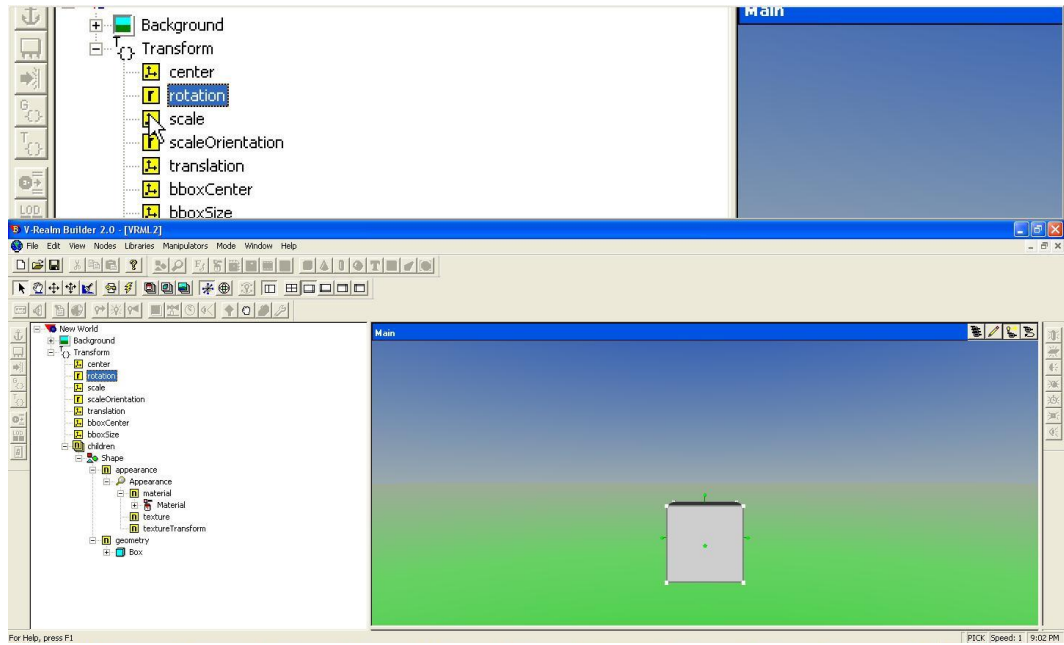


Figura A.2: Rotation

3. Selección del eje de rotación

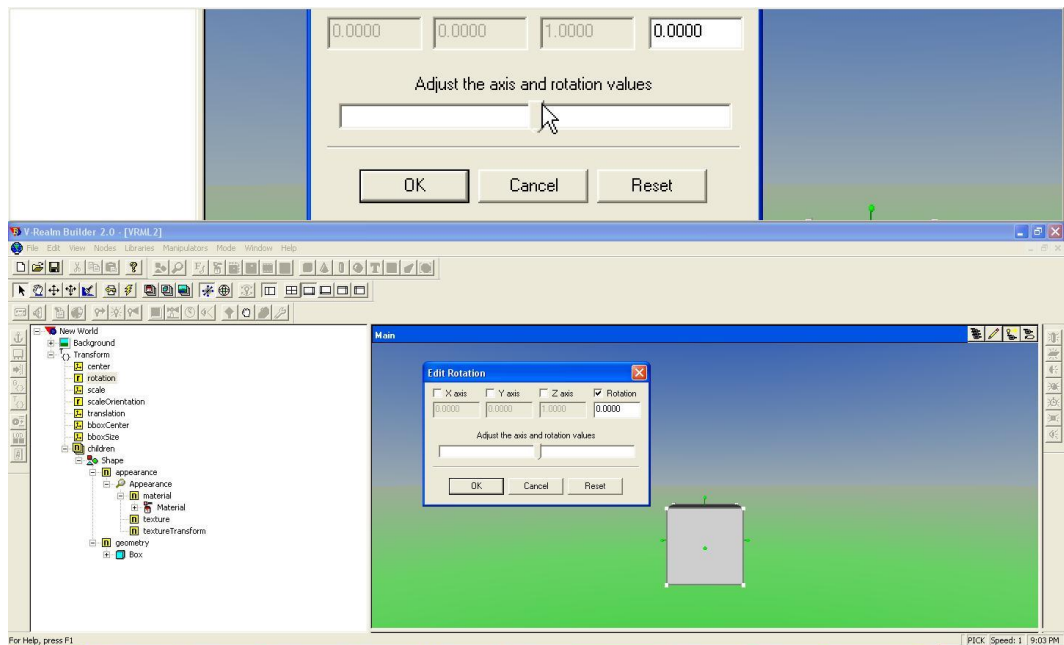


Figura A.3: Edit Rotation

4. Selección del ángulo de rotación en grados

