

UNIVERSIDAD AUTÓNOMA DE SINALOA  
FACULTAD DE INFORMÁTICA Y FACULTAD DE CIENCIAS DE LA  
TIERRA Y EL ESPACIO  
MAESTRÍA EN CIENCIAS DE LA INFORMACIÓN



DISEÑO DE UN GUANTE DE DATOS DE ALTA  
PRECISIÓN BASADO EN SENSORES  
INERCIALES

Para obtener el grado de  
**Maestro en Ciencias**  
presenta

**Juan Augusto Campos Leal**

DIRECTORES:  
ULISES ZALDÍVAR COLADO,  
XIOMARA PENÉLOPE ZALDÍVAR COLADO

Culiacán, Sinaloa Febrero 2018

## ÍNDICE

<b>Índice de Figuras</b>	<b>III</b>
<b>Índice de Tablas</b>	<b>V</b>
<b>Índice de Ecuaciones</b>	<b>VI</b>
<b>Palabras clave</b>	<b>VII</b>
<b>Resumen</b>	<b>VIII</b>
<b>Agradecimientos</b>	<b>XI</b>
<b>Capítulo 1. Introducción</b>	<b>1</b>
1.1 Introducción . . . . .	1
1.2 Descripción del problema . . . . .	5
1.3 Planteamiento del problema y justificación . . . . .	5
1.3.1 Justificación . . . . .	5
1.3.2 Planteamiento del problema . . . . .	5
1.3.3 Solución propuesta . . . . .	6
1.3.4 Modelado del ambiente virtual . . . . .	8
1.3.5 Medición de los datos . . . . .	9
1.4 Objetivos . . . . .	10
1.4.1 Objetivo general . . . . .	10
1.4.2 Objetivos específicos . . . . .	10
1.5 Hipótesis . . . . .	10
1.6 Organización de la tesis . . . . .	11
<b>Capítulo 2. Marco teórico y estado del arte</b>	<b>13</b>
2.1 Ambientes virtuales . . . . .	13
2.2 Manos virtuales . . . . .	17
2.3 Sistema de captura de movimientos . . . . .	21
2.4 Guantes de datos . . . . .	30
2.5 Sistema de Captura de movimientos con sensores inerciales . . . . .	36
2.6 Análisis de resultados . . . . .	40
<b>Capítulo 3. Marco metodológico</b>	<b>43</b>
3.1 Modelado de la mano virtual . . . . .	43
3.1.1 Modelo cinemático de la mano . . . . .	44
3.1.2 Obtención de las posiciones relativas de los elementos de la mano	46

3.1.3	Cálculo de la cinemática de la mano virtual . . . . .	52
3.1.4	Rotación de elementos en ángulos no permitidos . . . . .	54
3.2	Obtención de los datos de los sensores inerciales . . . . .	55
3.2.1	Obteniendo los datos de un sensor . . . . .	58
3.2.2	Obteniendo los datos de dos sensores . . . . .	59
3.3	Interpretando los valores de los sensores . . . . .	60
3.3.1	Librería I2Cdevlib . . . . .	61
3.3.2	Calibración de los sensores (respecto al mundo) . . . . .	62
3.3.3	Obteniendo cuaterniones de los sensores . . . . .	63
3.3.4	Aumentando la precisión . . . . .	65
3.4	Multi-hilos en la lectura de los sensores en la escena virtual . . . . .	67
3.5	Calibración de los sensores en la escena virtual . . . . .	71
3.6	Prototipo del guante de datos . . . . .	73
<b>Capítulo 4. Experimentación y resultados</b>		<b>78</b>
4.1	Lectura de datos de los sensores . . . . .	78
4.1.1	Deriva (Drifting) . . . . .	79
4.1.2	Lectura de datos . . . . .	80
4.1.3	Resolución de la lectura de datos . . . . .	81
4.2	Precisión . . . . .	82
4.2.1	Precisión de los sensores . . . . .	83
4.3	Manipulación de objetos virtual . . . . .	86
4.3.1	Detección de colisiones . . . . .	86
4.3.2	Agarrado del objeto (Grasping) . . . . .	86
4.3.3	Procedimiento . . . . .	87
<b>Capítulo 5. Conclusiones</b>		<b>96</b>
5.1	Conclusiones . . . . .	96
5.2	Trabajo futuro . . . . .	97
5.3	Producción . . . . .	97
<b>Bibliografía</b>		<b>99</b>

## ÍNDICE DE FIGURAS

FIGURA 1.1	Arquitectura de sistema . . . . .	9
FIGURA 2.1	Philco HMD (Head Mounted Display). . . . .	15
FIGURA 2.2	El VLP Dataglove (1987). . . . .	16
FIGURA 2.3	Sistema CAVE. . . . .	17
FIGURA 2.4	Posturas de la mano para el agarrado de un objeto. . . . .	18
FIGURA 2.5	Mano virtual de Borst e Indugula. . . . .	19
FIGURA 2.6	Posturas de la mano para el agarrado de un objeto. . . . .	20
FIGURA 2.7	Palillos como herramientas virtuales. . . . .	20
FIGURA 2.8	Cinemática en la mano virtual de Li. . . . .	21
FIGURA 2.9	Ejemplo de sistemas de captura de movimientos óptico. . . . .	24
FIGURA 2.10	Sistema de captura de movimientos electromagnéticos. . . . .	26
FIGURA 2.11	Sistema de captura de movimientos exoesquelético. . . . .	27
FIGURA 2.12	Sistema de captura de movimientos basado en sensores inerciales. . . . .	30
FIGURA 2.13	Guante de datos de efecto hall. . . . .	31
FIGURA 2.14	Guante de datos de tipo Pinch glove. . . . .	32
FIGURA 2.15	Guante de datos CyberGlove. . . . .	33
FIGURA 2.16	CyberGrasp montando en la mano del usuario. . . . .	34
FIGURA 2.17	Posturas de la mano para el agarrado de un objeto con el guante de datos. . . . .	35
FIGURA 2.18	Guante de datos experimental de Zannatha. . . . .	36
FIGURA 2.19	Plataforma auto balanceable de Popelka. . . . .	38
FIGURA 2.20	Sistema Moven de Xsens. . . . .	39
FIGURA 2.21	Usuario con sensores inerciales para la captura de movimientos. . . . .	40
FIGURA 3.1	Modelo de la mano virtual y falanges de los dedos. . . . .	44
FIGURA 3.2	Esquema cinemático de la mano virtual. . . . .	45
FIGURA 3.3	Modelo de la placa Arduino MEGA 2560. . . . .	56
FIGURA 3.4	Sensor InvenSense MPU-6050. . . . .	57
FIGURA 3.5	Conexión física entre la placa MEGA 2560 y el sensor MPU-6050. . . . .	58
FIGURA 3.6	Conexión física entre la placa MEGA 2560 y dos sensores MPU-6050. . . . .	60
FIGURA 3.7	Macros disponibles para la obtención de datos en la librería I2Cdevlib. . . . .	61
FIGURA 3.8	Variables de compensación (Offset) de calibración. . . . .	62
FIGURA 3.9	Valores obtenidos de la librería I2Cdevlib. Los valores son obtenidos en cuaterniones. . . . .	63

FIGURA 3.10 Los valores obtenidos en cuaterniones para dos sensores (formato). . . . .	64
FIGURA 3.11 Un paquete enviado por puerto serial. . . . .	65
FIGURA 3.12 Los valores obtenidos para dos sensores con precisión aumentada (formato). . . . .	66
FIGURA 3.13 Localización de los sensores en la mano humana. . . . .	68
FIGURA 3.14 Diagrama de hilos en la aplicación de la escena virtual. . . . .	70
FIGURA 3.15 Arquitectura de sistema. . . . .	71
FIGURA 3.16 a) Colocación de los sensores. b) Validando movimientos de palma y falange proximal. . . . .	73
FIGURA 3.17 a) Colocación del sensor en el prototipo. b) Puesto en la mano del usuario. c) Durante la flexión de la falange proximal. . . . .	74
FIGURA 3.18 a) Nuevo diseño. b) c) d) Colocación de los anillos con los sensores en las falanges. . . . .	75
FIGURA 3.19 a) Nuevo diseño. b) c) d) Colocación de los anillos con los sensores en las falanges. . . . .	76
FIGURA 3.20 Prototipo final del guante de datos. . . . .	77
FIGURA 4.1 Deriva del sensor durante 1 hora. . . . .	79
FIGURA 4.2 Cuaterniones por segundo. . . . .	80
FIGURA 4.3 Datos capturados para el experimento de resolución. . . . .	81
FIGURA 4.4 Diferencias angulares entre la lectura $i$ y la lectura $i+1$ durante 1 minuto para la precisión estándar. . . . .	82
FIGURA 4.5 Diferencias angulares entre la lectura $i$ y la lectura $i+1$ durante 1 minuto para la precisión mejorada. . . . .	82
FIGURA 4.6 Movimientos realizados durante el experimento. Colocación de los sensores en la placa de pruebas. . . . .	83
FIGURA 4.7 Movimientos realizados durante un minuto. . . . .	84
FIGURA 4.8 Lectura de datos de 4 segundos (tomado de los datos de 1 minuto). . . . .	84
FIGURA 4.9 Lectura de datos de 2 segundos (tomado de los datos de 4 segundos). . . . .	85
FIGURA 4.10 Lectura de datos de 1 segundo (tomado de los datos de 2 segundos). . . . .	85
FIGURA 4.11 Sujetos con el guante de datos prototipo durante el experimento. . . . .	88
FIGURA 4.12 Sujetos con el guante de datos prototipo durante el experimento. . . . .	88

## ÍNDICE DE TABLAS

TABLA 3.1	Posición de los elementos de la mano registrados en el sistema VEDAP II. . . . .	47
TABLA 3.2	Orientación de los elementos de la mano registrados en el sistema VEDAP II. . . . .	48
TABLA 3.3	Posición relativa de los elementos de la mano virtual. . . . .	51
TABLA 4.1	Datos registrados para la realización de la tarea. TTC (Tiempo de Tarea Completa). . . . .	90
TABLA 4.2	Datos registrados para el tiempo de agarrado. TA (Tiempo de Agarrado). . . . .	91
TABLA 4.3	Datos registrados para el tiempo de localización. TL (Tiempo de Localización). . . . .	92
TABLA 4.4	Datos registrados para el error en la distancia. ED (Error en la Distancia). . . . .	93

## ÍNDICE DE ECUACIONES

3.1 Representación de una matriz . . . . .	46
3.2 Representación de un cuaternión . . . . .	49
3.3 Matriz de transformación homogénea . . . . .	49
3.4 Invertir matriz . . . . .	49
3.5 Multiplicación de la matriz inversa . . . . .	49
3.6 Matriz B2 . . . . .	49
3.7 Matriz Inversa . . . . .	50
3.8 Obtención de matriz A con respecto a B . . . . .	50
3.9 Matriz falange proximal con respecto al mundo . . . . .	52
3.10 Matriz falanges mediales y distales con respecto al mundo . . . . .	52
3.11 Obtención falange proximal del dedo índice . . . . .	52
3.12 Obtención falange medial del dedo índice . . . . .	53
3.13 Obtención falange distal del dedo índice . . . . .	53
3.14 Matriz con rotación de un ángulo en particular . . . . .	54
3.15 Calcular el ángulo del elemento anterior . . . . .	55
3.16 Aplicar el ángulo en el eje específico . . . . .	55
3.17 Aplicar el ángulo a falanges proximales en el eje específico . . . . .	55
3.18 Matriz inicial del sensor con respecto al mundo . . . . .	72
3.19 Inversa de la matriz inicial . . . . .	72
3.20 Obtención del movimiento continuo en un tiempo $t + 1$ . . . . .	72
4.1 Matriz del objeto1 con respecto a la palma mediante su inversa . . . . .	87
4.2 Matriz del objeto1 con respecto a la palma . . . . .	87
4.3 Obtención de las nuevas coordenadas el objeto1 con respecto al mundo . . . . .	87

## PALABRAS CLAVE

*Keys: Sensores Inerciales, IMU, Captura de movimiento, Guante de datos, Realidad Virtual, Mpu6050*

## RESUMEN

Desde el nacimiento de la realidad virtual el humano ha buscado una manera natural de interactuar con estas escenas, y ha estado en una constante búsqueda de accesorios e intermediarios que puedan realizar esta tarea. En el mundo real, la forma más natural de interactuar con las cosas que nos rodean es a través de nuestras manos, por lo que es lógico pensar que estas sean las herramientas que nos ayuden con esta tarea.

Este trabajo presenta una propuesta de diseño para el desarrollo de un sistema para la captura de movimientos de la mano humana, con el objetivo de crear un guante de datos. El prototipo está compuesto por un conjunto de sensores inerciales colocados en la mano del usuario. Los sensores se utilizan para transmitir los movimientos de las falanges proximales y mediales de cada dedo en la mano. Para este prototipo, la flexión de la falange distal no se considera relevante, porque esta falange no puede ser controlada de manera regular por los humanos. La flexión del elemento distal se calcula con relación a la flexión mediana de la falange. El movimiento capturado por los sensores se transmite a través de una tarjeta Arduino <sup>TM</sup> a un ordenador, donde es posible hacer una mano virtual aplicando las transformaciones capturadas por los sensores. La mano virtual puede imitar la configuración de la mano real, incluyendo movimientos de abducción, aducción, flexión y extensión de los dedos.

Se realizó un diseño experimental para realizar una validación de captura de movimiento del sistema, obteniendo el desfase (drifiting) que se integra a las lecturas de los sensores, el frame rate y la resolución del sistema, en los cuales el primero se desfasa 1 grado aproximadamente cada 20 minutos, sin embargo eso se puede aminorar con el proceso de recalibración del guante, el frame rate fue aproximadamente de 101 hertz y la resolución mínima de 0.00001 grados. Se finaliza el trabajo con una tarea de manipulación de objetos en una escena virtual, la cual fue analizada estadísticamente con una tarea realizada con el guante CyberGlove II de CyberGlove Systems Ltd, líder en el sector comercial en cuanto a tareas de realidad virtual obteniendo resultados satisfactorios donde no existió diferencia significativa al completar la tarea.

## ABSTRACT

Since birth of virtual reality the human has sought a natural way to interact with these scenes, and has been in a constant search for accessories and intermediaries who can perform this task. In the real world, the most natural way to interact with the things that surround us is through our hands, so it is logical to think that these are the tools that help us with this task.

This work presents a design proposal for the development of a system for human hand motion capture system, aiming to create a data glove. This prototype will be composed of a set of inertial sensors placed in the user's hand. The sensors are used for transmitting the movements of the proximal, medial phalanges of each finger in the hand. For this prototype, the flexion of the distal phalanx is not relevant, because this phalanx cannot be normally controlled by the humans. The flexion of the distal element is calculated with relation to the medial phalanx flexion. The motion captured by the sensors are transmitted through an Arduino<sup>TM</sup> board to a computer, where it is possible to render a virtual hand applying the transformations captured from the sensors. The virtual hand can mimic the configuration of the real hand, including movements of abduction, adduction, flexion and extension of the fingers.

An experimental design was carried out to carry out a validation of the movement capture of the system, obtaining the drifting that is integrated into the sensor readings, the frame rate and the resolution of the system, in which the first one is offset by approximately 1 degree every 20 minutes, however that can be mitigated with the glove recalibration process, the frame rate was approximately 101 hertz and the minimum resolution was 0.00001 degrees. The work is finished with an object manipulation task in a virtual scene, which was statistically analyzed with a task done with CyberGlove II glove of CyberGlove Systems Ltd, leader in the commercial sector in terms of virtual reality tasks obtaining satisfactory results where there was no significant difference when completing the task.

*A mis padres,  
a mis hermanos  
y a mi esposa, ¡Con todo mi amor!*

— *Juan Augusto Campos Leal.*

## AGRADECIMIENTOS

Primero que nada quiero realmente agradecer a mi director de tesis Ulises Zaldívar Colado, por confiar en mi conocimiento limitado al inicio de este proceso, a la paciencia que me tuvo a lo largo de estos años y a la que aun me sigue teniendo, le agradezco el guiar, aconsejar y realmente adoptarme como un hijo, en la estancia a Versailles, Francia y recibirme prácticamente como en casa en cualquier parte del mundo, a su querida esposa Carolina Martínez Tirado y a sus hijos Leslie y Gael, muchas gracias a toda la familia.

También quiero agradecerle por todo lo que ha compartido conmigo durante la realización de este trabajo, desde el tiempo invertido hasta los conocimientos expresados en el proyecto y que se quedan plasmados en el estudiante.

Sin su guía este trabajo no sería posible.

A mi esposa Martha Isela Aguirre López, mi especial agradecimiento por su aliento y su asistencia material y moral permitido llevar a cabo esta tesis, por darme esa paciencia, toda esa motivación, por estar ahí cuando la necesitaba, por darme su apoyo cuando flaqueaba y ayudarme durante todo este proceso vivido.

También agradezco a mis padres, María Loreto Leal y Juan Isabel Campos Valderrama, mi hermana Paulina del Rosario Campos Leal y mi hermano Adrián Abdel Campos Leal.

Quiero agradecer a mis amigos y compañeros de generación, a Edgar Lao , Darío Osuna, Rosario Acuña, Óscar Castro, Ángel Sánchez, Paul Tamayo y a todos los que aun me faltan por nombrar, a cada uno que en determinado momento me ayudaron y me guiaron a lo largo de las clases y trabajos realizados, en el desarrollo de este proyecto y a todas esas discusiones fructíferas y para la ayuda que me dieron para obtener nuevos conocimientos.

Agradezco profundamente a mis maestros que me guiaron y me brindaron las herramientas para la realización de este trabajo, por tener paciencia durante las clases y por aconsejarme y criticar constructivamente este trabajo, a la Dra. Xiomara Zaldívar y al Dr Arturo Yee por estar allí desde el principio hasta el final y a todos los maestros de este posgrado.

Quiero agradecer a mis lectores críticos de esta tesis y por brindar ese aporte con sus comentarios, al maestro Zeus del valle y a la Dra. Dora Rodríguez.

Extiendo mi más sincero agradecimiento a los miembros del jurado que aceptaron juzgar mi trabajo.

Agradezco profundamente al Parque de Inovación tecnologica de esta universidad, en especial a su director MA José Ramón López Arellano por brindarme la oportunidad, animarme y motivarme para estudiar y terminar este proceso. A todos los compañeros del parque que de una u otra forma aportaron y pusieron su granito de arena para la realización de este trabajo.

Un agradecimiento especial para Samir Garbaya y su familia, que durante mi estancia en Francia me recibió como si me conociera de toda la vida, por su hospitalidad, por sus enseñanzas y todo ese conocimiento que aportó, por aceptar la realización de la instancia y la invitación a colaborar en su laboratorio y por aceptarme bajo su mando. Agradecer a Pierre Blazevic por la amabilidad y ser parte fundamental de la movilidad.

Agradezco a toda ese gente que se me permitió conocer durante el tiempo que duró mi movilidad en París, Francia, en especial a *LES ULIS*, a Redouane, J.F., a PAPIS, Abdul, a los dos David, y a todos los Eddy's, a Torean, a Twinkie, a Richard y Simón por permitirme hospedarme por un tiempo en su casa y a todos y cada uno del barrio de *le Parisien*, por recibirme como un amigo de toda la vida, por todo el cariño y todas las aventuras vividas.

Al consejo nacional de ciencia y tecnología por por apoyarme para la realización de estos estudios.

# CAPÍTULO 1

## INTRODUCCIÓN

En este capítulo se presenta una breve introducción al tema propuesto para el trabajo de investigación, a medida que se adentra en las secciones el lector puede identificar el problema a resolver y la debida justificación de este proyecto. Se explica brevemente como se propone una solución a dicha problemática y se marcan los objetivos particulares y generales que guiarán la metodología para el desarrollo de este trabajo de tesis. Al final del capítulo se formula la hipótesis y se explica brevemente la organización de la tesis y como se conforma cada capítulo.

### 1.1. Introducción

Nuestra primera conexión física con el mundo son nuestras manos, pues realizamos tareas cotidianas con ellas, desde tareas simples hasta tareas muy complejas sirviendo como dos herramientas significativas para las actividades de la vida diaria [Sturman and Zeltzer, 1994].

La mano humana es quizás la herramienta más útil y diversa que utilizamos para interactuar con el entorno que nos rodea. Es capaz tanto de hacer un potente agarre como empujar, tirar, o girar objetos, hasta girar o manejar objetos pequeños que requieran un agarre de precisión. Junto con el habla y la expresión facial, los gestos de la mano se utilizan como uno de los principales métodos de comunicación con las personas y para la manipulación de las cosas de nuestro alrededor. Dado que nuestras manos son una de las herramientas más útiles para interactuar con el mundo real, se han hecho muchas afirmaciones sobre el valor de usar las manos directamente en un entorno virtual como una interfaz en este entorno [Kessler et al., 1995].

Solo nos damos cuenta cuando sufrimos algún tipo de deficiencia ya sea motora o física, lo complicado que puede ser llevar a cabo las distintas actividades habituales. Para ello se han realizado diversos estudios en los que se pueden medir el movimiento de la mano, una herramienta que ha ayudado en éstos trabajos ha sido la computadora.

Pero la interacción que tenemos con las computadoras está limitada por intermediarios que tienen nula destreza o naturalidad como los dispositivos de entrada y salida, el mouse, el teclado, etc., y en un esfuerzo por cambiar esta situación se han diseñado, construido y estudiado maneras de “leer” (capturar) el movimiento de las manos. Desde los primeros prototipos de guantes que estaban equipados con un número limitado de sensores, tenían cableado complejo, eran engorrosos y fueron desarrollados para servir a aplicaciones muy específicas, se utilizaron brevemente, y nunca se comercializaron, hasta la gran diversidad de sistemas de capturas de movimientos.

Los guantes de datos de captura de movimiento utilizan con frecuencia dispositivos de entrada que interpretan los gestos de la mano humana para aplicaciones como la realidad virtual y la interacción hombre-computadora. El desarrollo de guantes de datos (captura de movimientos de los dedos de la mano) ha sido un gran paso en este aspecto, sin embargo, los guantes comerciales de datos de movimiento son demasiado caros para el consumo, y esto ha limitado su popularidad [Han, 2010].

Hoy en día los sistemas de captura de movimientos tienen mucha importancia en aplicaciones de realidad virtual, lo cuales son utilizados en medicina, industria automotriz, películas, videojuegos, etc. Existen diversos tipos de sistemas de captura de movimiento: mecánicos, ópticos, magnéticos, inerciales y en el caso de los guantes también existen de variación de resistencias [Abe et al., 1996]. Uno de los problemas de los dispositivos de captura de movimientos es el alto costo de los mismos lo que dificulta su adquisición, ya que pueden variar desde los 4,000 hasta los 200,000 dólares [Carrillo, 2007].

Este trabajo es enfocado a los sistemas de captura de movimientos de los dedos

de la mano, los cuales pueden llegar a costar cerca de 20,000 dólares, como es el caso del Cyberglove<sup>TM</sup>, de Cyberglove Systems Ltd que tiene como principal uso la manipulación de objetos virtuales [Borst and Indugula, 2005] [Borst and Indugula, 2006] [Zaldívar Colado, 2009] y el ensamble virtual [Zaldívar Colado, 2009] [Zaldívar-Colado et al., 2006].

En el campo de la medicina se han realizado diversos estudios para la medición precisa de la orientación de los segmentos del cuerpo humano usando una unidad de medición inercial (IMU por sus siglas en inglés: Inertial Measurement Unit). Algunos estudios han sido probados para los movimientos de la pelvis, el tronco y el antebrazo [Luinge and Veltink, 2005].

Otros trabajos también han estimado la orientación de los segmentos del cuerpo humano mediante la fusión de giroscopio, acelerómetro y magnetómetro, incluso cerca de materiales ferromagnéticos que perturban el campo magnético local terrestre y por lo tanto la estimación de dicha orientación. Para esto se ha aplicado un filtro en la lectura de los datos y se han comparado con sistemas ópticos de referencia, como VICON. Las estimaciones de orientación resultantes han sido precisas y sin desviaciones [Roetenberg et al., 2005].

Debido a lo mencionado anteriormente sobre la mano humana como una de las herramientas más útiles y diversas que utilizamos para interactuar con el ambiente que nos rodea [Sturman and Zeltzer, 1994], se han realizado diversos trabajos para estudiar el movimiento de los elementos de la mano enfocados a la rehabilitación de los pacientes, aun dada la importancia de la interacción con el mundo real, la integración de estas “herramientas” de interacción en la realidad virtual se ha estudiado desde hace muchos años [Kessler et al., 1995].

En este campo los guantes de datos se usan a menudo como dispositivos de entrada que interpretan los gestos de la mano humana para la interacción hombre-computadora en aplicaciones de realidad virtual. Se han desarrollado diversos trabajos en guantes de datos, como es el caso de Cyberglove, basado en la variación de resis-

tencias [Kessler et al., 1995][Fuchs and Moreau, 2006], el cual mide todos movimiento de los dedos de la mano (flexión, extensión, aducción y abducción) y es posible representarlos en la escena virtual. Otros trabajos han sido desarrollados basados en sensores inerciales, trabajando con la recogida de estos datos en pacientes lesionados y su proceso de rehabilitación, con lo que ha sido posible examinar los movimientos de la mano del paciente [Lin et al., 2014] [Hsiao et al., 2015].

También se han presentado propuestas basadas en la combinación de sensores inerciales y variación de resistencias que representan los movimientos de la mano del usuario en un entorno virtual, aunque el propósito del trabajo ha sido ayudar al paciente con su rehabilitación [Tavares et al., 2016], sin embargo carece de un modelo cinético que valide sus movimientos.

El sistema de captura de movimiento propuesto en este trabajo se centra en la captura de los movimientos de los dedos, que pueden ser utilizados como una alternativa a los dispositivos de captura de movimiento que se utilizan en la manipulación de objetos virtuales [Borst and Indugula, 2005] [Borst and Indugula, 2006] [Zaldívar Colado, 2009] y el ensamble virtual [Zaldívar Colado, 2009] [Zaldívar-Colado et al., 2006].

Debido a su bajo consumo de energía, bajo volumen, bajo costo y dado que los sensores inerciales han sido ampliamente utilizados en los últimos años en la fabricación de automóviles, teléfonos inteligentes y equipos, etc., la investigación ha dado un giro a los sistemas basados en sensores inerciales [Fei et al., 2014] [Tung and Van Thanh, 2015].

Por lo anterior, el objetivo es generar un dispositivo de captura de movimientos para las articulaciones de mano humana basado en IMU, que se pueda desarrollar a un bajo costo (de menos de \$ 300 usd). El dispositivo propuesto, basado en sensores inerciales evitará el proceso de calibración complicado y robusto que se presenta en otros guantes, como es el caso del CyberGlove<sup>TM</sup>[Kessler et al., 1995], mitigando éste con un proceso de calibración más sencillo. Además, este dispositivo evitará el desgaste y

el sobrecalentamiento que está presente en otros dispositivos cuando se trabaja durante un tiempo considerable [Borst and Indugula, 2005] [Borst and Indugula, 2006] [Zaldívar Colado, 2009].

## **1.2. Descripción del problema**

## **1.3. Planteamiento del problema y justificación**

### **1.3.1. Justificación**

Un guante para la captura de datos basado en sensores inerciales puede adoptar las ventajas que ya tienen los tipos de guantes de datos actualmente y evitar las principales desventajas.

Existen sistemas basados en sensores inerciales para captura de movimientos de distintas partes del cuerpo humano, sin embargo no se han implementado en la captura de movimientos de los dedos de la mano en los ambientes virtuales.

Un guante de datos o un sistema de captura de movimientos de los dedos de la mano basado en sensores inerciales, permitiría capturar movimientos de flexión, aducción y abducción de los dedos de la mano. Además, extendiendo el sistema de sensores al antebrazo y brazo del usuario se puede determinar las coordenadas geométricas de la palma de la mano en el espacio, esto se puede añadir fácilmente debido a la flexibilidad del sistema y trabajar con futuras versiones del prototipo.

### **1.3.2. Planteamiento del problema**

Actualmente existen diversos guantes de datos que nos permiten trabajar en un ambiente virtual los cuales brindan ventajas aceptables para la captura, pero también presentan desventajas considerables. La mayoría ofrecen buena precisión y buena resolución pero no todos ofrecen capturar movimientos de flexión, aducción y abducción.

Los complejos procesos de calibración, el desgaste de sensores o lo delicado que puede ser manejar un guante de datos pueden llegar a ser desventajas considerables a la hora de pensar en adquirir uno de estos productos, que aunado al precio comercial hace considerar la adquisición de un dispositivo como este.

El alto costo de los guantes de datos comerciales es otro factor a considerar, ya que debido a esto puede ser factor para algunos estudiantes que quisieran incursionar en la realidad virtual no lo realicen por no tener el equipo para realizar un proyecto.

Los guantes de datos permiten capturas de movimientos de los dedos de la mano pero requieren de algún dispositivo adicional de captura de movimientos para determinar la posición de la palma de la mano en el espacio.

Entre algunos tipos de guantes de datos se encuentran:

- Los ópticos: que están basados en un sistema de cámaras que observa al objetivo a través de marcadores.
- Los guantes basados en variación resistencia: basados en sensores que interpolan los valores dependiendo de la flexión del elemento.
- Los guantes magnéticos: basados en sensores magnéticos que emiten las señales entre un emisor y un receptor.
- Los guantes mecánicos: exoesqueletos articulados colocados en los elementos de interés que miden directamente los ángulos de flexión.

### **1.3.3. Solución propuesta**

El objetivo de este trabajo es diseñar e implementar un dispositivo de captura de movimientos de las articulaciones de la mano humana que mediante un sistema de sensores inerciales mida el movimiento para determinar la posición y orientación en el espacio de estas falanges, el cual puede desarrollarse a un bajo costo (300 dólares aproximadamente).

Nuestra propuesta es utilizar sensores IMU en las falanges de los dedos de la mano (proximal y medial, distal no ya que su movimiento no es relevante para este proyecto) y uno más en la palma de la mano con lo cual determinaremos las posiciones de los elementos en el espacio. Para determinar la posición de la palma se utilizará un dispositivo POLHEMUS FasTrak. El Polhemus Fastrak usa un emisor para producir un campo magnético y receptores para identificar su posición en relación con el emisor. Cada punto seguido tiene un cable que regresa al rastreador. El Fastrak tiene límites en sus puntos de seguimiento, actualización y rango, y es susceptible a la distorsión magnética, pero proporciona datos de a un costo menor [Wingrave et al., 2010].

Utilizar los sensores inerciales para la captura de movimientos evitará el proceso de calibración que se presenta en otros guantes como el Cyberglove<sup>TM</sup>, y evitará también el desgaste y sobrecalentamiento que presenta al trabajar con él, sobre un tiempo considerable [Borst and Indugula, 2005] [Borst and Indugula, 2006] [Zaldívar Colado, 2009]. El proceso de calibración desarrollado para el guante prototipo se realizó de manera fácil y sencilla, permitiendo al usuario realizar el proceso con presionar una tecla, incluso el proceso de recalibración se realiza de la misma manera. En cuanto al desgaste se puede mencionar que debido a la naturaleza que los sensores poseen, no se desgastarán como sí lo hacen las resistencias con el uso continuo.

Todos los sensores se comunican a una placa base que es la que recibe los valores emitidos por cada sensor a través de un protocolo de comunicación, la cual envía los valores por puerto serial hacia la aplicación en C++, que se encarga de determinar la posición y orientación en el espacio de cada elemento de la mano en la escena virtual a través de un modelo cinemático directo.

En el procedimiento experimental se realizan diferentes experimentos para darle la validación necesaria a este proyecto. Debido a la naturaleza el sensor, por ser un mecanismo electrónico, sufre de ruido e incluso la temperatura puede hacer que después de un tiempo determinado, los valores sufran de un desfase o deriva (drift)

con el tiempo, para esto se midió dicho barrido para graficar la curva de valores con respecto al tiempo y mitigar este problema.

Se realizó un procedimiento con el fin de conocer la frecuencia de datos (Hertz) que son entregados a la aplicación y si pudiera o no influir en la escena virtual. Para validar la resolución que se obtiene de los sensores fue necesario desarrollar otro procedimiento, en el cual fueron tomados los valores de los sensores y se calculó su resolución mínima. Ya que se trabajó para mejorar la precisión de los sensores fue necesario realizar un procedimiento que comparará la precisión que los sensores entregan normalmente contra la precisión que entregan después de aplicar el método de mejora. Por último, se realizó un experimento de manipulación de objetos virtual y mediante un análisis estadístico se evaluó la diferencia entre el prototipo desarrollado en este proyecto y un producto ya validado en el sector de la realidad virtual, encontrando resultados altamente satisfactorios.

#### **1.3.4. Modelado del ambiente virtual**

La solución que se propone en este trabajo para realizar el modelado del ambiente tridimensional, se basa en el uso del lenguaje de programación C++ bajo la plataforma Visual Studio C++ 2010 Express Edition de Microsoft y utilizando la librería de renderizado gráfico OpenGL (Open Graphics Library) de libre distribución. De la misma manera se utilizan las librerías de PQP ( Proximity Query Package) y VHT (VirtualHandToolkit), la primera para la detección de colisiones de uno o más objetos en un ambiente virtual y la segunda para la realizar la representación de la mano tridimensional, cabe destacar que esta última es proporcionada por CyberGlove Systems bajo su licencia.

El proyecto se complementa con el módulo desarrollado en el IDE proporcionado por Arduino, este módulo se instala en la placa y establece la comunicación entre ésta y los sensores inerciales, recibe los valores emitidos por cada sensor a través de un

protocolo de comunicación y después son enviados por puerto serial hacia la aplicación en C++, la cual se encarga de representar la escena virtual.

En la figura 3.15 se presenta la arquitectura del sistema.

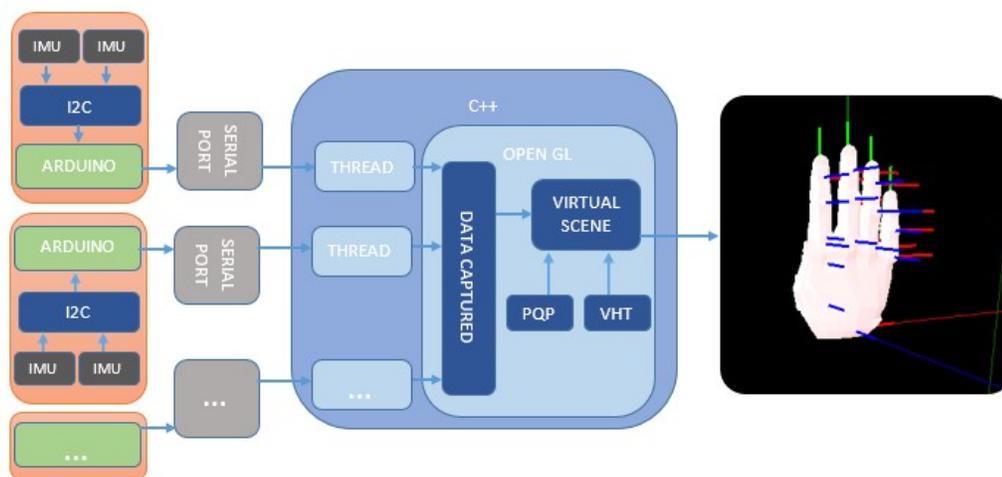


FIGURA 1.1. Arquitectura de sistema

La aplicación fue desarrollada y probada en una computadora con las siguientes características: una computadora con Windows 10 pro de 64 bits con una CPU Intel <sup>®</sup> Core <sup>™</sup> i7-270QM CPU a 2.20 GHz (8 CPU), 4 GB de memoria RAM DDR 3 y 500 GB disco duro, además de una tarjeta de video NVIDIA GeForce GT 540M de 2GB dedicada.

### 1.3.5. Medición de los datos

Se utilizarán unidades de mediciones inerciales (IMU por sus siglas en inglés Inertial Measurement Unit), específicamente el sensor InvenSense MPU-6050 que contiene un acelerómetro MEMS y un giroscopio MEMS en un solo chip (este modelo no contiene magnetómetro), el cual es muy preciso, ya que contiene un hardware de conversión de 16 bits analógicos a digital para cada eje. Para ello se capturan las coordenadas x, y, z del plano 3D al mismo tiempo. Dicho sensor utiliza el I2C-bus

para interconectar con Arduino. Como placa base se utilizará un Arduino MEGA 2560, el cual obtendrá los datos y los enviará a la pc a través de puerto serial.

## **1.4. Objetivos**

### **1.4.1. Objetivo general**

El objetivo del trabajo de investigación es diseñar, desarrollar y evaluar un dispositivo de captura de movimientos basado en sensores inerciales para la mano humana en aplicaciones de realidad virtual.

### **1.4.2. Objetivos específicos**

- Desarrollar un guante con una alta precisión y resolución, que sea comparable (estadísticamente similar) a los guantes de datos más utilizados mundialmente.
- Realizar un estudio comparativo sobre manipulación de objetos con otro guante de datos de captura de movimientos (experimental o comercial)
- Mejorar la precisión estándar que ofrecen actualmente los sensores para aprovechar al máximo su precisión.
- Realizar un análisis de los resultados obtenidos, después del estudio comparativo evaluar la precisión y resolución con respecto a los otros tipos de guantes.
- Validar la aplicabilidad mediante un caso de estudio (una tarea de manipulación virtual)

## **1.5. Hipótesis**

El desarrollo de un guante de datos de bajo costo, basado en sensores inerciales, mantendrá el desempeño del usuario en tareas de manipulación de objetos en un am-

biente virtual, comparado al obtenido con el guante de datos comercial más utilizado mundialmente; teniendo que el desempeño del usuario será medido en términos de:

- Tiempo de Tarea Completa (Task completion time)
- Tiempo de localización del objeto (Locating time)
- Tiempo de agarrado del objeto (Grasping time)
- Error en la distancia (Distance Error)

## 1.6. Organización de la tesis

A continuación se presenta la organización de este documento de tesis.

En el capítulo II se detalla el marco teórico y el estado del arte. Este capítulo es el sustento teórico de la investigación, se presentan las definiciones básicas que permiten contextualizar el desarrollo del presente trabajo. También se presenta en el estado del arte un recorrido por los trabajos en realidad virtual, analizando fortalezas y debilidades para obtener la información necesaria para la realización de este trabajo.

El capítulo III se centra en la metodología, donde se explican los pasos a seguir para probar la hipótesis y validar así la investigación. La metodología se presenta en etapas de desarrollo del sistema. En cada etapa se explica el sistema desarrollado hasta ese punto, su estructura y las tecnologías utilizadas. Se explican las arquitecturas de hardware y software empleadas, y posteriormente la unificación de ambas para implementar el sistema. En algunos casos, si es que aplica, se presentan pruebas y experimentos de la etapa. En los casos que aplica también se presentan las conclusiones y los trabajos a desarrollar en la etapa siguiente.

El capítulo IV presenta la experimentación y resultados, donde se presentan pruebas y experimentos de las etapas. Se complementa con análisis estadísticos para validar la investigación.

El capítulo V presenta las conclusiones finales del trabajo de tesis. También se proponen trabajos a futuro para extender la actual investigación. Posteriormente, se muestran los anexos, como son: publicaciones realizadas, exposiciones y participaciones en congresos.

Finalmente, se presentan las referencias bibliográficas utilizadas en este trabajo.

## CAPÍTULO 2

### MARCO TEÓRICO Y ESTADO DEL ARTE

En este capítulo se describen conceptos para entrar en contexto con el trabajo de investigación y ayuda al entendimiento del contenido de este documento. Se describen brevemente, primero dando un recorrido del surgimiento de algunos de ellos y como fueron apareciendo otros a lo largo de la historia. Al mismo tiempo, se ofrece un recorrido sobre algunos trabajos relevantes sobre el tema, surgimiento de dispositivos y conceptos e innovaciones que surgen con trabajos de investigación y que han guiado a la elaboración de este trabajo de investigación.

#### **2.1. Ambientes virtuales**

Los Ambientes Virtuales (AV) son una simulación tridimensional generada por computadora la cual busca que el usuario tenga la sensación de explorar y manipular un mundo tridimensional dentro de una pantalla bidimensional. Los AV's surgen en el área de los videojuegos pero se han extendido en campos como la medicina, simulaciones y se están empleando como herramientas muy efectivas en muchos de los procesos relacionados al diseño y manufactura de productos, en donde el diseño se ha convertido en uno de las etapas más importantes dentro del ciclo de manufactura de productos y produce una influencia en todas las etapas siguientes del desarrollo de un producto [Antonya and Talaba, 2007].

Mediante un proceso conocido como renderizado, los AV's se basan en la producción de gráficos por computadora semejantes al mundo real y generados en tiempo real, que permitan la interacción con objetos tridimensionales como el desplazarse, moverse o tomar objetos.

Los ambientes virtuales son una herramienta económica con el que se pueden simular equipos costosos que no se encuentren a nuestro alcance, pero que se desea interactuar con ellos. Algunas de las ventajas que la simulación tridimensional nos ofrece son, permitir observar el comportamiento de los prototipos en el mundo virtual sin tener que construirlos físicamente, desde realizar cambios rápidos al modelo del prototipo hasta obtener los resultados esperados ante algún evento.

Las primeras ideas de los ambientes virtuales surgen con las ideas de Huxley y Weinbaum en los años 1930, detallando las ideas de oír y tocar los objetos de una película en una pantalla, ¿Sería real o un sueño? Ésa era la pregunta.

Las ideas se mantuvieron hasta los años 1950 cuando Heiling desarrolló y patentó una máquina que ofrecía una experiencia de rodar en bicicleta de manera virtual, Sensorama fue llamada. Este fue el primer paso en un desarrollo real en la realidad virtual [Mihelj et al., 2014].

El siguiente paso fue alrededor de 1960, el que se puede considerar como el primer casco montado sobre la cabeza, Sutherland lo predijo en el 1965 y en 1968 lo conectó a un ambiente virtual, fue llamado Philco HMD (Head Mounted Display ) el cual con el tiempo lo llevo a desarrollar Sword of Damocles que consistía en un par de lentes que daba una ilusión de una visión en tres dimensiones [Sutherland, 1965]. La figura 2.1 muestra el HMD.

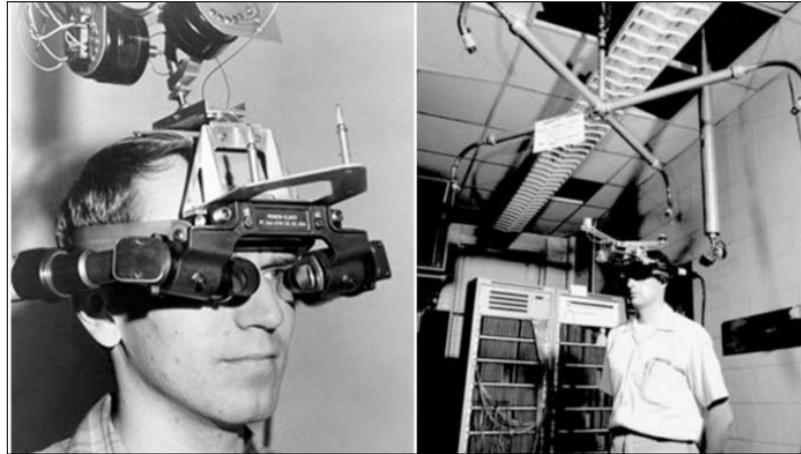
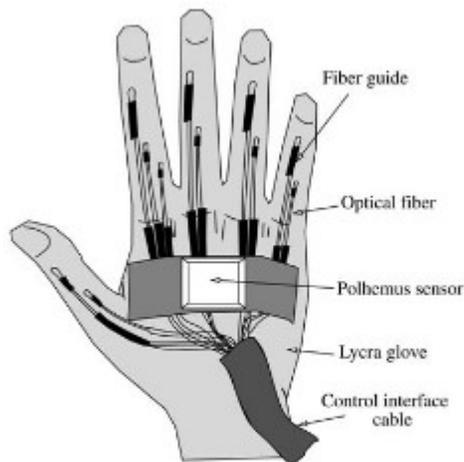


FIGURA 2.1. Philco HMD (Head Mounted Display).

Los primeros ambientes que reaccionaron a las acciones del usuario fueron desarrollados alrededor de 1970 por Myron Krueger con los desarrollos de los sistemas Grope I-III, que consistía en un sistema de cámaras y sensores que permitieron al usuario mover las moléculas en la pantalla usando una interfaz háptica especial, así como sentir las fuerzas entre las moléculas usando retroalimentación de fuerza [Mihelj et al., 2014].

En 1980 se dio el siguiente paso para el reconocimiento táctil en los ambientes virtuales, la aparición de los guantes de datos, como el guante Sayre y el VLP Dataglove, que a través de sensores que detectaban los movimientos de la mano se encontraba un método simple y barato del reconocimiento del movimiento humano [Mihelj et al., 2014]. La figura 2.2 muestra el VPL.



### The VPL DataGlove (1987)

FIGURA 2.2. El VLP Dataglove (1987).

Ya en los años noventa los ambientes virtuales tuvieron avances notables, a pesar de haber caído en un bache, tal vez el producto más famoso de realidad virtual fue CAVE (Cave Automatic Virtual Environment), una habitación cuyas paredes consisten en pantallas que muestran un espacio de ambiente virtual. La CAVE se ha vuelto muy popular y es utilizada por universidades de todo el mundo [Mihelj et al., 2014]. La figura 2.3 muestra el sistema CAVE.



FIGURA 2.3. Sistema CAVE.

Debido al alto precio, las versiones para el hogar aún no están disponibles, pero las mismas tecnologías se usan en parques de atracciones y cines como el cine en 3D.