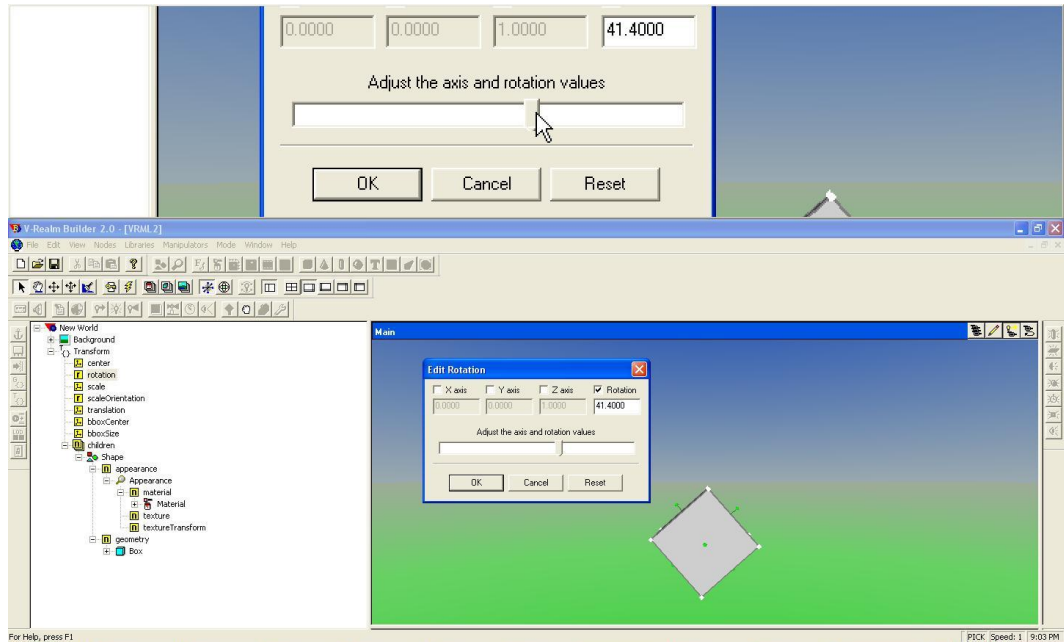


Figura A.4: Ángulo de Rotación

5. Traducción de una transformación (cubo)

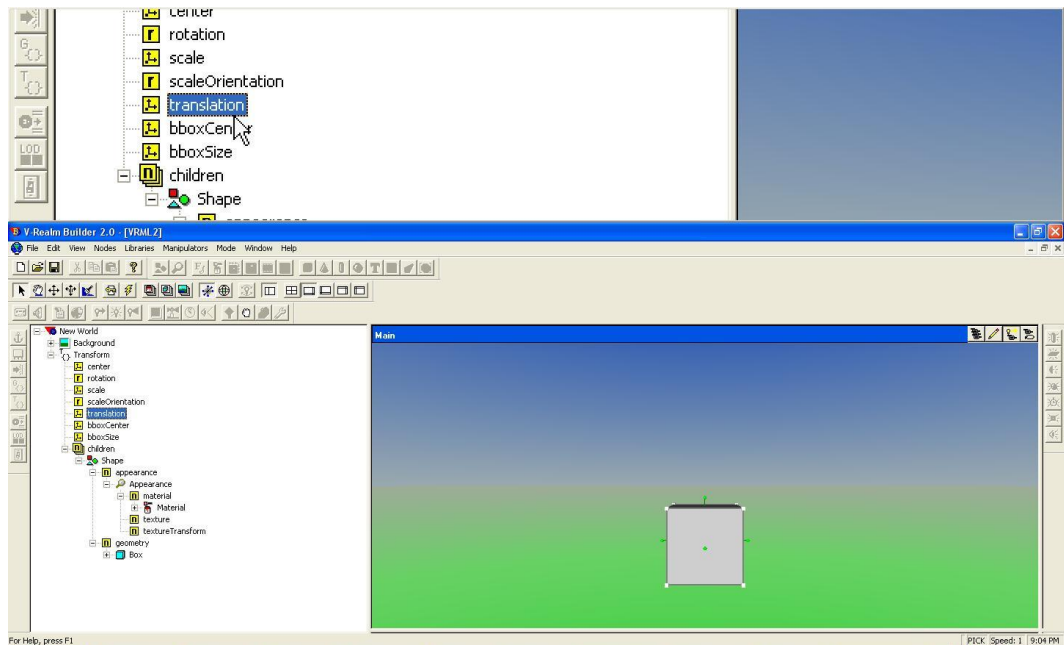


Figura A.5: Translation

Ejemplo 2:

1. Modelo CAD de una mano

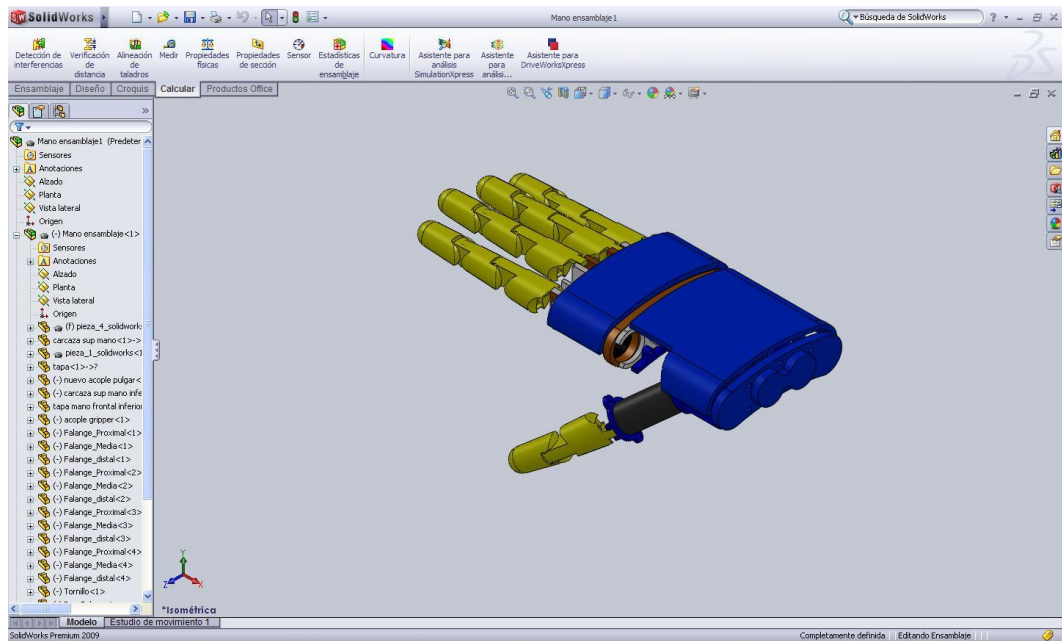


Figura A.6: Modelo CAD

2. Exportación del .SLDASM a .WRL

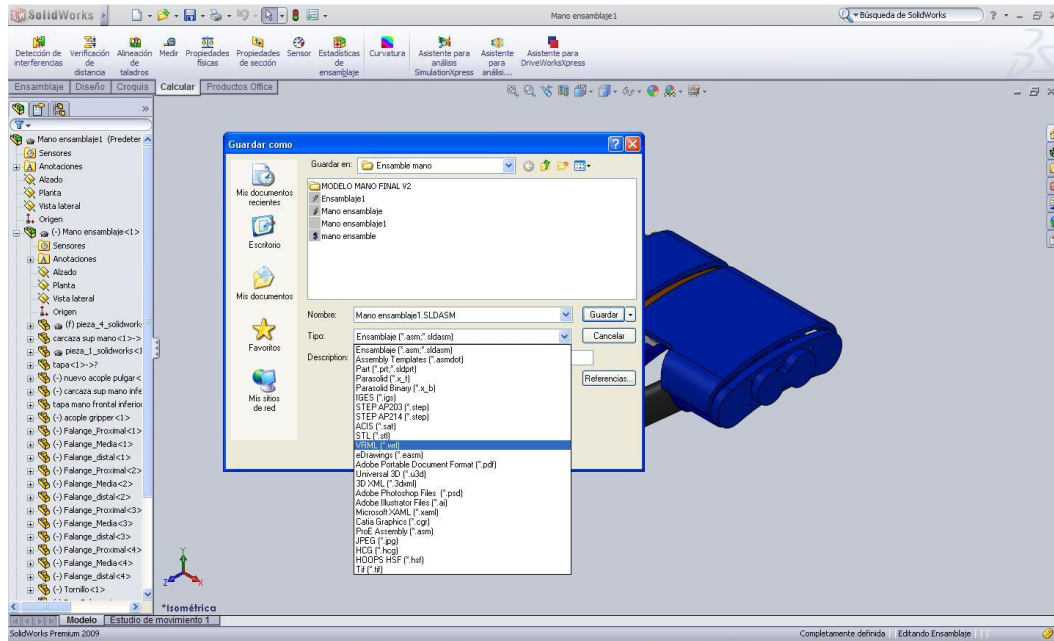


Figura A.7: Exportación

3. Importación de .WRL en VrealmBuilder

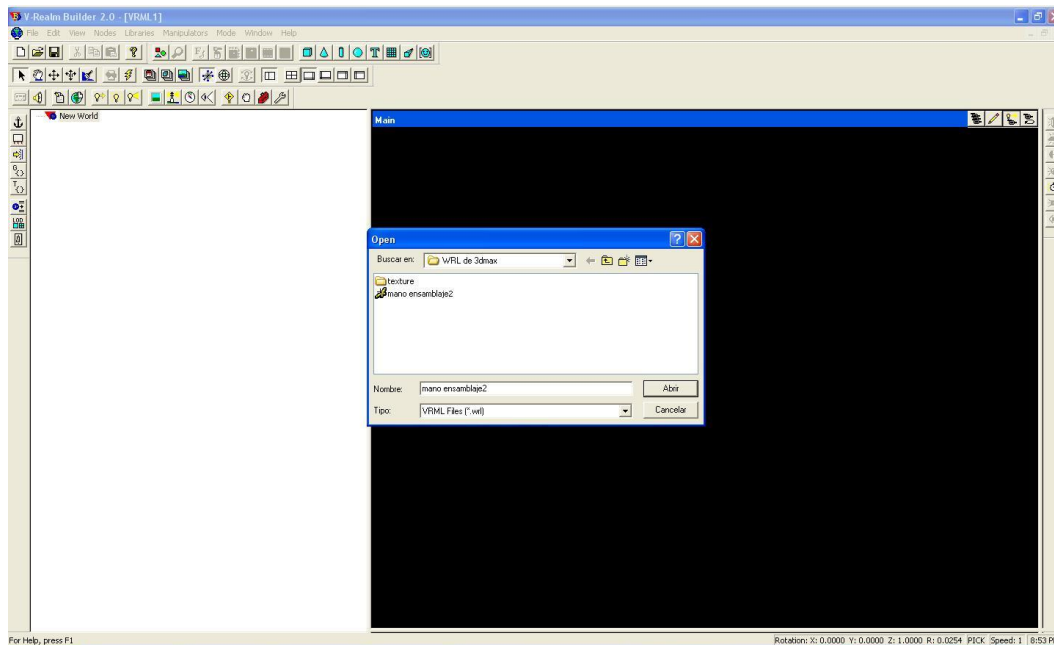


Figura A.8: Importación

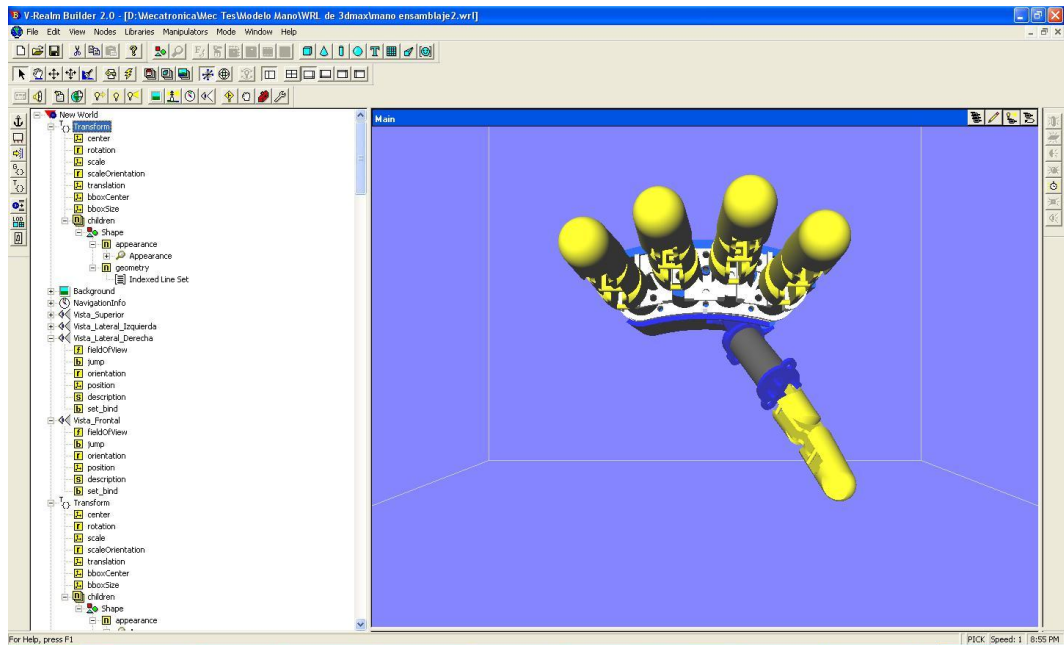


Figura A.9: Mano en VRML

4. Visualización del código generado ó programado en VRMLPAD

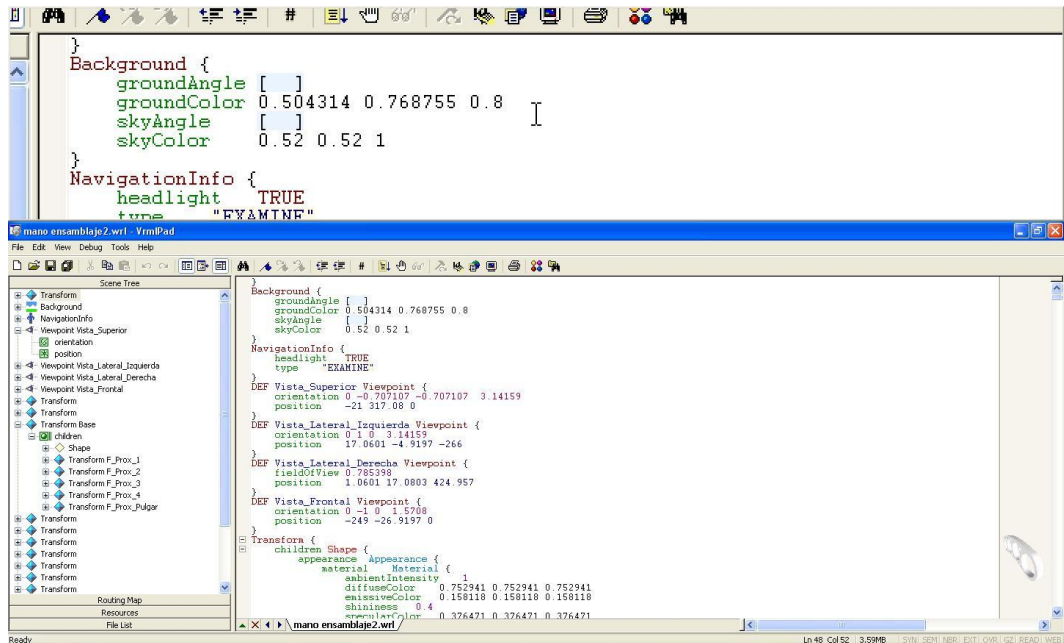