## 2.2. Manos virtuales

Para manipular objetos en un ambiente virtual se pueden utilizar diferentes herramientas, pero la necesidad de manipular de la forma más natural y que se asemeje al ambiente real nos ha llevado a trabajar con una mano virtual, que esta se comporte o se asemeje físicamente a una mano real es de gran ayuda sobre todo en aquellas aplicaciones de entrenamiento para las tareas de ensamble [Borst and Indugula, 2005].

Las manos virtuales surgen con la aparición de los guantes de datos, ya que al medir los movimientos de la mano, fue natural la creación de modelos que representaran estas mediciones en los ambientes virtuales.

En 1982 Thomas G. Zimmerman patentó un sensor flexible óptico montado a un guante el cual podía medir el doblamiento de los dedos, en él incluyó manos virtuales animadas que reproducían las posturas de la mano de un operador [Sturman and Zeltzer, 1994].

En sus inicios las manos virtuales eran representadas basando su diseño en modelos geométricos simples, en el caso del trabajo de Iwata en 1990, la mano se componía de 12 objetos, poseía 9 grados de libertad, en su propuesta plantea un método de fuerza-retroalimentación, en el que se manipulaban objetos en un ambiente virtual [Iwata, 1990]. La figura 2.4 muestra la mano virtual.

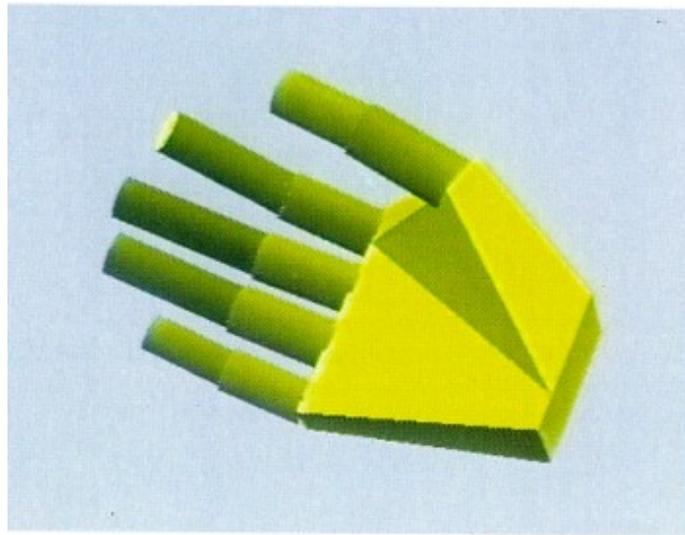


FIGURA 2.4. Posturas de la mano para el agarrado de un objeto.

[Jayaram et al., 1997] propone en su trabajo un diseño de un entorno virtual, en él, se logra la captura de los movimientos de las manos por medio de trackers electromagnéticos, y el movimiento de los dedos y muñeca de las manos del usuario por medio de guantes de datos. Estos dispositivos, dispuestos de esta manera, mejoran la interacción natural del usuario con la escena creada.

[Borst and Indugula, 2006] propusieron un diseño de una mano virtual con un alto grado de realismo y simulación, la cual se componía de 16 elementos, el modelo se componía por dos diseños de manos, una forma sólida y otra representada por una malla, la última interpretaría el objeto al momento del agarrado, al salir de la zona de colisión la mano de malla seguiría a la mano sólida. Esta técnica tuvo la ventaja de ser rápida en sus cálculos y solo era necesario las posiciones de los puntos de los

objetos en el espacio tridimensional y las uniones entre ellos para formar los puntos. La figura 2.5 representa la mano virtual.

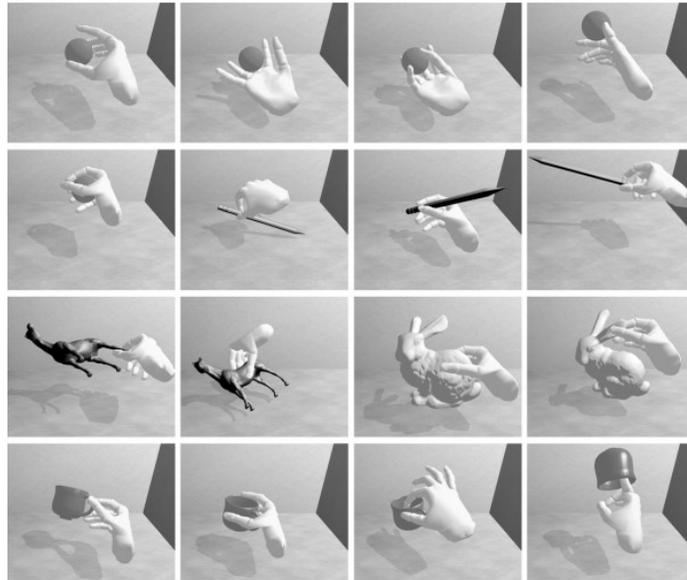


FIGURA 2.5. Mano virtual de Borst e Indugula.

[Ullmann and Sauer, 2015] presentó un trabajo que contenía varios puntos de detección para una posible colisión con algún objeto dentro de un ambiente virtual, su propuesta no tenía retroalimentación háptica, sin embargo se guiaba a través de la visión, él proponía trabajar con manos virtuales a través de guantes de datos.

[Garbaya and Zaldivar-Colado, 2009] presenta una mano virtual compuesta de 16 modelos geométricos su sistema Virtual Environment for Design and Assembly Planning II (VEDAP-II) el cual fue desarrollado para el ensamble virtual de piezas mecánicas, y utiliza un guante de tipo Cyberglove<sup>TM</sup> que animan la mano virtual en la escena. Este sistema fue utilizado por [Zaldivar Colado, 2009] para representar la mano virtual.

[Zaldivar-Colado et al., 2010] [Lizárraga Reyes, 2010] presentan un trabajo donde propone una técnica de interacción con un ambiente de ensamble virtual con el cual puede agarrar y manipular un objeto tridimensional a través de movimientos

interpolados entre una postura inicial y final de una mano. Se utilizó un dispositivo de sensación háptica para desplazar la mano en el ambiente virtual, pero no para además de detectar proximidad y colisiones entre los dedos de la mano virtual y los objetos tridimensionales. La figura 2.6 se observa la mano virtual en su estado normal y su estado de agarrado.

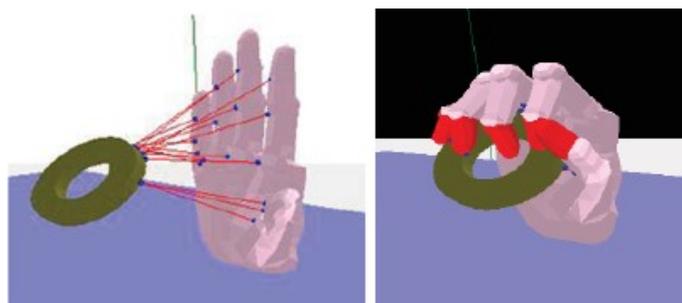


FIGURA 2.6. Posturas de la mano para el agarrado de un objeto.

Pero no siempre se han utilizado manos virtuales, como [Kitamura et al., 2009] que utilizó los palillos chinos como herramienta de manipulación, debido a su amplia utilización en Asia y oriente y por tener una forma muy simple, su gran facilidad de representar en un AV fue uno de los motivos, sin embargo no son muy fáciles de manejar para la mayoría de las personas, siendo más natural el uso de las manos. La figura 2.7 muestra los palillos utilizados por Kitamura.

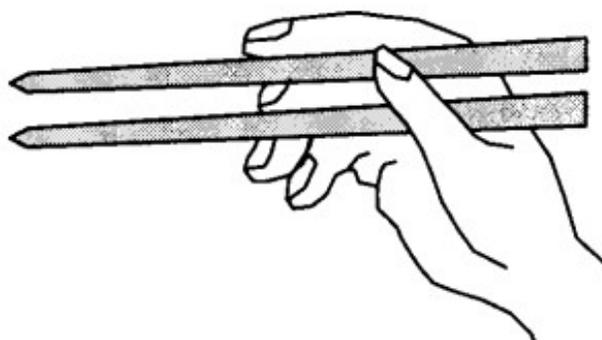


FIGURA 2.7. Palillos como herramientas virtuales.

Junto con CyberGlove apareció el modelo que ofrecía hasta 22 grados de libertad que permitió representar la mayoría de los movimientos de la mano [Kessler et al., 1995]. Este modelo era proporcionado por el software para el manejo de CyberGlove.

En el trabajo realizado por [Li et al., 2016] utiliza el guante CyberGlove y para apoyar el hardware este módulo implementa las interfaces de hardware con la mano virtual, además propone un enfoque de agarre virtual mecánico basado en guante de datos, además analiza la estructura de la mano y los posibles movimientos con el modelo cinemático directo Denavit-Hartenberg [Denavit and Hartenberg, 1964]. La figura 2.8 muestra los palillos utilizados por Li.

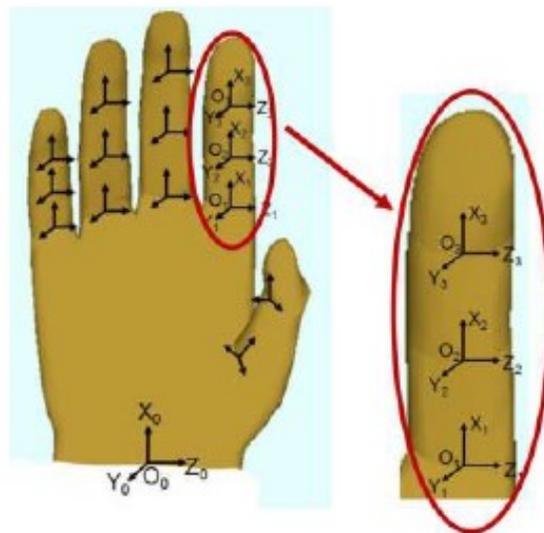


FIGURA 2.8. Cinemática en la mano virtual de Li.

### 2.3. Sistema de captura de movimientos

El objetivo de este apartado es presentar las interfaces específicas para localizar y determinar el movimiento del cuerpo o parte de él, el propósito de la interfaz no es determinar en general los movimientos precisos, pero sí permitir una locomoción del usuario a través del entorno virtual.

Las interfaces específicas de localización son interfaces motrices, que miden en tiempo real la posición afectada del cuerpo, permitiendo el seguimiento de su trayectoria. Estos dispositivos son necesarios en las aplicaciones de realidad virtual de inmersión corporal.

Estas interfaces específicas mejoran la interacción de la materia en el mundo virtual, ya que puede actuar de manera bastante natural: por movimientos gestuales, si es posible, idénticas a las que se realiza en el mundo real. Por lo tanto, aumenta la calidad de la inmersión.

Las interfaces de localización del cuerpo son utilizadas para detectar los movimientos de todo el cuerpo y a menudo también determinar la configuración del esqueleto en todo instante. En el lenguaje habitual, por lo general se emplea el término captura de movimiento “motion capture” [Fuchs and Moreau, 2006].

En el mundo real, el agarre de un objeto con la mano es un fenómeno que implica muchos procesos complejos al nivel sensorimotor y cognitivo, incluso si es subconsciente.

Los sistemas Mocap o “motion capture” disponibles comercialmente en la actualidad se pueden clasificar en cinco grupos: sistemas ópticos, sistemas magnéticos, sistemas mecánicos, sistemas de ultrasonidos e inerciales, aunque los últimos dos son muy poco utilizados en el entorno de entretenimiento. Cada tipo tiene fortalezas y debilidades.

Un sistema óptico típico consiste en 4 a 32 cámaras y una computadora que controla las cámaras. Con la mayoría de los sistemas ópticos, los sujetos de captura usan marcadores, donde los marcadores son reflexivos (pasivos) o emisores de luz (activos). Los marcadores pasivos están hechos de materiales reflectantes y sus formas son esféricas, semi-esféricas o circulares. Las formas y los tamaños de los marcadores dependen de las resoluciones de la cámara y de los sujetos de captura (por ejemplo, se utilizan marcadores más pequeños para capturas faciales y manuales) [Kitagawa and Windsor, 2012].

Ventajas de los sistemas ópticos:

- Los datos ópticos son precisos.
- La velocidad de captura es alta.
- Múltiples sujetos pueden ser capturados simultáneamente.
- Se puede usar un gran número de marcadores.
- Las configuraciones de marcadores se pueden cambiar fácilmente, dependiendo de las metas del proyecto.
- Los sujetos de captura de los sistemas ópticos pueden moverse libremente en un volumen de captura.
- El volumen de captura puede ser mayor que la mayoría de los otros sistemas.
- Se pueden generar datos esqueléticos.

Desventajas de los sistemas ópticos:

- Se requiere un post-procesamiento extensivo.
- Los datos de rotación deben ser calculados a partir de datos de posición en el post-procesamiento.
- Los marcadores pueden ser ocluidos por sujetos de captura o accesorios, lo que resulta en la pérdida de datos.
- La iluminación necesita ser controlada para la mayoría de los sistemas ópticos, especialmente los sistemas pasivos.
- La retroalimentación visual en tiempo real se limita a las cifras de palo.
- El hardware es a menudo más caro que otros tipos de equipos de mocap.

La figura 2.9 muestra un ejemplo de un sistema de captura de movimientos óptico.



FIGURA 2.9. Ejemplo de sistemas de captura de movimientos óptico.

Muchos trabajos se han desarrollado con sistemas de captura de movimientos ópticos; usan cámaras infrarrojas para localizar marcadores reflexivos colocados en el cuerpo humano, ejemplos de estos sistemas son Vicon<sup>TM</sup>, Optitrack<sup>TM</sup>, MotionAnalysis<sup>TM</sup>.

Como lo hace [Zhao et al., 2004] donde propone la función de similitud del movimiento humanoide en base a la captura de movimiento del cuerpo humano y formula restricciones cinemáticas debido a que la cinemática y dinámica del humanoide difieren a las de un actor humano.

[Nakaoka et al., 2003] realizó un sistema para que un robot imite las danzas humanas extrayendo posturas esenciales en movimientos de brazo y primitivas de paso en movimientos de piernas. Luego, los ángulos de articulación se modifican para satisfacer las restricciones mecánicas del robot.

Los sistemas de captura de movimientos magnéticos (electromagnéticos) a veces también llamados trackers magnéticos, se derivaron de los sensores colocados en el casco de un piloto de aviones militares para rastrear la posición de la cabeza del piloto y la

orientación para la pantalla montada en el casco. Con un sistema de mocap magnético, se colocan de 12 a 20 sensores de seguimiento en un sujeto de captura para medir la relación espacial con un transmisor magnético. Los sensores de seguimiento transmiten sus traducciones y orientaciones. Por lo tanto, no se requiere post-procesamiento para calcular las rotaciones. Este hecho permite que los sistemas magnéticos se utilicen para aplicaciones en tiempo real [Kitagawa and Windsor, 2012].

Ventajas de los sistemas magnéticos:

- Posición y orientación están disponibles sin post-procesamiento.
- La retroalimentación en tiempo real permite aplicaciones en tiempo real.
- Los sensores de seguimiento no son ocultados por objetos no metálicos.
- Se pueden capturar múltiples ejecutantes simultáneamente con varias configuraciones.
- Los sistemas magnéticos son menos costosos que los sistemas ópticos.

Desventajas de sistemas magnéticos:

- Los sensores de seguimiento son propensos a interferencias magnéticas y eléctricas.
- El cableado y las baterías para los sensores de seguimiento pueden limitar los movimientos de los sujetos de captura.
- Los sensores magnéticos tienen una tasa de muestreo menor que la mayoría de los sistemas ópticos.
- Los datos magnéticos tienden a ser ruidosos. Las configuraciones de los sensores de seguimiento son difíciles de cambiar. Los volúmenes de captura de los sistemas magnéticos son normalmente más pequeños que los sistemas ópticos.

La figura 2.10 muestra un ejemplo de un sistema de captura de movimientos electromagnético.



FIGURA 2.10. Sistema de captura de movimientos electromagnéticos.

Los sistemas de captura de movimientos mecánicos (exoesqueléticos) miden directamente los ángulos articulares de un sujeto de captura que lleva un dispositivo articulado que consta de varillas rectas y potenciómetros. Las barras rectas están conectadas con potenciómetros en las articulaciones del cuerpo, diseñadas para medir ángulos de articulación a medida que se mueve el sujeto de captura. El dispositivo se parece a un exo-esqueleto. Otros tipos de sistemas mecánicos incluyen guantes de datos y armaduras digitales [Kitagawa and Windsor, 2012].

Ventajas de los sistemas magnéticos:

- Tiempo real.
- Relativamente barato.
- Sin oclusión.
- No hay interferencias magnéticas o eléctricas.

- Muy portátil.
- Amplio rango de captura.

Desventajas de sistemas magnéticos:

- No hay traducciones globales.
- Restricciones en el movimiento del sujeto de captura.
- Quebradizo.
- Configuración fija de sensores.
- Baja frecuencia de muestreo.

La figura 2.11 muestra un ejemplo de un sistema de captura de movimientos exoesquelético.

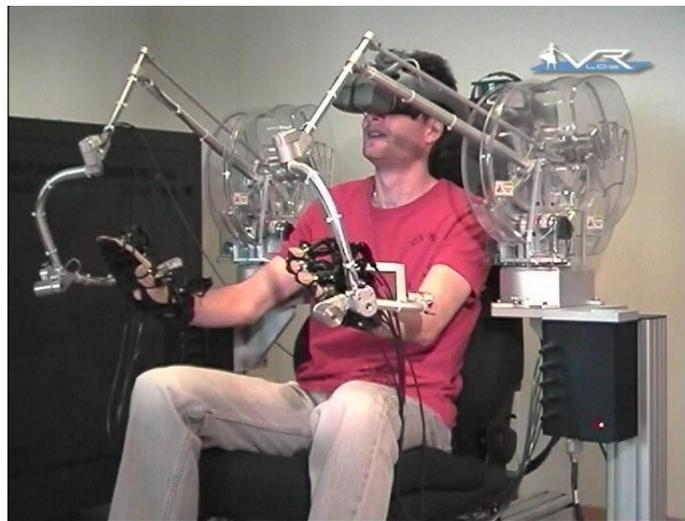


FIGURA 2.11. Sistema de captura de movimientos exoesquelético.

En los sistemas de captura de movimientos mediante ultrasonidos se utilizan emisores que generan señales ultrasónicas que son capturados por receptores situados en posiciones específicas, para determinar la posición del emisor en el espacio, e incluso

su orientación en algunos casos. A menudo se utilizan combinados con los sistemas de mediciones inerciales [Vlasic et al., 2007].

Ventajas de los sistemas ultrasónicos:

- Amplio rango de medición.
- La detección no es afectada por materiales u objetos.
- Combinarse con otros sistemas de captura de movimientos.

Desventajas de los sistemas ultrasónicos:

- Velocidad de respuesta lenta.
- Precisión baja.
- Interferencia por temperatura y humedad.
- Ruido en caso de existir otras señales ultrasónicas.

Los sistemas de captura de movimiento inercial miden la rotación de los ángulos articulares usando pequeños sensores llamados giroscopios o acelerómetros colocados en los miembros de interés. Este tipo de sistemas de captura de movimiento no utiliza mecanismos externos, por lo general son de tamaño pequeño y tienen un amplio rango de captura [Sato et al., 2011].

Ventajas de los sistemas inerciales:

- Alta precisión.
- Bajo costo.
- Pequeño tamaño.
- Alta velocidad de respuesta.

Desventajas de los sistemas inerciales:

- Pueden ser afectados por la gravedad.
- Sufren de ruido en las lecturas.
- Sufren de drifting con el tiempo.
- Es necesario aplicar filtros para evitar errores en las lecturas.

La interfaz motriz que parece la más adecuada para este trabajo es la teoría de los guantes de datos, capaz de medir los movimientos de los dedos y de la mano, combinada con los sensores inerciales.

[Kim et al., 2009] utilizando la unidad de medición inercial (IMU) se reconocen los movimientos de la marcha humana y son utilizados como entradas de control de robot humanoide para la imitación de caminar en línea. También se sugiere un método de reconocimiento de la marcha basado en la detección de soporte de pie y la estimación de la postura.

[Koenemann and Bennewitz, 2012] [Koenemann et al., 2014] propone un sistema que permite a un robot humanoide imitar movimientos complejos de todo el cuerpo de humano en tiempo real convirtiendo los ángulos de las articulaciones humanas en ángulos articulares del robot para garantizar la estabilidad. Los movimientos angulares del usuario son previamente capturados por un sistema de captura de movimientos basado en sensores inerciales.

La figura 2.12 muestra un ejemplo de un sistema de captura de movimientos inerciales.



FIGURA 2.12. Sistema de captura de movimientos basado en sensores inerciales.

## 2.4. Guantes de datos

Los guantes de datos, permiten medir los movimientos relativos de los dedos en relación con la mano. Para determinar los movimientos absolutos por completo, también hay que tomar nota de la posición de la mano con un rastreador. Los tipos de rastreador utilizados para esta función son principalmente sensores electromagnéticos y, a veces los sensores ultrasónicos. La principal dificultad es medir las rotaciones de las falanges con dispositivos ligeros y compactos. Varios principios están en competencia: la medición de la deformación de fibras ópticas, de efecto Hall o variación de resistencia.

Los guantes de variación de resistencias: Estos guantes utilizan el principio de variación de una resistencia como una función de la elongación del soporte sobre el que se fija. Cuando se pliega el transportador, cambia el valor de resistencia. Gracias a un procedimiento de calibración, uno puede determinar los movimientos de los dedos [Zaldívar Colado, 2009].

Los guantes de fibra óptica: en cada dedo del guante se fijan dos (o tres) fibras

ópticas dispuestas en la alineación de los dedos. Son más o menos del largo de cada articulación para detectar el movimiento. Las fibras ópticas están conectadas a un dispositivo optoelectrónico. Un extremo de cada bucle óptico está conectado a un diodo LED y el otro extremo a un fototransistor. Si la fibra está derecha, no hay reducción de la luz transmitida. Por el contrario, si la fibra se dobla, la intensidad de la luz recibida se atenúa, porque ya no está en condiciones de reflejo perfecto (repetición). La exactitud de los ángulos de articulaciones es del orden de 5 a 10 grados [Zaldívar Colado, 2009].

Los guantes de efecto Hall: un guante Hall tiene cuatro sensores de posición por dedo, de modo que son veinte para todo el guante. Cada rotación de una falange es medida por un sensor de efecto Hall colocado en la unión mecánica del guante. La resolución (12 bits) de las mediciones se da en la décima de grado [Zaldívar Colado, 2009].

La figura 2.13 muestra un tipo de guante basado en el efecto hall.

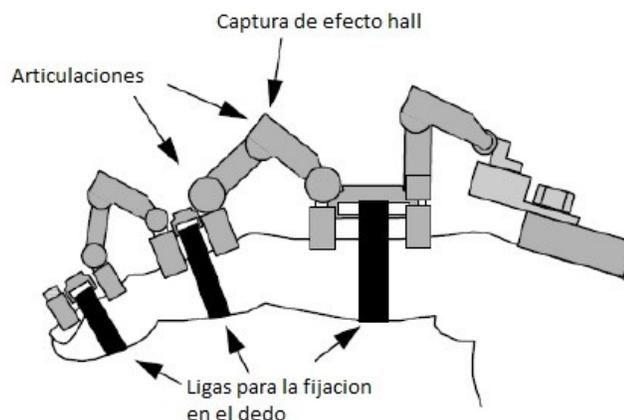


FIGURA 2.13. Guante de datos de efecto hall.

Los tipos de guantes Pinch Glove: los guantes antes descritos miden la deformación de los dedos. Este no es el caso del par de guante "Guante Pinch", que está basado en el fenómeno siguiente: el contacto entre dos o más dedos crea un bucle conductor (circuito cerrado), lo que permite una serie gestos de pellizco virtuales que se pue-

den utilizar en aplicaciones. Al igual que con otros guantes de datos, un rastreador está puesto en cada guante [Zaldívar Colado, 2009].

La figura 2.14 muestra un Pinch Glove.

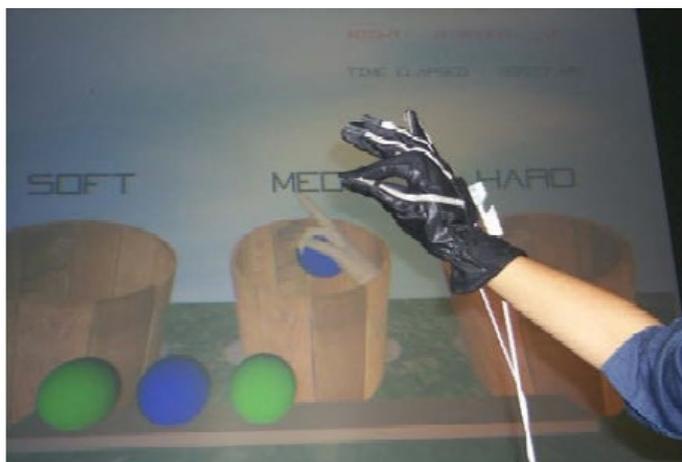


FIGURA 2.14. Guante de datos de tipo Pinch glove.

Los problemas técnicos importantes en estos dispositivos son la repetibilidad, la precisión y la fiabilidad de los dispositivos [Fuchs and Moreau, 2006].

Hoy en día, podemos encontrar en el mercado una variedad de dispositivos para la captura de los movimientos de la mano, ya que desde su aparición su evolución ha sido constante, desde los primeros prototipos creados por el MIT y su guante Sayre [Sturman and Zeltzer, 1994], hasta guantes muy sofisticados y con captura de movimientos de flexión y extensión, aducción y abducción [Borst and Indugula, 2005] [Borst and Indugula, 2006] [Zaldívar Colado, 2009].

En 1987, Tom Zimmerman diseñó el primer sensor de movimiento de tipo guante para tocar música sin tocar ningún instrumento musical, y fundó VPL Research para vender su sensor llamado DataGlove. DataGlove es un guante ligero que monta diez cables de fibra óptica a lo largo de cada dedo. Cada cable fibroótico equipa un LED en un extremo y un fototransistor en el otro extremo. Cuando el cable se flexiona, parte de la luz del LED sale de pequeños arañazos en la superficie del cable óptico, por

lo que el sensor obtiene la postura de una mano humana. Después de la invención de DataGlove, se presentan varios tipos de guantes de datos [Sturman and Zeltzer, 1994].

En 1991 el Immersion CyberGlove, inventado por Jim Kramer y comercializado por la empresa Virtex (ahora Immersion), el CyberGlove se vale de 18 ó 22 delgados medidores de esfuerzo montados sobre un fino tejido elástico de nylon. El tejido elástico no cubre la superficie palmar (tampoco la punta de los dedos en el modelo de 18 sensores) lo que permite una mejor ventilación. Los sensores son rectangulares para medir la flexión, y en forma de U para medir la abducción. Dos de ellos se utilizan para medir el cabeceo (pitch) y guiñada (yaw) de la muñeca [Kessler et al., 1995].

La figura 2.15 muestra el CyberGlove.



FIGURA 2.15. Guante de datos CyberGlove.

Comercialmente encontramos el FakeSpace Pinch Gloves, estos guantes sirven para detectar el contacto entre dos o más dedos (gesto de pinza o pinch gestures), gracias a la tela conductora que se encuentra en las yemas de los dedos del guante. Su mayor ventaja es que no precisan proceso alguno de re calibración [Bowman et al., 2001].

El Fifth Dimension Data Glove permite medir la flexión de cada dedo y la orientación de la mano. Para ello incorpora bucles de fibra óptica en cada articulación, y una unidad de control que incluye la circuitería que mide la orientación. El ex-

tremo de cada bucle está conectado a un LED y la luz que vuelve al otro extremo es captada por un fototransistor. Cuando la fibra está recta, no hay atenuación de la luz transmitida. Al flexionar, se pierde parte y el fototransistor capta menos luz [Kuroda et al., 2004].

Immersion CyberTouch, este guante consiste en 6 pequeños estimuladores vibrotáctiles que se le añaden al guante CyberGlove, uno en cada dedo y uno más en la palma. Mediante todo el conjunto es posible generar desde sensaciones simples hasta complejos patrones de realimentación táctil [Díaz-Tribaldos et al., 2015].

Immersion CyberGrasp, este producto consiste en un exoesqueleto externo que se une al guante CyberGlove para proporcionar un retorno de fuerzas resistivo a cada dedo. Para ello aplica fuerzas individuales mediante una red de tendones guiados por el exoesqueleto y se puede programar para evitar que los dedos del usuario atraviesen o aplasten un objeto virtual. Gracias a esta estructura externa las fuerzas se aplican de forma perpendicular a los dedos a través de todo su rango de movimiento [Zaldívar Colado, 2009].

La figura 2.16 muestra el CyberGrasp.



FIGURA 2.16. CyberGrasp montando en la mano del usuario.

Como guates de datos experimental encontramos los trabajos de [Zaldívar-Colado et al., 2010] [Zaldívar-Colado et al., 2011] donde presenta un siste-

ma de captura de movimientos de los dedos de la mano compuesto por un sistema mecánico que trasmite los movimientos de las falanges proximal y medial de cada dedo a través de un cable que es captado por una cámara utilizando el método de Lucas-Kanade para seguir el movimiento en tiempo real, al medir el desplazamiento del cable permite determinar la variación angular de la flexión generada por las falanges.

La figura 2.17 muestra el guante de Zaldívar.

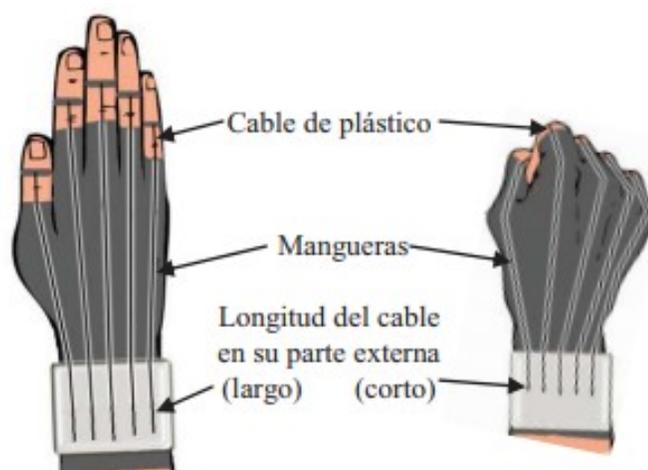


FIGURA 2.17. Posturas de la mano para el agarrado de un objeto con el guante de datos.

[Zannatha et al., 2011] presenta el desarrollo de un sistema de tracking sin contacto que permite detectar la posición y orientación de las manos del operador en el espacio así como el movimiento de cada uno de sus dedos. El movimiento de los dedos se detecta calculando el flujo óptico en imágenes capturadas por una cámara web instalada en el dorso de cada una de las manos del operador que registran la imagen de los hilos de nylon que transmiten el movimiento de flexión-extensión entre las falanges medial y proximal.

La figura 2.18 muestra el guante de Zannatha.



FIGURA 2.18. Guante de datos experimental de Zannatha.

## 2.5. Sistema de Captura de movimientos con sensores inerciales

Como se mencionó en las secciones anteriores, los sistemas de captura de movimientos basado en sistemas inerciales utilizan pequeños sensores que son capaces de medir la velocidad y la aceleración. Al igual que los sistemas mecánicos, son portátiles, pero no pueden medir posiciones y distancias directamente para aplicaciones que deben ejemplificar la geometría de objetos en el entorno.

Principalmente en el campo de la medicina, por su relación con el movimiento humano el uso de sensores inerciales en miniatura se ha convertido en una práctica común para su análisis. Los giroscopios y acelerómetros micromecanizados se utilizan en varias aplicaciones que incluyen el monitoreo de las actividades de la vida diaria, la evaluación de la carga de trabajo mecánica interna en los estudios de ergonomía, medición de trastornos neurológicos, y la realidad mixta y aumentada [Roetenberg et al., 2005] [Tamayo Serrano, 2016] [Zaldívar-Colado et al., 2017].

Los giroscopios Microelectromecánico (MEMS) son exactos para las mediciones de velocidad angular, pero sólo pueden utilizarse durante un corto tiempo para

calcular la orientación angular, ya que existe un error de desplazamiento relativamente pequeño debido a los efectos de la temperatura en la señal del giroscopio e introducirá grandes errores de integración [Roetenberg et al., 2005].

Se han propuesto algunas soluciones para el manejo de este error, aplicando ciertos filtros, como el filtro de Kalman utilizado por [Luinge and Veltink, 2005], y por [Roetenberg et al., 2005]. Otros investigadores como Tavares han utilizado librerías con resultados prometedores [Tavares et al., 2016].

A partir de los años dos mil, han surgido algunos trabajos en el uso de sistemas inerciales de captura de movimientos para el análisis de movimiento ambulatorio, en gran parte se debe al avance tecnológico de los sensores MEMS que son pieza clave en las unidades de movimiento inercial. Las mediciones de estos sensores se combinan en la fusión de sensor y algoritmos de perturbación antimagnética para producir seguimiento de movimiento preciso y virtualmente libre de barrido [Roetenberg et al., 2009].

[Popelka, 2014] informa sobre una solución particular para auto balancear una plataforma basada en el sensor MPU6050 y sus parámetros que pueden escanear 6DOF, señala que es apropiado utilizar los giroscopios y acelerómetros MEMS, ampliamente extendidos y relativamente precisos. Su consumo y tamaño son insignificantes y pueden usarse directamente desde los modelos o juguetes más pequeños. Aunque señala que es importante eliminar el ruido y la deriva que estos producen.

La figura 2.19 muestra el trabajo de Popelka.

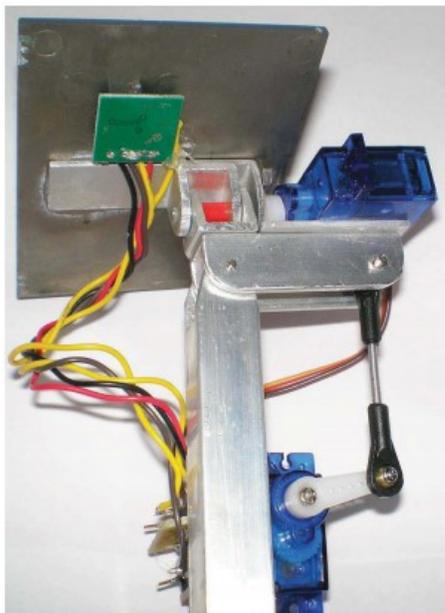


FIGURA 2.19. Plataforma auto balanceable de Popelka.

Hay trabajos realizados sobre guantes de datos que se han desarrollado basados en sensores inerciales. Algunos han trabajado con la recolección de estos datos en pacientes lesionados y su proceso de rehabilitación, con lo que ha sido posible examinar los movimientos de la mano del paciente, algunos de ellos se han desarrollado con 6 grados de libertad (DOF) [Lin et al., 2014] y otros con 9 DOF [Hsiao et al., 2015].

También se han hecho propuestas que representan los movimientos de la mano del usuario en un entorno virtual, aunque el propósito del trabajo ha sido ayudar al paciente con su rehabilitación [Lin et al., 2014] [Hsiao et al., 2015] [Tavares et al., 2016], le falta un modelo cinético que valide sus movimientos.

Esos avances han llevado al surgimiento de MOVEN [XsensTechnologies, 2015], el MOVEN mide la posición tridimensional (seis grados de libertad) y la orientación de los segmentos corporales en un sistema de coordenadas global mediante el uso de 16 unidades de sensores inerciales ocultas en un traje de lycra de cuerpo completo. Cada unidad de sensor contiene giroscopios 3D, acelerómetros 3D, Magnetómetros 3D y un sensor de compensación de temperatura. Los giroscopios miden la veloci-

dad angular que se integra a lo largo del tiempo para encontrar la orientación del segmento. Los acelerómetros miden la aceleración lineal que, después de retirar el componente de gravedad, se integra dos veces para encontrar la posición del segmento [XsensTechnologies, 2015] [Cloete and Scheffer, 2008].

La figura 2.20 muestra el sistema Moven.



FIGURA 2.20. Sistema Moven de Xsens.

Estos parámetros, junto con la información del magnetómetro y de la temperatura, se introducen en un filtro de Kalman que se actualiza continuamente para corregir las incertidumbres de la deriva y la posición de la junta. Los módulos sensores se colocan en la cabeza, los hombros, la pelvis, el brazo superior e inferior, la parte superior e inferior de la pierna, las manos y los pies [Cloete and Scheffer, 2008].

Sin embargo, a pesar de que los sistemas inerciales de captura de movimientos ofrecen algunas ventajas con respecto al costo y flexibilidad con los sistemas ópticos, mecánicos, etc., no han sido muy utilizados por la falta de confianza en su precisión de medición. Por lo anterior, se han realizado estudios de comparación de sistemas de captura de movimientos inerciales y ópticos, con resultados realmente alentadores [Cloete and Scheffer, 2008].

En la imagen 2.21 se observa a un usuario con los sensores inerciales en su traje.

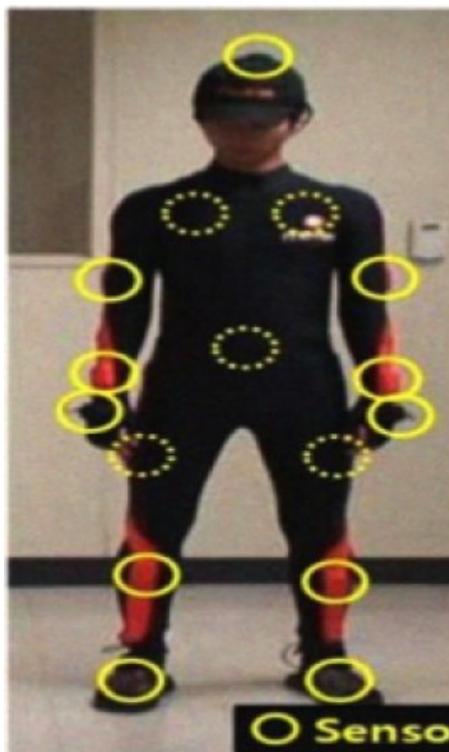


FIGURA 2.21. Usuario con sensores inerciales para la captura de movimientos.

Por otra parte, hay autores como [Tung and Van Thanh, 2015] que piensan que el campo de la investigación está dando un giro hacia los sistemas inerciales, debido a que en los últimos años han sido ampliamente utilizados en la fabricación de automóviles, teléfonos inteligentes y diversos equipos, ya que es bajo su consumo de energía, volumen y sobre todo su costo.

## 2.6. Análisis de resultados

Después de analizar trabajos recientes y significativos nos damos cuenta que existen opciones aparte de comprar o adquirir un dispositivo para un sistema de captura de movimientos y mapearlo en la realidad virtual. Existen trabajos que han desarrollado prototipos experimentales con resultados alentadores. Debido a que la

adquisición de estos dispositivos de entrada comerciales no está al alcance de muchos investigadores o estudiantes por su alto precio como algunos autores lo señalan en sus investigaciones, este trabajo pretende desarrollar un prototipo de guante de datos a bajo costo que pueda realizar las tareas de manipulación de objetos que realizan dispositivos líderes en el área de la realidad virtual.

Entre los sistemas de captura de movimientos, hemos encontrado que los sensores inerciales son una buena opción para este trabajo debido a su tamaño, costo y consumo, además existen trabajos analizados en el estado del arte, los cuales han capturado el movimiento del cuerpo humano o algún determinado elemento, ya que el objetivo de este trabajo de investigación es capturar el movimiento de los dedos de la mano, podemos encontrar las bases suficientes para la realización de un sistema de captura de movimientos con sistemas inerciales para la mano humana.

Sin embargo, los sensores entregan valores que pueden contener ruido y no ser tan precisos, una manera de mitigar esto es utilizar métodos o filtros que minimicen estos sesgos como ya se ha hecho en trabajos descritos en las secciones anteriores.

Por lo tanto, de acuerdo a las secciones anteriores, es posible modelar una mano virtual que sea capaz de representar los movimientos de la mano humana a través de sensores inerciales, una vez analizada la estructura de la mano y sus movimientos, es posible modelar el movimiento por un modelo de cinemática directa.

Debido a su pequeño tamaño, se pueden colocar en las falanges de los dedos para medir la flexión de cada uno y enviarlos a una computadora, como se han realizado en trabajos con fines de rehabilitación de pacientes, la pc contendrá una aplicación que por medio de un modelo cinemático directo sea capaz de representar los movimientos en la escena virtual. Los trabajos revisados muestran que los sistemas de captura de movimientos inerciales son confiables y se pueden enfocar en la mano humana para el desarrollo de un guante de datos que nos brinde una alta precisión con una fácil calibración y que no sufra de desgaste con el uso como lo presentan otros guantes en el sector comercial.

Realizar casos de estudio y experimentos de comparación con otro guante de datos existente en el mercado, sería un buen sinodal que permita evaluar los parámetros esperados en la hipótesis.

## CAPÍTULO 3

# MARCO METODOLÓGICO

### 3.1. Modelado de la mano virtual

Los sistemas de realidad virtual que incluyen manos virtuales en la mayoría de los casos utilizan guantes de captura de movimientos que permiten realizar la animación de las posturas de la mano del operador, donde el realismo depende de la cantidad de sensores existentes en los guantes.

La mano virtual está compuesta de 16 modelos geométricos, los cuales fueron obtenidos del sistema Virtual Environment for Design and Assembly Planning II (VEDAP-II) [Zaldívar Colado, 2009]. Dicho sistema fue desarrollado para el ensamble virtual de piezas mecánicas, y utiliza un guante de tipo Cyberglove<sup>TM</sup> con 21 sensores que animan la mano virtual en la escena.

Para el desarrollo de este trabajo, se utilizaron los modelos geométricos de la mano. La flexión se realiza para las 3 falanges de los dedos (proximal, medial y distal). En la figura 3.1 se presenta el modelo de la mano virtual indicando las falanges de los dedos.

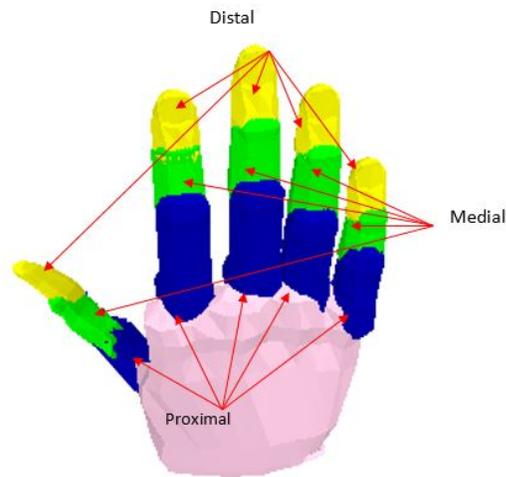


FIGURA 3.1. Modelo de la mano virtual y falanges de los dedos.

Cada elemento de la mano está almacenado en un archivo de texto que contiene la malla triangular del objeto. Esta malla tridimensional está representada por los vértices de cada triángulo que conforman el objeto. Con el fin de graficar el modelo de la mano en la escena, utilizamos la librería OpenGL [51] invocada desde el lenguaje de programación C++, utilizando la plataforma de desarrollo Visual C++ 2010 Express Edition.

### 3.1.1. Modelo cinemático de la mano

El modelo cinemático directo de la mano permite obtener la posición y orientación de cada uno de sus dedos a partir de los ángulos de giro de sus articulaciones. Para obtener este modelo, se estudia cada uno de los dedos de la mano de manera independiente aplicando el método clásico de Denavit-Hartenberg [Denavit and Hartenberg, 1964], el cual ha sido utilizado en otros trabajos donde se involucran guantes de datos y manos virtuales [Zaldívar Colado, 2009] [Zaldívar-Colado et al., 2006] [Zaldivar-Colado et al., 2010] [Lizárraga Reyes, 2010] [Li et al., 2016] [Tamayo Serrano, 2016] [Rodríguez Vega, 2016]. Este se podrá utili-

zar a la hora de identificar los sistemas de coordenadas asociados a los eslabones de cada dedo.

La propuesta es utilizar un sensor en cada falange de los dedos de la mano (proximal, medial y distal), uno en la palma de la mano y otro más en el antebrazo con lo cual determinaremos la posición de la palma de la mano en el espacio.

Todos los sensores estarán conectados a una placa base con la cual se realizará la lectura de cada falange y se enviará a la computadora por puerto serial, leyendo los datos desde la aplicación y aplicando cada valor del sistema de coordenadas a cada sensor en el ambiente virtual.

Los elementos de la mano virtual son representados como una cadena cinemática, y se puede observar que cada uno de los dedos tiene 4 grados de libertad, los cuales son representados por  $D_{i,j}$ .

El índice  $i$  representa el número de dedo (1-pulgar; 2-índice; 3-medio; 4-anular y 5-meñique), mientras que el índice  $j$  representa la falange que se rota en la articulación (1-proximal; 2-medial y 3-distal).

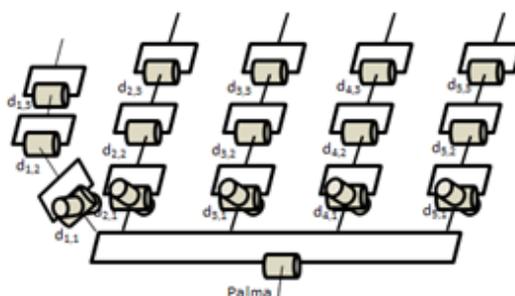


FIGURA 3.2. Esquema cinemático de la mano virtual.

Cada elemento de la mano, con excepción de la palma, tiene definido su sistema de coordenadas geométricas con respecto a otro elemento. Por ejemplo, la falange proximal de cualquier dedo tiene sus coordenadas definidas con respecto al marco de referencia de la palma; la falange medial de cualquier dedo las tiene con respecto al

marco de referencia de la falange proximal; y la falange distal al marco de referencia de la falange medial. La palma de la mano es el único elemento que está definido en el sistema de coordenadas del mundo virtual, y su posición y orientación son especificadas arbitrariamente en el espacio.