Figura A.10: Código

5. Transformaciones de las Falanges determinando centros de rotación

En el círculo rojo se observa la transformación en 3D que estamos trabajando en este caso es la Falange Proximal 1.

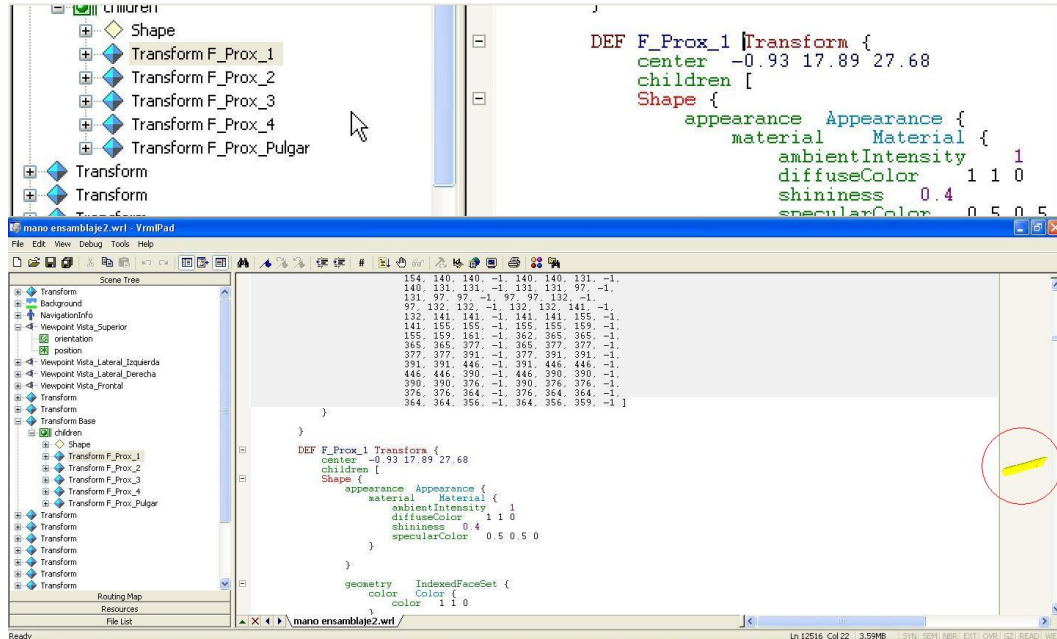


Figura A.11: Transformacion Falanges

Aplicaciones:

Arquitectura y gestión inmobiliaria:

Desarrollo de proyectos tridimensionales, que pueden ser recorridos libremente.

Decoración:

Una empresa de decoración puede ofrecer a sus usuarios la posibilidad de elegir ellos mismo, en tiempo real, la decoración que quieren y verla desde diferentes puntos de vista, conociendo también en tiempo real, el precio de las diversos elementos decorativos.

Instrucciones de montaje:

Posibilidad de realizar animaciones interactivas que muestren, de forma tridimensional, los pasos de montaje de un determinado aparato o de un kit de bricolaje.

Ciencia:

Visualización tridimensional de un número amplio de compuestos así como sus modos de vibración. Estudio de las tensiones internas que sufre un sólido al deformarse, etc. Estudios de astronomía, arqueología, etc.

Posibilidad de realizar laboratorios virtuales, en los que se demuestra determinado fenómeno físico. En estos laboratorios las condiciones del experimento se pueden modificar, observando que ocurre en cada caso.

Turismo:

Acceso a ciudades virtuales de lugares reales, para aquellas personas que no les guste o no puedan viajar.