Por lo anterior, se tiene una matriz de transformación homogénea para cada elemento  $D_{i,j}$ , representada por  ${}^{d_{i,j-1}}M_{d_{i,j}}$ , lo cual se lee como: la matriz de transformación del elemento  $D_{i,j}$  con respecto al marco de referencia del elemento  $D_{i,j-1}$ .

Se considera la siguiente excepción: para la matriz de transformación de los elementos correspondientes a la falange proximal, éstos son definidos en el marco de referencia de la palma de la mano, por lo que se considera a la palma como  $D_{i,0}$ .

La matriz de transformación de la palma, está definida en el marco de referencia del mundo virtual  $W$  (sistema global de coordenadas):  ${}^W M_{d_{i,0}}$ .

Se considera una matriz de transformación homogénea con la siguiente representación:

$$M = \begin{bmatrix} R & \vec{P} \\ \vec{0} & 1 \end{bmatrix} = \begin{bmatrix} R_{1,1} & R_{1,2} & R_{1,3} & P_x \\ R_{2,1} & R_{2,2} & R_{2,3} & P_y \\ R_{3,1} & R_{2,2} & R_{2,3} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

donde  $R$  es la matriz de rotación y  $\vec{P}$  es el vector de posición.

### 3.1.2. Obtención de las posiciones relativas de los elementos de la mano

Como ya se mencionó anteriormente, el modelo virtual de la mano se obtuvo del sistema VEDAP-II [Zaldívar-Colado et al., 2006], sin embargo, los valores iniciales para el cálculo de la cinemática de la mano fueron desconocidos. Se obtuvo el registro de una posición de la mano en el espacio, con las respectivas coordenadas cartesianas de cada elemento de la mano con respecto al sistema de coordenadas del mundo virtual. Es decir, se conocía la posición y la orientación de cada falange y la de la

palma con respecto al mundo virtual ( ${}^W M_{d_{i,0}}$  y  ${}^W M_{d_{i,j}}$ ), pero no las coordenadas geométricas de cada falange con respecto al elemento anterior ( ${}^{d_{i,j-1}} M_{d_{i,j}}$ ). Para poder calcular la cinemática de la mano mediante la metodología de Denavit-Hartenberg es necesario conocer las coordenadas geométricas las falanges con respecto al elemento anterior correspondiente ( ${}^{d_{i,j-1}} M_{d_{i,j}}$ ).

Las tablas 3.1 y 3.2 presentan las coordenadas geométricas (posición y orientación) de cada elemento de la mano con respecto al sistema de coordenadas global, que fueron obtenidas del sistema VEDAP II.

TABLA 3.1. Posición de los elementos de la mano registrados en el sistema VEDAP II.

	Posición		
	Px	Py	Pz
Palma	3.85	-0.94	-5.97
Pulgar Proximal	3.71	-2.50	-10.02
Pulgar Medial	5.05	-4.66	-14.06
Pulgar Distal	6.15	-6.88	-15.53
Índice Proximal	7.73	-6.33	-14.38
Índice Medial	10.51	-8.47	-17.09
Índice Distal	12.12	-9.80	-18.66
Medio Proximal	8.79	-7.72	-12.45
Medio Medial	13.01	-7.84	-14.31
Medio Distal	15.16	-9.16	-15.96
Anular Proximal	9.23	-8.28	-10.22
Anular Medial	12.88	-9.26	-11.77
Anular Distal	14.88	-11.11	-13.16
Meñique Proximal	9.41	-8.49	-7.90
Meñique Medial	11.66	-10.50	-8.73
Meñique Distal	12.78	-12.30	-9.36

---

TABLA 3.2. Orientación de los elementos de la mano registrados en el sistema VEDAP II.

Elementos	Vector de Rotación			
	Angulo $\theta$	x	y	z
Palma	230.21	0.78	-0.01	0.62
Pulgar Proximal	116.95	-0.94	0.02	-0.34
Pulgar Medial	211.36	0.98	0.21	0.00
Pulgar Distal	200.24	0.98	0.21	0.00
Índice Proximal	119.22	-0.76	-0.06	-0.64
Índice Medial	239.07	0.76	0.05	0.64
Índice Distal	108.48	-0.72	-0.15	-0.67
Medio Proximal	97.74	-0.65	-0.31	-0.69
Medio Medial	117.24	-0.74	-0.12	-0.65
Medio Distal	113.80	-0.73	-0.15	-0.66
Anular Proximal	110.47	-0.68	-0.29	-0.67
Anular Medial	231.07	0.75	0.12	0.65
Anular Distal	119.95	-0.71	-0.22	0.67
Meñique Proximal	224.22	0.72	0.21	0.66
Meñique Medial	213.51	0.74	0.10	0.66
Meñique Distal	206.84	0.75	0.06	0.65

Para obtener las coordenadas geométricas de cada elemento de la mano relativas a su elemento anterior, fue necesario realizar una serie de transformaciones geométricas para obtener los datos deseados.

En primera instancia, se generó una matriz de transformación de cada elemento a partir de su posición y orientación.

Cabe mencionar que en los datos obtenidos del sistema VEDAP II, la orientación está representada como un vector de rotación, el cual tuvo que ser convertido a la forma de matriz. Lo anterior se realizó en dos pasos, primero se obtuvo el cuaternión  $Q$  que representa la orientación en la ecuación 3.2:

$$Q = [q_x \ q_y \ q_z \ q_s], \text{ donde } \begin{bmatrix} q_x = x \sin \frac{\theta}{2} \\ q_y = y \sin \frac{\theta}{2} \\ q_z = z \sin \frac{\theta}{2} \\ q_s = s \cos \frac{\theta}{2} \end{bmatrix} \quad (3.2)$$

Posteriormente, se obtuvo la matriz de transformación homogénea en la ecuación 3.3:

$$M = \begin{bmatrix} 1 - 2q_y^2 - 2q_z^2 & 2q_xq_y - 2q_sq_z & 2q_xq_z + 2q_sq_y & P_x \\ 2q_xq_y + 2q_sq_z & 1 - 2q_x^2 - 2q_z^2 & 2q_yq_z - 2q_sq_x & P_y \\ 2q_xq_z - 2q_sq_y & 2q_yq_z + 2q_sq_x & 1 - 2q_x^2 - 2q_z^2 & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

Una vez obtenidas las matrices de transformación de dos elementos (A y B) con respecto al mundo virtual, se aplicaron las siguientes transformaciones (por ejemplo, para conocer las coordenadas geométricas de A con respecto a B):

- Invertir la matriz de B. Esta operación da como resultado la matriz del sistema de coordenadas del mundo virtual con respecto a B.

$${}^W M_B^{-1} = {}^B M_W \quad (3.4)$$

$${}^W M_B^{-1} = {}^W M_{B2} * I \quad (3.5)$$

donde:

$$B2 = \begin{bmatrix} B_{1,1} & B_{1,2} & B_{1,3} & 0 \\ B_{2,1} & B_{2,2} & B_{2,3} & 0 \\ B_{3,1} & B_{2,2} & B_{2,3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

$$I = \begin{bmatrix} 1 & 0 & 0 & -B_{1,4} \\ 0 & 1 & 0 & -B_{2,4} \\ 0 & 0 & 1 & -B_{3,4} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.7)$$

- Multiplicar la inversa de la matriz de B por la matriz de A. Esta operación da como resultado la matriz de A con respecto a B.

$${}^B A = {}^B W * {}^W A \quad (3.8)$$

La matriz 3.8 contiene la información que necesitamos para poder aplicar la cinemática de la mano virtual.

La Tabla 3.3 presenta los valores obtenidos con el procedimiento antes mencionado:

TABLA 3.3. Posición relativa de los elementos de la mano virtual.

	Posición		
	Px	Py	Pz
Palma	0.00	0.00	0.00
Pulgar Proximal	-2.50	3.40	1.00
Pulgar Medial	0.00	4.78	0.00
Pulgar Distal	0.00	2.88	0.00
Índice Proximal	-2.60	10.40	0.00
Índice Medial	0.00	4.43	0.00
Índice Distal	0.00	2.61	0.00
Medio Proximal	0.00	10.60	0.00
Medio Medial	0.00	4.62	0.00
Medio Distal	0.00	3.01	0.00
Anular Proximal	2.20	9.80	0.00
Anular Medial	0.00	4.08	0.00
Anular Distal	0.00	3.06	0.00
Meñique Proximal	4.20	8.60	0.00
Meñique Medial	0.00	3.13	0.00
Meñique Distal	0.00	2.21	0.00

Los valores presentados en la tabla 3.3 nos indican lo siguiente:

- El elemento correspondiente a la falange proximal del dedo índice (**índice proximal**) se localiza a 2.6 centímetros a la izquierda del sistema de coordenadas de la palma de la mano y a 10.4 centímetros hacia arriba,
- El elemento correspondiente a la falange medial del dedo índice (**índice medial**) se localiza a 4.43 centímetros hacia arriba del sistema de coordenadas de la falange proximal,
- El elemento correspondiente a la falange distal del dedo índice (**índice distal**) se localiza a 2.61 centímetros hacia arriba del sistema de coordenadas de la falange medial de dicho dedo.
- Se sigue la misma representación para los demás elementos de la mano.

El valor de la orientación no se considera importante (en este cálculo), ya que si la mano está extendida (abierta), la rotación de cada elemento de la mano es de 0 grados, con respecto a su elemento anterior.

### 3.1.3. Cálculo de la cinemática de la mano virtual

Para graficar los elementos de la mano en el mundo virtual, es necesario obtener las coordenadas geométricas de cada elemento en el marco de referencia del mundo virtual  $W$ .

Para obtener los elementos de las falanges proximales en el marco de referencia de  $W$ , realizamos la siguiente operación:

$${}^W M_{d_{i,1}} = {}^W M_{d_{i,0}} * {}^{d_{i,0}} M_{d_{i,1}} \quad (3.9)$$

Los elementos de las falanges mediales y distales en el marco de referencia de  $W$  se obtienen por:

$${}^W M_{d_{i,j}} = {}^W M_{d_{i,j-1}} * {}^{d_{i,j-1}} M_{d_{i,j}} \quad (3.10)$$

Por ejemplo, para calcular las coordenadas geométricas de los elementos correspondientes al dedo índice, se realizan las operaciones de las ecuaciones 3.11, 3.12 y 3.13 :

$${}^W M_{d_{2,1}} = {}^W M_{d_{i,0}} * {}^{d_{i,0}} M_{d_{2,1}} \quad (3.11)$$

$${}^W M_{d_{2,2}} = {}^W M_{d_{2,1}} * {}^{d_{2,1}} M_{d_{2,1}} \quad (3.12)$$

$${}^W M_{d_{2,3}} = {}^W M_{d_{2,2}} * {}^{d_{2,2}} M_{d_{2,1}} \quad (3.13)$$

Donde:

- ${}^W M_{d_{2,1}}$  es la matriz de transformación de la falange proximal del dedo índice con respecto al marco de referencia del mundo virtual.
- ${}^W M_{d_{2,2}}$  es la matriz de transformación de la falange medial del dedo índice con respecto al marco de referencia del mundo virtual.
- ${}^W M_{d_{2,3}}$  es la matriz de transformación de la falange distal del dedo índice con respecto al marco de referencia del mundo virtual.
- ${}^W M_{d_{i,0}}$  es la matriz de transformación de la palma de la mano virtual con respecto al marco de referencia del mundo virtual.
- ${}^{d_{i,0}} M_{d_{2,1}}$  es la matriz de transformación de la falange proximal del dedo índice con respecto al marco de referencia de la palma de la mano virtual.
- ${}^{d_{2,1}} M_{d_{2,1}}$  es la matriz de transformación de la falange medial del dedo índice con respecto al marco de referencia de la falange proximal del mismo dedo.
- ${}^{d_{2,2}} M_{d_{2,1}}$  es la matriz de transformación de la falange distal del dedo índice con respecto al marco de referencia de la falange medial del dedo índice.

La misma formulación realizada para el dedo índice se realiza para los demás dedos de la mano virtual.

Cuando se tienen calculados todos los elementos de la mano en el marco de referencia global  $W$ , estos son graficados en la escena virtual utilizando las librerías de renderizado gráfico de OpenGL.

### 3.1.4. Rotación de elementos en ángulos no permitidos

Con el fin de evitar la rotación de los elementos de la mano en ángulos donde no puede girar en la vida real, se aplicó un esquema de restricciones para complementar el modelo Denavit-Hartenberg. Este esquema trata de restringir estos movimientos y aplica solo las rotaciones en los ejes correspondientes al elemento. Esta restricción surgió debido a la naturaleza de los sensores, que permite medir la rotación independiente de cada elemento.

En la vida real, un dedo no puede rotar en un eje  $Y$  mientras la palma esta inmóvil, o una falange medial o distal no puede rotar independientemente de la falange proximal. La dependencia de un elemento con respecto a su anterior se resolvió con el modelo  $DH$ , sin embargo los valores de referencia para la rotación son tomados de cada sensor, de esta manera la rotación era independiente para cada elemento.

Para evitar el problema del párrafo anterior se realizó el siguiente procedimiento:

Se obtienen los ángulos de  $Euler(\phi, \theta, \psi)$  de la matriz de transformación homogénea de la Palma  $M$  y se toma la rotación en  $\phi$ . Luego, se resta  $\phi$  con respecto a su elemento anterior.

$$M\phi = \begin{bmatrix} R & \vec{P} \\ \vec{0} & 1 \end{bmatrix}, R = M(\phi, 1, 0, 0) \quad (3.14)$$

Donde:

- $R$  es el vector de rotación
- $\phi$  es el ángulo a rotar permitido
- $\vec{P}$  es el vector de traslación

Luego se realizan los cálculos necesarios para restar ese ángulo a los demás elementos. De esta manera todos los elementos giran con respecto al giro o rotación que se obtiene de la palma.

$${}^{d_{i,j-1}}M\phi_{d_{i,j}} = {}^WM\phi_{d_{i,j}} * [{}^{d_{i,j}}M_{d_{i,j-1}}]^{-1} \quad (3.15)$$

$${}^WM_{d_{i,j}} = {}^WM_{d_{i,j-1}} * {}^{d_{i,j-1}}M\phi_{d_{i,j}} \quad (3.16)$$

Por último para agregar los movimientos de aducción y abducción a las falanges proximales se realiza la siguiente operación.

- Agregar rotación en  $\psi$  a las falanges proximales.

$${}^WM_{d_{i,j}} = {}^WM_{d_{i,j-1}} * {}^{d_{i,j-1}}M\psi_{d_{i,j}} \quad (3.17)$$

Donde  ${}^{d_{i,j-1}}M\psi_{d_{i,j}}$  debe contener el ángulo de interés ya restado. Para obtener la matriz con respecto al mundo del elemento  $d_{i,j}$ .

### 3.2. Obtención de los datos de los sensores inerciales

Con respecto a las unidades de mediciones inerciales, colocadas en cada falange de los dedos de la mano, la propuesta del trabajo es obtener estos datos por medio del MPU 6050 que opera a 3.3 voltios y utiliza un protocolo de comunicación I2C. Un IMU es un dispositivo capaz de medir la fuerza de aceleración y la velocidad angular, consta de un acelerómetro y giroscopio ambos de 3 ejes, por lo tanto tenemos 6 grados de libertad.

Para la lectura de los datos se utilizó una placa Arduino MEGA 2560 la cual es un microcontrolador basado en el ATmega2560. Cuenta con 54 pines digitales de entrada / salida (de los cuales 15 se pueden utilizar como salidas PWM), 16 entradas analógicas, 4 UART (puertos serie de hardware), un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, una cabecera ICSP y un botón de

reinicio. Contiene todo lo necesario para soportar el microcontrolador; simplemente se conecta a un ordenador con un cable USB o bien con un adaptador de CA a CC o una batería para empezar. La placa Mega 2560 es compatible con la mayoría de los shields (módulos específicos) diseñados para el Uno y las boards anteriores Duemilanove o Diecimila [ArduinoBoardMega2560, 2015].

En la figura 3.3 se presenta el modelo de la placa Arduino MEGA 2560.

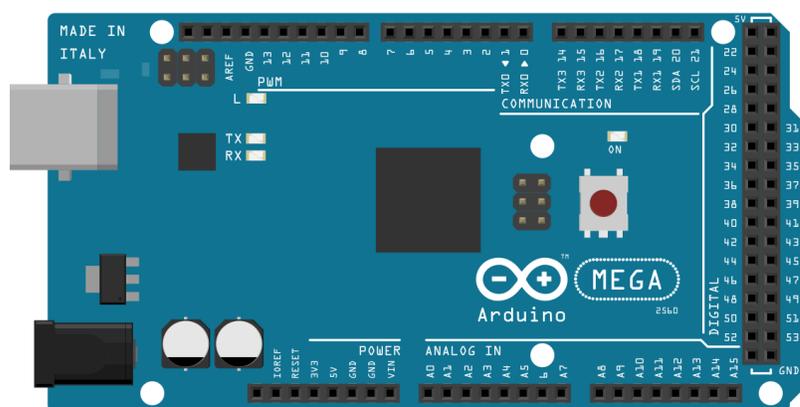


FIGURA 3.3. Modelo de la placa Arduino MEGA 2560.

Como dispositivo de medida se utilizó el sensor InvenSense MPU-6050 que contiene un acelerómetro MEMS y un giroscopio MEMS en un solo chip. Es muy preciso, ya que contiene hardware de conversión de 16 bits analógico a digital para cada canal. Por lo tanto, captura los canales  $x$ ,  $y$  y  $z$  al mismo tiempo. El sensor utiliza el I2C-bus para la interfaz con el Arduino.

El MPU-6050 no es caro, especialmente teniendo en cuenta el hecho de que combina un acelerómetro y un giroscopio [ArduinoMPU6050, 2015].

En la figura 3.4 se presenta el sensor MPU6050.

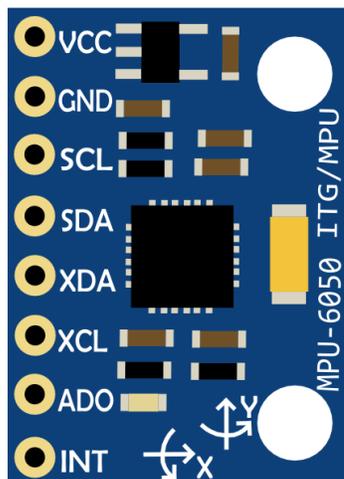


FIGURA 3.4. Sensor InvenSense MPU-6050.

Para establecer la comunicación entre el sensor MPU6050 y la placa Arduino Mega 2560 fue necesario instalar el software que provee Arduino para manipular y codificar sus placas, este software se encuentra disponible en la página web de Arduino [ArduinoSoftware, 2015], fue necesario añadir también las librerías I2C para la comunicación y la del sensor que se pueden encontrar en [Rowberg, 2015].

El sensor tiene un "Digital Motion Processor" (DMP), también llamado "Digital Motion Processing Unit". Este DMP puede ser programado con firmware y es capaz de hacer cálculos complejos con los valores del sensor. Para este DMP, InvenSense tiene una política de desaliento, al no proporcionar suficiente información para programar el DMP. Sin embargo, algunos han utilizado la ingeniería inversa para capturar el firmware como lo hacen las librerías [ArduinoMPU6050, 2015].

El DMP ("Digital Motion Processor") puede hacer cálculos rápidos directamente en el chip. Esto reduce la carga del microcontrolador (como el Arduino). El DMP es incluso capaz de hacer cálculos con los valores de sensor de otro chip, por ejemplo un magnetómetro conectado al segundo (sub) -I2C-bus.

Los valores del acelerómetro y del giroscopio se denominan valores "crudos". Esto es igual que con otros sensores de acelerómetro y giroscopio. Una aplicación más

sofisticada es usar el DMP para recuperar valores calculados específicos del sensor [ArduinoMPU6050, 2015]. Además se han utilizado en trabajos para medir la rotación de la muñeca en guantes de datos [Tavares et al., 2016].

### 3.2.1. Obteniendo los datos de un sensor

Con los demos, ejemplos y las librerías proporcionadas en la página web de Arduino, se pueden leer los valores que envía el giroscopio, la configuración permite conectar un sensor MPU6050 a un MEGA 2560, pero para eso se debe hacer la conexión física adecuada entre el sensor y la placa, de lo contrario no se podrán obtener los valores del sensor. La conexión física necesaria para la correcta lectura de los valores se muestra en la figura 3.5.

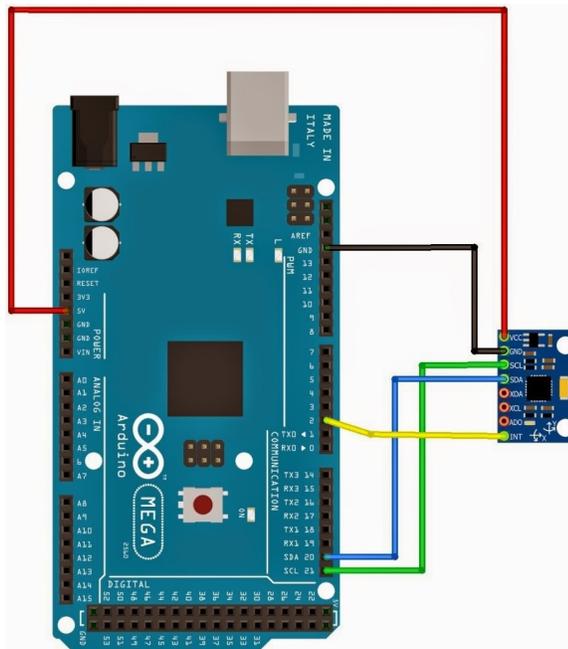


FIGURA 3.5. Conexión física entre la placa MEGA 2560 y el sensor MPU-6050.

Una vez que se tiene la conexión correcta entre la placa y el sensor y el programa cargado en la placa Arduino, es posible leer los valores en una pc por puerto serial.

### 3.2.2. Obteniendo los datos de dos sensores

La idea de utilizar un sensor para cada falange de los dedos de la mano y uno más para la palma de la mano sería un total de 16 sensores, y dado que se utilizaba una placa por cada sensor, se buscó reducir el número de placas Arduino.

Dado que la conexión se realiza mediante I2C como se ha mencionado anteriormente, es necesario conectar los pines SDA y SCL, alimentación y un pin de interrupción (INT0), como se mostró en la figura 3.5, La dirección de la MPU-6050 es b110100X que tiene 7 bits de longitud. El bit menos significativo (LSB) de la dirección de 7 bits está determinado por el nivel lógico en el pin AD0. Esto permite conectar dos MPU-6050s al mismo bus I2C. Cuando se utiliza en esta configuración, la dirección de uno de los dispositivos debe ser b1101000 (el pin AD0 es lógico bajo) y la dirección del otro debe ser b1101001 (el pin AD0 es lógico alto). Por lo tanto, si se desea modificar la dirección I2C del dispositivo se dispone del pin AD0, solo hay que conectar este pin a GND o 3.3V para la otra dirección I2C [InvenSense, 2015]. En la figura 3.6 se muestra el esquema de conexión.

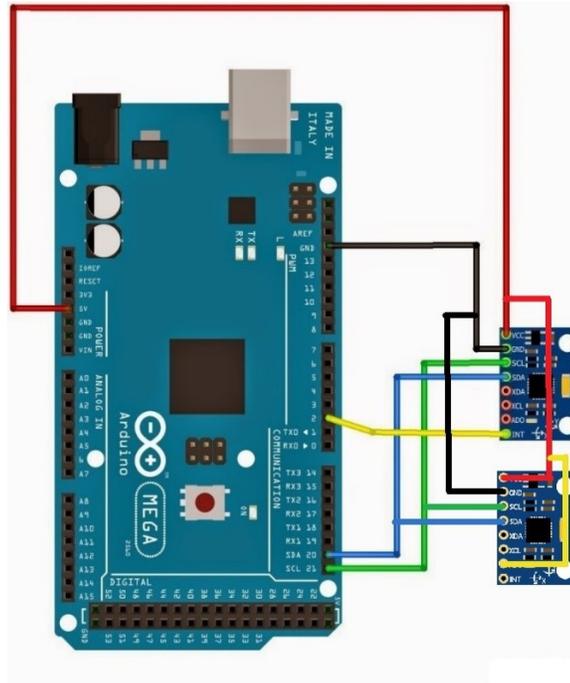


FIGURA 3.6. Conexión física entre la placa MEGA 2560 y dos sensores MPU-6050.

Cuando se necesitan más sensores MPU-6050 en un proyecto, el bus I2C puede extenderse con multiplexores [ArduinoMPU6050, 2015].

### 3.3. Interpretando los valores de los sensores

Para la realización de este proyecto fue necesaria la programación de las placas Arduino y de la aplicación de la escena virtual. Para la primera parte fue necesario utilizar el demo DMP con el que es posible realizar la lectura de un sensor, proporcionado por la librería I2CdevLib [Rowberg, 2015], cabe destacar que fue necesario la modificación de este archivo para hacer posible la lectura de dos sensores en una misma placa Arduino, además de la conexión física que fue explicada en la sección anterior.

### 3.3.1. Librería I2Cdevlib

La librería I2Cdevlib fue desarrollada por Jeff Rowberg y se puede encontrar en su sitio web o en el sitio oficial de Arduino. La librería utiliza el Digital Motion Processor (DMP) y obtiene los valores angulares en formas diferentes a través de Macros que se pueden configurar antes de subir el código a la placa Arduino. Las macros definidas son representadas en la figura 3.7.

```
#define OUTPUT_READABLE_QUATERNION  
  
#define OUTPUT_READABLE_EULER  
  
#define OUTPUT_READABLE_YAWPITCHROLL  
  
#define OUTPUT_READABLE_REALACCEL  
  
#define OUTPUT_READABLE_WORLDACCEL
```

FIGURA 3.7. Macros disponibles para la obtención de datos en la librería I2Cdevlib.

Para utilizar una de ellas se deben deshabilitar las demás opciones, ya que cada una entrega los valores en formas diferentes. Para la realización de este proyecto se optó por utilizar los cuaterniones que permiten representar la rotación de un elemento, al ser una estructura de cuatro valores, libre de gimbalLock [Tavares et al., 2016], que se puede manejar e interpretar de forma sencilla, esto para facilitar la comunicación entre las placas Arduino y la aplicación.

Como se explicó en las secciones anteriores, el sensor puede adoptar las direcciones 0x68 o 0x69 que depende de la conexión física entre el sensor y la placa Arduino, este valor se especifica en una variable del código que proporciona la librería, la cual contiene la conexión entre el código y el sensor físico y es a través de ella configuramos o leemos información del sensor.

### 3.3.2. Calibración de los sensores (respecto al mundo)

Al igual que cualquier otro sensor, el MPU6050 necesita ser calibrado antes de ser utilizado por primera vez debido a que probablemente no se encuentre 100 % en una posición horizontal, tenga errores de posición al soldar en una placa, o sencillamente esté desnivelado. Para solucionar ésto, un proceso de calibración tiene que realizarse en el cual existen dos parámetros que deben de ajustarse, la ganancia y el desplazamiento, para realizar este procedimiento el sensor debe de encontrarse en una posición plana y estable [Yalzin, 2014]. El proceso de calibración se realiza a través de un programa que debe cargarse a la placa Arduino de forma similar que el código que lee el sensor, los cuales son proporcionados por la misma librería I2CdevLib.

El proceso realiza un promedio de mil lecturas, estas lecturas durante los segundos del proceso definen el mínimo y máximo de valores esperados durante el ciclo y al final despliega los valores requeridos para remover el error. Estos valores deben especificarse en las siguientes 4 variables en la librería antes de ser cargada a la placa Arduino, como se muestra en la figura 3.8.

```
mpu.setXGyroOffset(-32);  
mpu.setYGyroOffset(-6);  
mpu.setZGyroOffset(34);  
mpu.setZAccelOffset(1893);
```

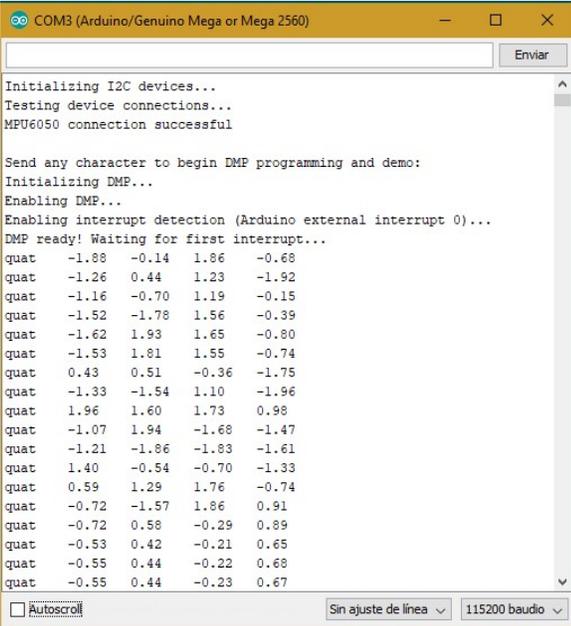
FIGURA 3.8. Variables de compensación (Offset) de calibración.

Estas compensaciones (OFFSET, nombrado de esta manera en inglés) son agregadas a los valores leídos por lo sensores con respecto a la posición en la que se encuentran. Cabe destacar que durante la fase de desarrollo, estos valores variaron entre un punto y otro refiriéndose a posiciones en el planeta tierra, por lo tanto los valores varían de una ciudad a otra, o con la orientación que el sensor tenga al Norte. Durante una estancia realizada en el tercer semestre de los estudios para la realización

de este trabajo, en el Laboratorio END-ICAP: Evolution of Neuromuscular Diseases: Innovative Concepts and Practices (URM 1179) de la Universidad de Versailles, Francia, los valores tomados después de la calibración variaban por mucho con respecto a los valores tomados en Culiacán, Sinaloa, México.

### 3.3.3. Obteniendo cuaterniones de los sensores

Como se explicó al principio de la sección 3.3.1, es posible leer los cuaterniones de la librería activando una macro, por lo tanto los cuaterniones se leen directamente desde el sensor y la librería entrega en una estructura de cuatro valores flotantes al puerto serial, en la figura 3.9 se muestra como se entregan los valores de salida:



```

COM3 (Arduino/Genuino Mega or Mega 2560)
Enviar
Initializing I2C devices...
Testing device connections...
MPU6050 connection successful

Send any character to begin DMP programming and demo:
Initializing DMP...
Enabling DMP...
Enabling interrupt detection (Arduino external interrupt 0)...
DMP ready! Waiting for first interrupt...
quat -1.88 -0.14 1.86 -0.68
quat -1.26 0.44 1.23 -1.92
quat -1.16 -0.70 1.19 -0.15
quat -1.52 -1.78 1.56 -0.39
quat -1.62 1.93 1.65 -0.80
quat -1.53 1.81 1.55 -0.74
quat 0.43 0.51 -0.36 -1.75
quat -1.33 -1.54 1.10 -1.96
quat 1.96 1.60 1.73 0.98
quat -1.07 1.94 -1.68 -1.47
quat -1.21 -1.86 -1.83 -1.61
quat 1.40 -0.54 -0.70 -1.33
quat 0.59 1.29 1.76 -0.74
quat -0.72 -1.57 1.86 0.91
quat -0.72 0.58 -0.29 0.89
quat -0.53 0.42 -0.21 0.65
quat -0.55 0.44 -0.22 0.68
quat -0.55 0.44 -0.23 0.67
Autoscroll Sin ajuste de línea 115200 baudio

```

FIGURA 3.9. Valores obtenidos de la librería I2Cdevlib. Los valores son obtenidos en cuaterniones.

Esto representa la rotación del elemento como se explicó en la sección 3.1.2 “Obtención de las posiciones relativas de los elementos”, por lo tanto estos valores representan la rotación real del sensor.

Debido a que en este trabajo se conectan dos sensores a una misma placa Arduino, la manera en la que se lee un sensor debió cambiar, desde asignar las dos direcciones en el código, como asignar los valores de offset después de la calibración. Los datos a entregar en el puerto serial debían de poder identificarse, conocer el origen del sensor que envía, por lo tanto se desarrolló un formato en el que son enviados los valores leídos. En la figura 3.10 se muestra como se envían los valores.

```

$-0.51X-0.72Y-0.27Z0.36A0.93B-0.36C0.06D-0.00|
$-0.52X-0.72Y-0.28Z0.37A0.93B-0.36C0.06D-0.00|
$-0.52X-0.72Y-0.28Z0.37A0.93B-0.36C0.06D-0.00|
$-0.52X-0.72Y-0.28Z0.37A0.93B-0.36C0.06D-0.00|
$-0.52X-0.72Y-0.28Z0.37A0.93B-0.36C0.06D-0.00|
$-0.52X-0.72Y-0.29Z0.37A0.93B-0.36C0.06D-0.00|
$-0.51X-0.72Y-0.29Z0.38A0.93B-0.36C0.06D-0.00|
$-0.51X-0.71Y-0.29Z0.38A0.93B-0.36C0.06D-0.00|
$-0.51X-0.71Y-0.29Z0.38A0.93B-0.36C0.06D-0.00|
$-0.51X-0.71Y-0.29Z0.38A0.93B-0.36C0.06D-0.00|
$-0.51X-0.71Y-0.30Z0.38A0.93B-0.36C0.06D-0.00|
$-0.51X-0.71Y-0.30Z0.38A0.93B-0.36C0.06D-0.00|
$-0.51X-0.71Y-0.30Z0.39A0.93B-0.36C0.06D-0.00|
$-0.51X-0.71Y-0.30Z0.39A0.93B-0.36C0.06D-0.00|
$-0.51X-0.70Y-0.30Z0.39A0.93B-0.36C0.06D-0.00|
$-0.51X-0.70Y-0.31Z0.39A0.93B-0.36C0.06D-0.00|
$-0.51X-0.70Y-0.31Z0.39A0.93B-0.36C0.07D-0.00|
$-0.51X-0.70Y-0.31Z0.40A0.93B-0.36C0.07D-0.00|
$-0.51X-0.70Y-0.31Z0.40A0.93B-0.36C0.07D-0.01|
$-0.51X-0.70Y-0.31Z0.40A0.93B-0.36C0.07D-0.01|
$-0.51X-0.69Y-0.32Z0.40A0.93B-0.36C0.07D-0.01|
$-0.51X-0.69Y-0.32Z0.40A0.93B-0.35C0.07D-0.01|
$-0.50X-0.69Y-0.32Z0.41A0.93B-0.35C0.07D-0.01|
$-0.50X-0.69Y-0.32Z0.41A0.93B-0.35C0.07D-0.01|
$-0.50X-0.69Y-0.32Z0.41A0.93B-0.35C0.07D-0.01|
$-0.50X-0.69Y-0.32Z0.41A0.93B-0.35C0.07D-0.01|
$-0.50X-0.69Y-0.33Z0.41A0.93B-0.35C0.07D-0.01|
$-0.50X-0.69Y-0.33Z0.41A0.93B-0.35C0.07D-0.01|

```

FIGURA 3.10. Los valores obtenidos en cuaterniones para dos sensores (formato).

En la figura 3.10 observamos que:

- **\$** indica que un paquete de datos ha llegado al puerto serial y que a partir de él los valores corresponden al primer valor del primer cuaternión, hasta encontrar el caracter A.
- Todo el valor que sigue corresponde al primer valor del cuaternión.
- **X** representa que los siguientes valores numéricos le pertenecen.

- **Y** representa que los siguientes valores numéricos le pertenecen.
- **Z** representa que los siguientes valores numéricos le pertenecen.
- **A** representa que han terminado los valores del primer cuaternión y marca el inicio de los valores del segundo cuaternión
- **B** representa que los siguientes valores numéricos le pertenecen.
- Lo mismo sucede con **C** y **D**.
- **|** significa que el paquete está completo y ha finalizado el segundo cuaternión.

En la figura 3.11 se muestra la trama y como está compuesta.

```

$-0.51X-0.72Y-0.27Z0.36A0.93B-0.36C0.06D-0.00|

```

FIGURA 3.11. Un paquete enviado por puerto serial.

Posteriormente estos valores pueden ser utilizados en la aplicación que dibuja la escena virtual mapeando los sensores a cada elemento de la mano.

### 3.3.4. Aumentando la precisión

La precisión estándar entregada por los sensores en cuanto a valores se trata, es aceptable para las mediciones de los movimientos de un determinado elemento. El sensor después de combinar los valores entregados por el giroscopio y el acelerómetro y siendo procesados en su DMP entrega mediante la librería, como ya lo vimos en secciones anteriores una estructura de 4 valores flotantes con una precisión de dos decimales. De acuerdo a la naturaleza del código de Arduino no fue posible dar formato a los valores de la estructura de la librería. De tal manera que se utilizó un poco de código en ANSI C para trabajar con los valores sin tener que reducir los decimales.

Para aumentar la precisión de los datos entregados por los sensores se buscó convertir los valores a un tipo de dato extremadamente grande, para obtener los máximos valores que permitan leer los sensores y de esta manera sean enviados a la aplicación.

Al tomar un valor flotante entregado por el sensor, éste se multiplica y se convierte a un valor muy grande, lo que nos entrega el valor con toda la precisión que se puede obtener. En la imagen 3.12 se observa la trama de valores de los cuaterniones con la precisión aumentada:

```

COM3 (Arduino/Genuino Mega or Mega 2560)
$-129644X0115576Y-107879Z-120635A0093487B-034942C0006237D0000006|
$0160754X-142407Y0132507Z0149530A0093481B-034948C0006274D-0000006|
$-146075X0174157Y-187591Z-161364A0093493B-034906C0006329D-000054|
$-178540X-136437Y0096600Z0176477A0093505B-034857C0006402D-000128|
$0165515X-180206Y0129498Z-169927A0093518B-034808C0006481D-000213|
$0071722X-056402Y0116491Z-067431A0093530B-034765C0006549D-000292|
$0193804X0162567Y-103228Z0195007A0093542B-034729C0006610D-000354|
$-036981X-097351Y-188891Z-030987A0093542B-034704C0006658D-000384|
$-040667X-110394Y0189288Z-036016A0093548B-034686C0006689D-000408|
$-044934X-123992Y-185095Z-045599A0093554B-034667C0006701D-000427|
$-060491X-174212Y0139758Z-061297A0093560B-034643C0006707D-000445|
$0064080X0182928Y-146868Z0064868A0093573B-034625C0006713D-000463|
$-072351X0063531Y-105053Z-072900A0093579B-034600C0006707D-000476|
$0186303X-163488Y-132354Z0186004A0093591B-034576C0006695D-000482|
$-080157X0140917Y-188061Z-085290A0093597B-034552C0006683D-000476|
$0100549X0155194Y0065093Z0077856A0093609B-034527C0006665D-000482|
$-160272X-027032Y0153393Z0190466A0093621B-034503C0006646D-000482|
$-104888X0051940Y0118463Z-133624A0093627B-034478C0006634D-000482|
$-157537X-121038Y0126440Z-090307A0093634B-034460C0006634D-000482|
$0174029X0064300Y-077325Z-141571A0093640B-034442C0006640D-000488|
$-175018X-163964Y-199945Z0090673A0093652B-034423C0006652D-000506|
$-101531X-122174Y0052221Z-174932A0093658B-034393C0006689D-000537|
$-031317X-038073Y0016503Z-054144A0093664B-034362C0006726D-000585|
$-041210X-049987Y0021966Z-071478A0093670B-034332C0006774D-000640|
$-041546X-050262Y0022491Z-072406A0093676B-034307C0006823D-000689|
$-041430X-049987Y0022784Z-072570A0093682B-034277C0006872D-000738|
$-041308X-049719Y0023077Z-072729A0093688B-034259C0006909D-000781|
Autoscroll Sin ajuste de línea 115200 baudio

```

FIGURA 3.12. Los valores obtenidos para dos sensores con precisión aumentada (formato).

Por lo tanto, si tomamos un valor “0006909”, y suponemos que ya está multiplicado por 100,000, al dividirlo por este mismo, el valor quedará en “0.06” y en caso de

que se envíe de esta forma a la aplicación en la escena virtual, el resto de la precisión se perderá, como es el caso de la precisión que entrega la librería estándar. Por otro lado, si los datos se envían en su forma convertida y se divide dentro de la aplicación en la escena virtual, la precisión continuará y al momento de realizar los cálculos de la cinemática, será con la precisión completa.

Este proceso es utilizado para los cuatro valores que conforman el cuaternión y fue aplicado para todos los sensores. Para validar este proceso se realizó un procedimiento experimental que se puede encontrar en el siguiente capítulo en la sección 4.2 “Precisión”.

### **3.4. Multi-hilos en la lectura de los sensores en la escena virtual**

Una vez que la lectura de los valores de los sensores se lleva a cabo en la placa Arduino y se envía por puerto serie a la pc, es necesario interpretarlo en la aplicación del AV. Por lo tanto fue necesario desarrollar una interfaz para capturar los datos del Arduino, que analizara gramaticalmente la trama de datos recibida con los valores de los dos sensores conectados a la placa.

Para realizar este proceso se utilizó programación basada en hilos, debido a que se utilizaron 6 placas Arduino, que leen dos sensores cada uno, excepto el de la palma, teniendo un total de 11 sensores para este trabajo de investigación, como se puede observar en la figura 3.13.

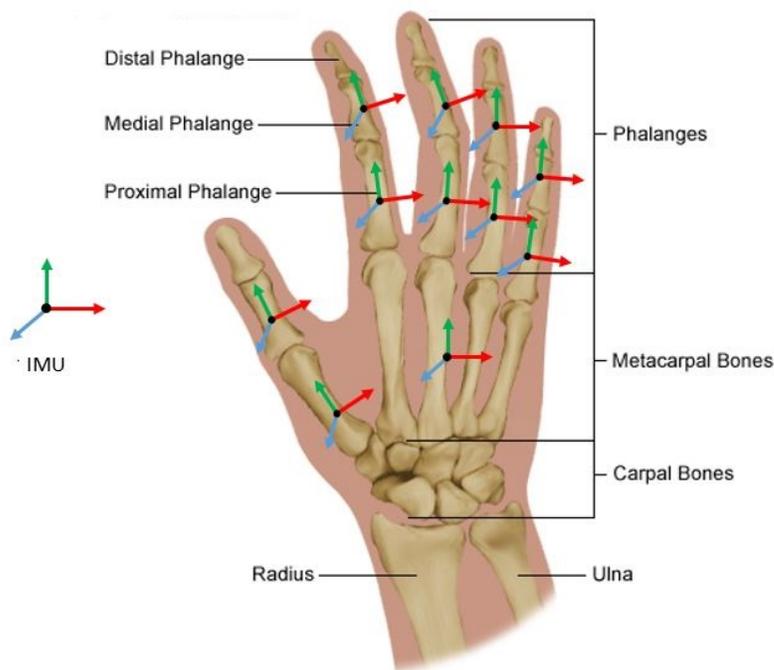


FIGURA 3.13. Localización de los sensores en la mano humana.

Se utilizó un hilo para cada placa Arduino, que se encarga de abrir el puerto y escuchar todas las tramas que la placa envía a la aplicación. Cabe mencionar que la lectura a cada placa se realiza de manera asíncrona para evitar tener que actualizar el los movimientos de la mano hasta que se lean todos los sensores.

Una vez recibida la trama, el modelo busca el inicio de paquete que corresponde a un \$, y comienza a leer y a organizar los datos de los cuaterniones. El formato en el que viene la trama es bien conocido por lo que el modelo acomoda los valores en variables temporales hasta encontrar el símbolo del final del paquete |.

Una vez que el paquete se ha entregado por completo, los valores almacenados son sincronizados con los cuaterniones en la aplicación, para esto el primer paso es castear cada valor de la trama ya que, en la sección anterior se mostró como enviar los valores con toda la precisión disponible. Los valores son divididos entre 100,000 que fue el dato por el que se multiplico en el código de Arduino, y son asignados a

un tipo de dato flotante, el cual, no solo contiene dos decimales sino toda la precisión que se puede obtener.

El diagrama del proceso se puede observar en la figura 3.14.

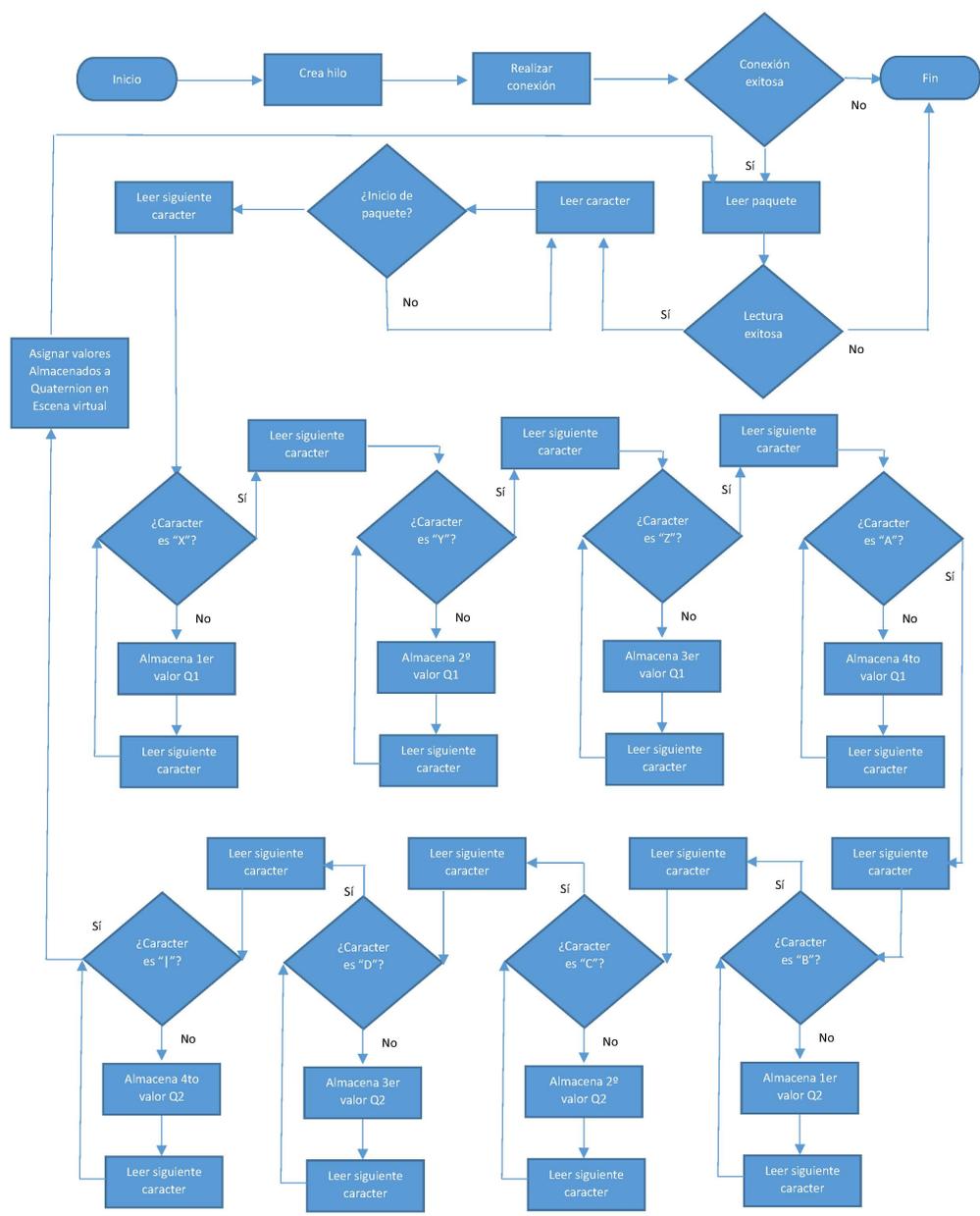


FIGURA 3.14. Diagrama de hilos en la aplicación de la escena virtual.

Mientras la aplicación está en ejecución, los hilos siguen sincronizando los cuaterniones hasta que esta finaliza.

Para auxiliar la programación en OpenGL se utilizó la librería VHT (Virtual-HandToolkit) que facilita las conversiones de cuaternión a matriz de rotación y viceversa. Como se vió en la sección 3.1.2 Obtención de las posiciones relativas de los elementos de la mano, es posible obtener una matriz de rotación desde un cuaternión.

El motivo de manejar las matrices de transformación es por la simplificación y claridad de las operaciones. Una vez que son obtenidas todas las matrices de rotación de los sensores se realiza el cálculo de la cinemática de la mano virtual para representar los movimientos de la mano virtual como se detalla en la sección 3.1.3.

En la figura 3.15 se muestra la arquitectura de la aplicación.

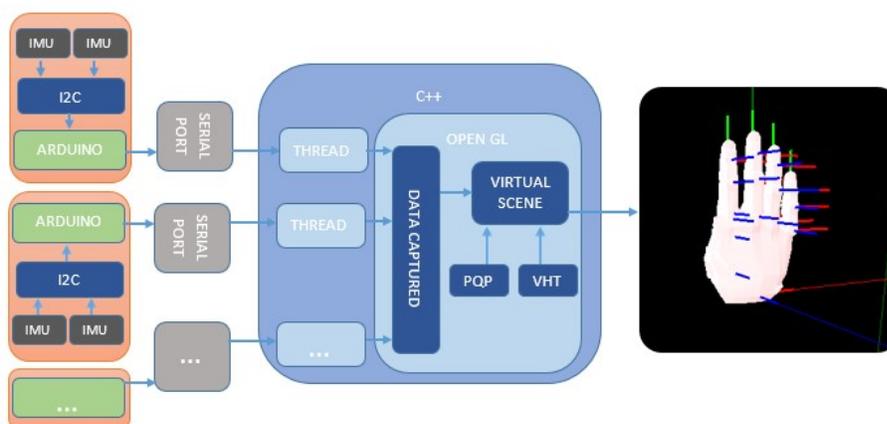


FIGURA 3.15. Arquitectura de sistema.

### 3.5. Calibración de los sensores en la escena virtual

Este proceso se realiza una vez iniciada la aplicación de la escena virtual, cuando la sincronización de los sensores está dada con los elementos de la mano. La sincronización ocurre debido a la naturaleza de los sensores y sucede de 10 a 20 segundos después de que el sensor empieza a emitir valores, mientras se estabiliza la lectura de datos del sensor.

En este punto aún no es posible trabajar con el guante y la mano virtual, así que por el mismo motivo que se realiza la primera calibración, se aplican nuevos valores de

compensación (OFFSET) que son controlados por la aplicación de la escena virtual, esto se realiza dado que pueden existir pequeñas variaciones de movimiento entre los usuarios. Una vez colocado el guante en la mano del usuario y la aplicación ha sido sincronizada correctamente, se pide al usuario colocar su mano en una determinada posición (la palma abierta 90 grados con respecto a la tierra) y él mismo puede aplicar las compensaciones (OFFSET) que sincronizarán la mano real con la mano virtual.

Una vez realizado este proceso la mano del usuario y la virtual están listas para trabajar en la aplicación. El proceso se detalla a continuación.

- Se toman los valores iniciales de cada sensor en dicha posición en una matriz  $M$  del elemento  $i,j$  con respecto al mundo real.

$${}^W M_{SINI_{i,j}} \quad (3.18)$$

- Se invierte la matriz de transformación.

$$[{}^W M_{SINI_{i,j}}]^{-1} \quad (3.19)$$

- Para obtener el movimiento continuo de los sensores se multiplica la matriz de transformación de los elementos por la matriz de transformación.

$${}^W M_{SINI_{i,j}} t + 1 = {}^W M_{SINI_{i,j}} t * [{}^W M_{SACT_{i,j}}]^{-1} \quad (3.20)$$

El resultado es almacenando en la matriz actual correspondiente al sensor en un tiempo  $t+1$ , y de esta manera se obtienen los valores que se compensaran (offset)

del momento del comienzo de la lectura hasta el momento en el que el usuario calibra el guante. Este es todo el proceso que se realiza para llevar a cabo la calibración y sincronización de la mano del usuario y la virtual.

### 3.6. Prototipo del guante de datos

Una de las partes necesarias para este trabajo fue la construcción del guante prototipo y aunque ya se tenían indicadas las falanges a medir, aún faltaba la colocación de los sensores en la mano del usuario para completar este proceso. Se consideró un diseño funcional del guante prototipo para la realización de experimentos y validación de este trabajo, aunque no es la finalidad ni objetivo de este proyecto validar el diseño mecánico del guante.

A lo largo de la fase de desarrollo y pruebas unitarias se diseñaron algunas versiones del guante para mejorar la ubicación de los sensores en la mano del usuario. En una primera fase se colocaron dos sensores en el usuario, uno en la palma y otro en la falange proximal del dedo índice, como prueba unitaria y validar los movimientos en la escena virtual. Lo anterior se muestra en la figura 3.16.



FIGURA 3.16. a) Colocación de los sensores. b) Validando movimientos de palma y falange proximal.

En una primera versión se diseñó un prototipo de anillos que contenían al sensor y se podía fijar a las falanges del usuario, sin embargo a pesar de que el tamaño del prototipo era pequeño justo al tamaño del sensor, interfería con los movimientos

naturales de la mano del usuario y tenía mucha inestabilidad. El prototipo se observa en la figura 3.17.



FIGURA 3.17. a) Colocación del sensor en el prototipo. b) Puesto en la mano del usuario. c) Durante la flexión de la falange proximal.

Buscando reducir el tamaño en el prototipo anterior se simplificó el diseño de los anillos, que permitiera realizar los movimientos del usuario con naturalidad y que su colocación en las falanges fuera sencilla. Esta nueva versión permitió un mejor movimiento de los dedos garantizando los movimientos de flexión y extensión, sin embargo los anillos de las falanges se giraban hacia los lados, por lo cual no se garantizaban los movimientos de aducción y abducción. Los anillos se muestran en la figura 3.18.



FIGURA 3.18. a) Nuevo diseño. b) c) d) Colocación de los anillos con los sensores en las falanges.

En la última versión de los anillos se optó por unir las piezas de la falange proximal y la medial para evitar que se rotaran sobre el dedo, así garantizar los movimientos de aducción y abducción. Este diseño seguía permitiendo realizar los movimientos naturales de la mano del usuario. El diseño se muestra en la imagen 3.19.

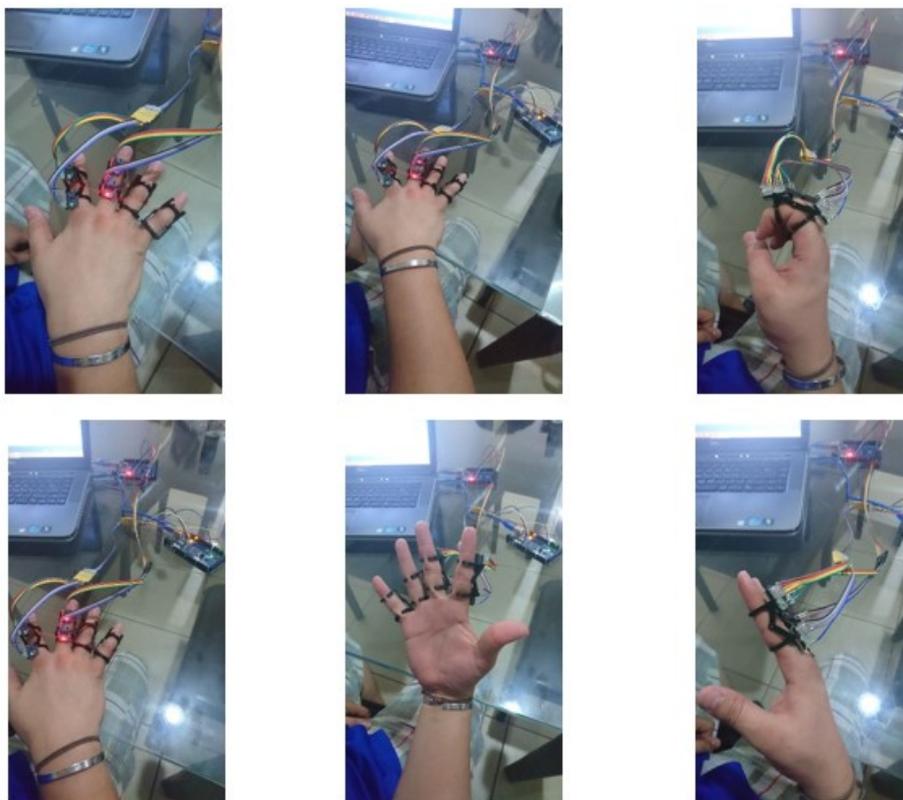


FIGURA 3.19. a) Nuevo diseño. b) c) d) Colocación de los anillos con los sensores en las falanges.

Después de tener el prototipo de los anillos funcionando, fue necesario colocar en un guante de tela el sensor que corresponde a la palma, y se realizó el cableado correspondiente. El usuario se coloca el guante de tela y después los anillos a cada dedo en las falanges correspondientes, en la palma, el guante contiene velcro con el cual puede fijarse el guante y los cables a la mano. En la figura 3.20 se muestra un usuario con el prototipo del guante de datos final.

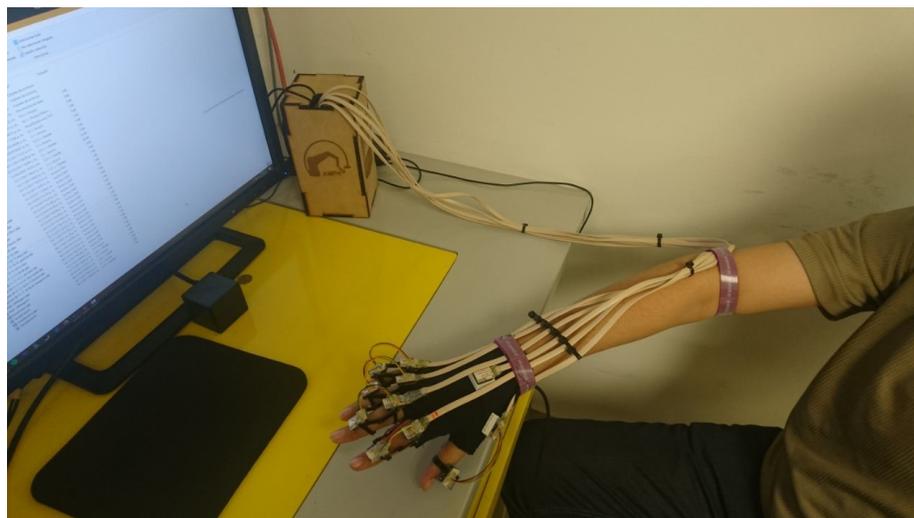


FIGURA 3.20. Prototipo final del guante de datos.

## CAPÍTULO 4

# EXPERIMENTACIÓN Y RESULTADOS

Con el fin de llevar a cabo una validación de los dispositivos propuestos, se llevaron a cabo cinco experimentos divididos en tres partes, la primera parte está relacionada con la lectura de los sensores, donde se realizaron tres procedimientos: en el primero, el sensor midió el desfase (drift por su tecnicismo en inglés) o deriva que se produce durante una hora de lectura, en el segundo se midió la frecuencia (hertz) de lectura de datos de los sensores, en el tercer experimento la resolución que estos sensores nos ofrecen. La segunda parte de la experimentación se refiere a la precisión donde se muestran los resultados obtenidos. En la tercera y última parte de este capítulo se realizó una manipulación de objetos virtual, donde se contabilizaron tiempos de interés indicados en nuestra hipótesis, al final de éste se realiza un análisis (estadístico) de varianza para validar este trabajo.

### **4.1. Lectura de datos de los sensores**

En esta sección del proceso experimental fueron llevados a cabo los experimentos relacionados con la lectura de los sensores, de acuerdo a la naturaleza de estos es importante conocer la deriva (drift) que puedan acumular con respecto al tiempo, la frecuencia con la que entregan valores a nuestra aplicación y como en todo sistema de captura de movimientos, conocer la precisión éste ofrece.

En la siguientes tres secciones se detalla cada uno de los procedimientos.

### 4.1.1. Deriva (Drifting)

Con el fin de determinar cual es la deriva de los valores de los sensores con respecto al tiempo en la lectura de los datos, se realizó un experimento que nos permite determinar y conocer cuáles son sus valores de sesgo. Este experimento consistió en tomar el valor angular de un sensor colocado en una posición fija (con el fin de garantizar que se toma el mismo valor en la medición) durante el periodo de una hora. Cabe destacar que el sensor quedó inmovilizado durante todo el experimento en una placa de prueba, los valores angulares fueron almacenados con respecto al tiempo.

Estos valores fueron analizados, y se determinó que el sensor tenía un desfase de 2,5 grados en una hora. En la figura 4.1 se presenta la medida desfase (drift).

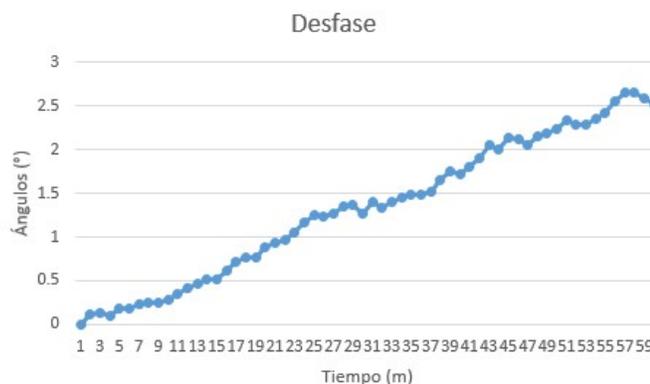


FIGURA 4.1. Deriva del sensor durante 1 hora.

En la figura 4.1 observamos que durante aproximadamente 20 minutos se tiene un barrido de menos de un grado (valor angular), por lo tanto se puede trabajar durante este tiempo sin problema. Durante el procedimiento se observó que, este sesgo se puede mitigar si el usuario calibra de nuevo los sensores, para que el desfase sea incluido en los valores de compensación (Offset).

### 4.1.2. Lectura de datos

Con el fin de medir la frecuencia real que tienen los sensores con respecto a la aplicación, se realizó un procedimiento capaz de validar la lectura y anchura de datos del dispositivo. Esto consiste en un analizador que es responsable de leer una cadena de datos (256 bytes), y separarla en cuaterniones para su uso en la escena virtual. Para realizar esta validación, se realizó un recuento del número de veces que se llena un cuaternión durante el análisis (en un intervalo de tiempo de un segundo). Se determinó que el cuaternión se puede rellenar 101 veces en un segundo, lo que indica que la velocidad de lectura de datos es de aproximadamente 100 Hertz.

La figura 4.2 presenta la cantidad de cuaterniones capturados por segundo durante un minuto.



FIGURA 4.2. Cuaterniones por segundo.

Posteriormente, se midió la velocidad del relleno de dos cuaterniones leyendo dos sensores en la misma placa Arduino<sup>TM</sup>. El procedimiento experimental nos permitió determinar que la tasa de llenado de dos cuaterniones es de 101 veces por segundo para dos cuaterniones.

Esto nos permite asumir que la comunicación de datos desde el sensor al ordenador es mayor que la velocidad de procesamiento de datos en el ordenador con la escena virtual. Por lo anterior, suponemos que si trabajamos con computadoras

dedicadas a la lectura de datos, o una computadora más potente, podemos tener una mejor velocidad de lectura.

#### 4.1.3. Resolución de la lectura de datos

En todos los sistemas de captura de movimiento es importante conocer la resolución de los sensores con respecto a los movimientos que pueden percibir. Por lo tanto, se realizó una prueba para validar esta resolución. Los valores angulares se obtuvieron durante el período de 1 minuto, haciendo movimientos de los sensores de 0 a 90 grados (4.3), ciclando estos movimientos, primero lentamente, y luego rápidamente, para tener una amplia gama de valores.

Después de leer los valores, se calculó la diferencia mínima entre dos valores consecutivos, y la diferencia máxima entre dos valores consecutivos, entre la lista completa de datos capturados.

Los resultados fueron los siguientes: una resolución mínima de 0.00004 grados y una resolución máxima de 6.76 grados para la precisión estándar (4.4) y para una precisión mejorada se encontró una resolución mínima de 0.00001 grados y una resolución máxima de 1.32 grados (4.5).

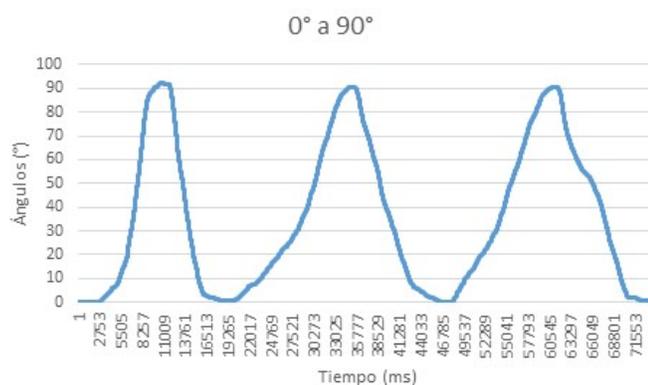


FIGURA 4.3. Datos capturados para el experimento de resolución.

Estos datos obtenidos son realmente interesantes, porque suponemos que la

precisión del guante estará asociada a esta resolución.

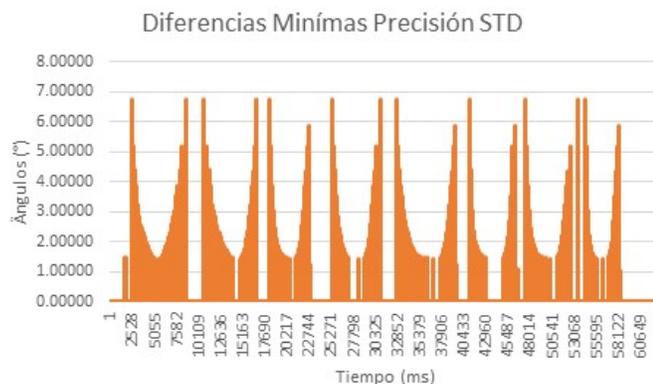


FIGURA 4.4. Diferencias angulares entre la lectura  $i$  y la lectura  $i+1$  durante 1 minuto para la precisión estándar.



FIGURA 4.5. Diferencias angulares entre la lectura  $i$  y la lectura  $i+1$  durante 1 minuto para la precisión mejorada.

## 4.2. Precisión

El sistema propone una alta precisión en los valores angulares entregados por los sensores, por lo que fue indispensable un procedimiento experimental que valide el trabajo realizado en este proyecto explicado anteriormente en el capítulo tres. Con el fin de comparar los valores de la precisión ofrecida por las librerías estándar y

la precisión ofrecida por el método propuesto, se realizó un experimento que nos permitió conocer los valores de los dos diferentes sensores, uno con el código de la librería estándar por su puesto, y el segundo con la precisión mejorada.

#### 4.2.1. Precisión de los sensores

Este experimento consistió en leer datos de cada sensor durante periodos de un minuto, realizando movimientos entre 0 y 90 grados. Los dos sensores fueron colocados en la misma posición sobre una placa de pruebas, por lo tanto, los movimientos fueron similares para cada uno, lo que significa que los valores angulares son prácticamente los mismos. Sin embargo el propósito de este procedimiento es conocer el momento en que se desfasan estos valores y las diferencias entre ellos.

El valor angular de los dos sensores se tomó con respecto al tiempo del procesador de la computadora, el proceso continuó durante un minuto aproximadamente, el proceso se ejecutaba con la escena virtual, tomando el valor del tiempo y las mediciones angulares de cada sensor, estos datos fueron almacenados para su posterior análisis. Para los efectos de visualización los valores se representaron en dos sistemas de referencia en el entorno virtual que representan las rotaciones de cada sensor. En la figura 4.6 se muestra la colocación de los sensores y los movimientos que se realizan.

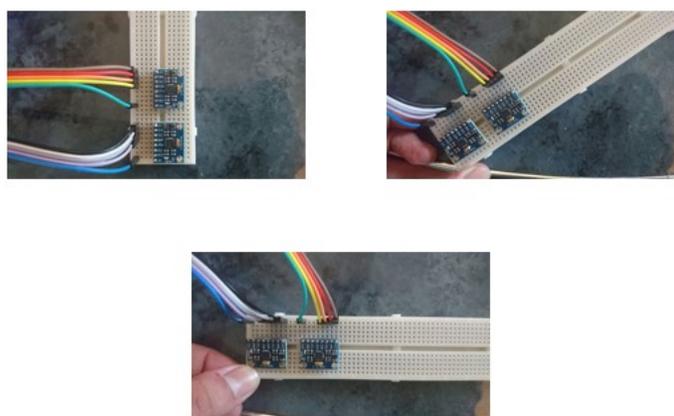


FIGURA 4.6. Movimientos realizados durante el experimento. Colocación de los sensores en la placa de pruebas.

Estos valores se analizaron y graficaron y se presentan en las figuras 4.7 - 4.10 donde se puede observar el comportamiento de los valores. El color naranja corresponde a la precisión de la biblioteca estándar y color azul para la precisión mejorada.

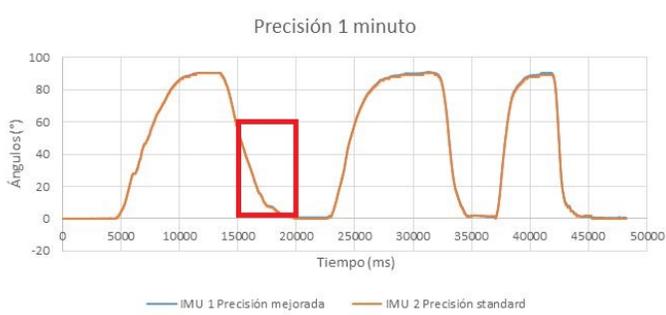


FIGURA 4.7. Movimientos realizados durante un minuto.

La figura 4.7 muestra las mediciones de los ángulos de los dos sensores con respecto al tiempo, con una duración aproximada de 1 minuto. Se observa cómo las dos líneas siguen una trayectoria similar. Sin embargo, cuando nos concentramos y ampliamos los datos trazados, comenzamos a ver las diferencias de lectura entre los valores reportados por la biblioteca estándar y la precisión mejorada.

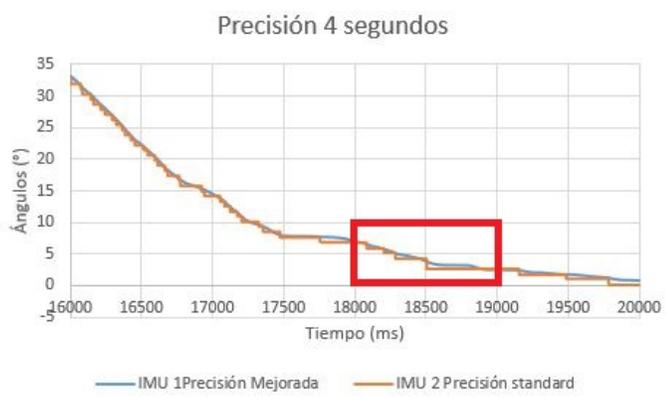


FIGURA 4.8. Lectura de datos de 4 segundos (tomado de los datos de 1 minuto).

La figura 4.8 muestra que al acercar entre los segundos 16 y 20 (del minuto tomado) comenzamos a notar pequeñas diferencias entre las trayectorias, apareciendo

pequeños pasos entre los valores de los diferentes sensores.

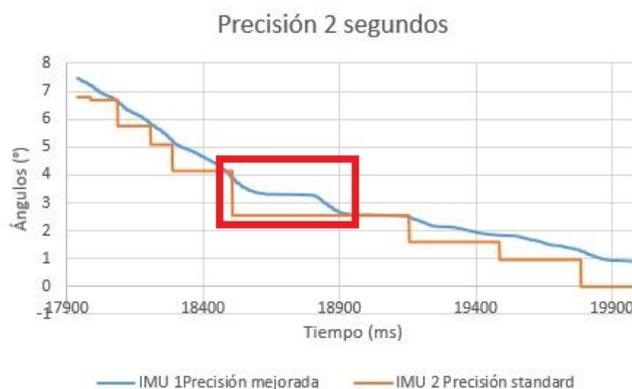


FIGURA 4.9. Lectura de datos de 2 segundos (tomado de los datos de 4 segundos).

Si seguimos haciendo zoom, en la figura 4.9, en la que se tomaron los valores de 2 segundos, se nota la diferencia entre las mediciones entre la IMU con el caso de precisión mejorado 1, frente a la IMU con la precisión estándar, caso 2.

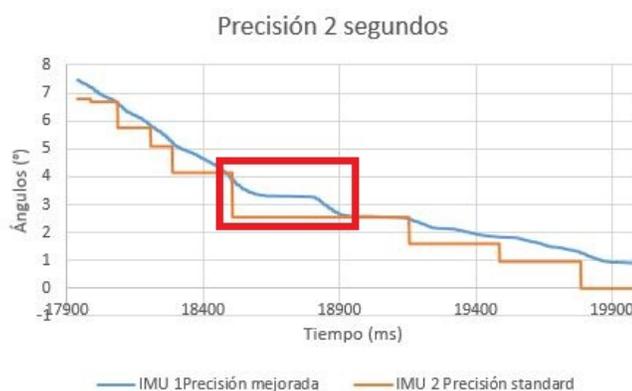


FIGURA 4.10. Lectura de datos de 1 segundo (tomado de los datos de 2 segundos).

La Figura 4.10 muestra el desajuste completo entre las lecturas de los dos sensores en 1 segundo transcurrido. Esta gráfica representa 1 segundo de los 60 analizados, si observamos las figuras 4.7 - 4.10, en las primeras siguen trayectorias similares, se observa que en la última figura estas trayectorias cambian de forma plana entre los

dos sensores.

## 4.3. Manipulación de objetos virtual

### 4.3.1. Detección de colisiones

Con la finalidad de determinar si la mano virtual está tocando un objeto de la escena virtual, se implementó un método de detección de colisiones en este sistema. Dicho método es conocido como *Proximity Query Package* (PQP) que permite detectar la distancia mínima y si existe colisión entre dos modelos 3D.

Durante cada actualización de la escena virtual, se realizan consultas al sistema de detección de colisiones por pares de objetos. Por lo tanto, se realizan 15 consultas para determinar cuáles falanges están colisionando con un objeto virtual.

Debido a que podemos determinar la distancia de proximidad entre cada par de modelos 3D, podemos determinar cuál es el objeto más cercano a la mano, y de esta manera saber cuál es el que se desea agarrar.

### 4.3.2. Agarrado del objeto (Grasping)

Cuando existen colisiones entre las falanges y un objeto en la escena, pueden cumplirse ciertas condiciones que permitan el agarrado (Grasping) del objeto. Las condiciones definidas para el *agarrado* son las siguientes:

- La mano debe estar libre (no debe estar agarrando ningún objeto).
- La mano debe de entrar en colisión con el objeto.
- Al menos un elemento del dedo pulgar y un elemento de cualquier otro dedo debe de estar colisionando con el objeto.

Al cumplirse las 3 condiciones mencionadas anteriormente, el objeto pasa al estado de *agarrado*.

Cuando el objeto se encuentra en estado agarrado, sus coordenadas geométricas siguen a las coordenadas geométricas de la palma de la mano.

Para lograr lo anterior es necesario obtener la matriz de transformación del objeto con respecto al marco de referencia de la palma de la mano:  ${}^{Palma}M_{O1}$ . Esto lo obtenemos multiplicando la inversa de la matriz de transformación de la palma con respecto al marco del sistema de coordenadas del mundo virtual  ${}^{Palma}M_{O1}^{-1}$  por la matriz de transformación del objeto con respecto al sistema de coordenadas del mundo virtual  ${}^WM_{O1}$ , como se muestra en las ecuaciones 4.1 y 4.2:

$${}^{Palma}M_{O1} = {}^{Palma}M_{O1}^{-1} * {}^WM_{O1}, \quad (4.1)$$

$${}^{Palma}M_{O1} = {}^{Palma}M_W * {}^WM_{O1} \quad (4.2)$$

Una vez que hemos obtenido  ${}^{Palma}M_{O1}$ , podemos desplazar la mano virtual, aplicando transformaciones de rotación y translación, y obtener las nuevas coordenadas geométricas del objeto en su estado de **agarrado**:

$${}^WM_{O1} = {}^WM_{O1} * {}^{Palma}M_{O1} \quad (4.3)$$

Lo anterior nos asegura que el objeto seguirá las coordenadas geométricas de la mano virtual mientras éste esté siendo agarrado.

### 4.3.3. Procedimiento

Este estudio experimental tiene como objetivo evaluar el guante de datos prototipo y su efecto sobre el rendimiento del usuario en una tarea de manipulación de objetos en un entorno virtual, de la misma manera se realiza la comparación con un

guante bien posicionado comercialmente, CyberGlove de CyberGlove Systems. En las figuras 4.11- 4.12 se observa a los usuarios con el guante de datos prototipo.

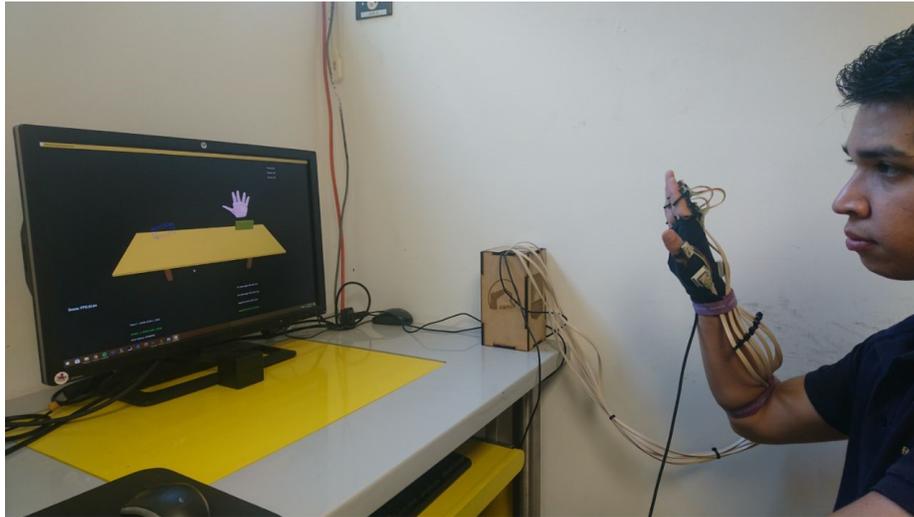


FIGURA 4.11. Sujetos con el guante de datos prototipo durante el experimento.



FIGURA 4.12. Sujetos con el guante de datos prototipo durante el experimento.

La configuración experimental se basa en el concepto de diseño de medidas repetidas, compensado por las condiciones. Se definieron dos condiciones experimentales: C1 y C2. El usuario ejecuta la tarea de manipulación de objetos virtual (mover un

objeto virtual de posición sobre la mesa) con el guante CyberGlove (C1) y luego ejecuta la misma tarea con el guante prototipo (C2).

En el entorno virtual, la escena se construyó a partir de aproximadamente 6K polígonos, mantenidos a una velocidad de cuadro de 50 fps. La versión actual del sistema se utiliza para la interacción de escritorio con la escena virtual visualizada en un monitor de 24 pulgadas. La mano del usuario se rastrea usando un sistema Fastrack POLHEMUS<sup>TM</sup> que rastrea los movimientos físicos de la mano del usuario.

El experimento consiste en agarrar y mover la pieza sobre la mesa, a la nueva posición que es indicada también sobre la mesa. El tiempo es registrado al completar la tarea y se registra para el análisis del rendimiento del usuario.

Veinte sujetos de la comunidad de laboratorio sin experiencia previa con sistemas de realidad virtual participaron en el experimento. Hombres y mujeres, diestros y de entre 19 y 30 años. El mismo grupo de sujetos participó en las dos condiciones experimentales (C1 y C2) para llevar a cabo la prueba de manipulación. El intervalo de tiempo entre la ejecución de diferentes experimentos en diferentes condiciones es de 1 semana.

**Experimento C1** El usuario realiza la tarea de manipulación de objetos con el guante CyberGlove, él realiza la tarea guiándose con sus ojos, cabe señalar que no existe retroalimentación de fuerza, solo un cambio de color a los elementos de la mano que colisionen con el objeto a manipular. Los sujetos recibieron una pequeña capacitación y explicación de los ambientes virtuales, realizando pequeños movimientos para familiarizarse con estos ambientes. El experimentador explica el objetivo del experimento y proporciona al sujeto con una descripción detallada de la operación a realizar.

Cuando el sujeto está listo para hacer la prueba, el experimentador da la señal de inicio y observa al sujeto ejecutando la tarea de manipulación. La aplicación registra los tiempos de interés en segundos como tiempo de duración de la tarea (Tabla 4.1),

tiempo de duración de agarrado de la pieza (Tabla 4.2) y tiempo de localización del objeto en el lugar a colocar (Tabla 4.3), otros datos también fueron registrados como el error en la distancia entre el centro de la pieza y el centro de la posición donde debe quedar colocado (Tabla 4.4).

**Experimento C2** Al mismo grupo de sujetos que participaron en el experimento anterior se les pide que lleven a cabo la tarea de manipulación con el guante prototipo. Los tiempos de interés también son registrados para cada sujeto (Tabla 4.1 – 4.4).

TABLA 4.1. Datos registrados para la realización de la tarea. TTC (Tiempo de Tarea Completa).

Sujeto	TTC:C1 (seg)	TTC:C2 (seg)
1	9.02	12.63
2	6.18	9.29
3	10.81	9.29
4	7.81	9.13
5	4.78	4.68
6	12.57	17.23
7	18.25	17.38
8	28.29	12.16
9	16.38	7.8
10	5.94	9.02
11	8.78	7.16
12	10.36	8.34
13	7.75	12.84
14	9.91	9.78
15	13.04	11.17
16	5.3	6.19
17	13.6	10.75
18	14.34	10.61
19	16.24	16.96
20	17.77	22.37

---

TABLA 4.2. Datos registrados para el tiempo de agarrado. TA (Tiempo de Agarrado).

Sujeto	TA:C1 (seg)	TA:C2 (seg)
1	8.98	8.32
2	6.16	7.56
3	10.7	9.26
4	7.8	8.85
5	4.76	4.52
6	12.56	17.18
7	7.21	8.53
8	28.12	10.36
9	11.34	7.78
10	5.85	8.52
11	8.77	7.14
12	10.34	8.33
13	7.73	12.39
14	6.26	6.8
15	8.63	9.02
16	5.29	6.17
17	11.92	7.7
18	14.31	10.6
19	16.23	16.95
20	17.75	21.71

---

TABLA 4.3. Datos registrados para el tiempo de localización. TL (Tiempo de Localización).

Sujeto	TL:C1 (seg)	TL:C2 (seg)
1	3.24	3.05
2	3.17	3.61
3	4.27	4.69
4	4.51	6.61
5	1.75	1.94
6	8.34	9.95
7	2.3	3.29
8	17.5	7.49
9	5.87	4.22
10	0.95	4.77
11	3.11	3.72
12	6.12	4.93
13	2.36	6.4
14	1	2.42
15	2.5	3.42
16	2.81	2.19
17	8.87	0.25
18	8.31	4.8
19	6.91	7.97
20	10.16	9.96

---

TABLA 4.4. Datos registrados para el error en la distancia. ED (Error en la Distancia).

Sujeto	ED:C1 (cm)	ED:C2 (cm)
1	0.612109	0.368244
2	4.144261	2.494955
3	1.261419	1.76955
4	1.850312	1.028981
5	1.583594	4.872068
6	1.84106	2.689545
7	4.386101	3.182043
8	1.715654	1.489759
9	2.646351	4.491447
10	6.824642	3.728026
11	0.694735	2.562957
12	0.765037	3.222129
13	1.059677	3.249237
14	5.787786	2.716308
15	2.93274	0.696589
16	0.825119	1.025727
17	2.460582	4.574279
18	4.064777	4.898637
19	1.036735	0.465776
20	2.706861	2.136727

**Análisis de datos y resultados** Para probar que en el guante prototipo no existe diferencia significativa con respecto a CyberGlove en el rendimiento del usuario, los datos recopilados de los experimentos C1 y C2 se utilizaron para realizar análisis **ANOVA** en el tiempo de la tarea completa (TTC) y factores de la condición experimental. Los resultados no muestran un efecto principal significativo de la condición experimental en el tiempo de finalización de la tarea con un nivel de confianza del 95 % [ $F(1, 39) = 0.31, P < 0.05$ ].

De la misma manera se realizó un análisis **ANOVA** entre los tiempos de duración del agarrado del objeto (TA) y los factores de la condición experimental. Los resultados tampoco muestran un efecto significativo de la condición experimental en el

tiempo de agarrado del objeto con un nivel de confianza del 95 % [ $F(1, 39) = 0.37, P < 0.05$ ].

El análisis **ANOVA** también se llevó a cabo en el tiempo de localización del objeto (TL) entre los experimentos C1 y C2, donde los resultados no muestran una diferencia significativa con la condición experimental en el tiempo de localización del objeto con un nivel de confianza del 95 % [ $F(1, 39) = 0.29, P < 0.05$ ].

Por último, se realizó un análisis **ANOVA** entre los errores en la distancia de colocación del centro del objeto (ED) y los factores de la condición experimental. Los resultados no muestran un efecto significativo de la condición experimental en los errores en la distancia del objeto con un nivel de confianza del 95 % [ $F(1, 39) = 0.09, P < 0.05$ ].

El estudio de las condiciones experimentales y los resultados estadísticos confirman que no existe una diferencia significativa sobre el tiempo de manipulación de un objeto simple en la realidad virtual, respecto al guante prototipo y CyberGlove.

**Evaluación subjetiva** Para recibir comentarios de los usuarios después de su participación a los experimentos, un procedimiento de evaluación subjetiva fue usado. A cada participante se le solicitó completar un cuestionario informando sobre sus opiniones sobre los modos de interacción C1 y C2. Debido a como está elaborado el prototipo los usuarios reportaron que fue incómoda la colocación del guante, y debido a que era un solo tamaño existía variación en las manos de los usuarios. 11 de 20 usuarios piensan que el movimiento del prototipo inercial es más fluido, y que el movimiento de CyberGlove es más mecanizado. Todos los usuarios comentaron que fue más cómodo la colocación y descolocación de CyberGlove. 7 sujetos comentaron que sintieron más preciso el guante prototipo en cuanto a la relación del movimiento de los dedos y los elementos virtuales. 15 de los 20 notificaron que fue difícil guiarse en la escena virtual sin otro tipo de guías o retroalimentaciones.

Entre los comentarios de los sujetos se encontró que la versión de este prototipo

es incómoda con respecto a su competidor, que tal vez pueda mejorar con otro diseño, en cuanto a la precisión y fluidez de los movimientos, no hay una diferencia entre el número de usuarios que piense que el prototipo sea mejor o sea peor que CyberGlove.

# CAPÍTULO 5

## CONCLUSIONES

### 5.1. Conclusiones

Como se muestra en experimentos anteriores, la captura de movimiento basada en sensores inerciales nos da una buena frecuencia en términos de lectura de datos. La prueba realizada para 1 y 2 sensores fue satisfactoria y no afectó en absoluto con más de 1 sensor alimentando la aplicación. En cuanto a la resolución que nos ofrece la IMU, afirmamos que es muy buena, ya que el resultado del experimento fue de 0.00001 grados, haciendo que los sensores detecten movimientos muy finos.

La deriva que se presentó durante la fase de prueba puede considerarse aceptable para la realización de tareas de pocos minutos, ya que en un lapso de 20 minutos se obtiene un margen de error menor de 1 grado. Además, es interesante profundizar en el tema, para determinar si la aplicación de algún filtro de datos puede ayudar a reducir o eliminar la deriva.

Con la medición de los Hertz, nos permitió asumir que la comunicación de datos desde el sensor al ordenador es mayor que la velocidad de procesamiento de datos en el ordenador con la escena virtual. Al realizar las pruebas con uno o dos sensores conectados a la placa Arduino, se encontró que la frecuencia de llenado era la misma de acuerdo a la forma en que los datos son enviados por el puerto serial, lo que nos da un amplio rango para trabajar en la escena virtual. La frecuencia fue aproximada a 100 hertz.

La realización del proceso experimental de manipulación de objetos virtual permitió validar el funcionamiento y verificar lo que los otros experimentos habían arrojado como resultado. La prueba fue simple, colocar en la escena virtual una mesa para el usuario, poner un objeto a la derecha de la mesa y pedir que lo colocase en otra

posición en la parte izquierda de la mesa. Se tomaron los datos permitientes al proceso para su posterior análisis y verificar que el diseño del guante de datos prototipo es tan funcional como los principales guantes de datos en el mercado.

Los análisis ANOVA que se aplicaron a los tiempos y a las distancias arrojaron que no hubo diferencia significativa entre la tarea realizada con el CyberGlove y el guante de datos prototipo desarrollado en este proyecto.

## 5.2. Trabajo futuro

Como trabajo futuro sería realizar más experimentos en términos de derivación y frecuencia. También es necesario realizar tareas de ensamble de objetos, y validar la aplicabilidad del guante en comparación con otros dispositivos existentes, ya sean comerciales o experimentales.

Elaborar un diseño mecánico que permita trabajar con diferentes tamaños de las manos y ampliar los sensores a las falanges distales de los dedos de la mano.

## 5.3. Producción

- Campos-Leal, J. A. Zaldivar-Colado, U. Zaldivar-Colado, X. P. Diseño de un guante de datos basado en sensores inerciales. Memorias del XVIII Congreso Mexicano de Robótica 2016, Universidad Autónoma de Sinaloa y Asociación Mexicana de Robótica e Industria AC. [Internet] 2016.
- Estancia internacional en el Laboratorio END-ICAP: Evolution of Neuromuscular Diseases: Innovative Concepts and Practices (URM 1179) de la Universidad de Versailles, Francia, 1 de septiembre de 2016 a 28 de febrero de 2017, Dr Samir Garbaya
- Séminaire de l'unité U1179 UVSQ-INSERM Handicap Neuromusculaire : Physiologie, Biothérapie et Pharmacologie appliquées, Équipe 3 Dr. Marcel Bonay,

1 decembre 2016, Le ferme du manet,France

- U. Zaldivar-Colado, J. Campos-Leal, S. Garbaya, X. Zaldívar-Colado, P. Blazevic, "Design of a high precision Data Glove based on inertial sensors,in XIX Congreso Mexicano de Robótica (Comrob 2017), 2017, pp. 283-289. ***Nominado a Premio Dr. Rafael Kelly 2017.***
- Artículo listo para enviar a RO-MAN 2018 del 27 de agosto al 1 de septiembre nanjing, China

## BIBLIOGRAFÍA

- [Abe et al., 1996] Abe, N., Zheng, J. Y., Tanaka, K., and Taki, H. (1996). A training system using virtual machines for teaching assembling/disassembling operation to novices. In *Systems, Man, and Cybernetics, 1996., IEEE International Conference on*, volume 3, pages 2096–2101. IEEE.
- [Antonya and Talaba, 2007] Antonya, C. and Talaba, D. (2007). Design evaluation and modification of mechanical systems in virtual environments. *Virtual Reality*, 11(4):275–285.
- [ArduinoBoardMega2560, 2015] ArduinoBoardMega2560 (2015). [Arduinoboardmega2560 arduino.cc](http://www.arduino.cc).
- [ArduinoMPU6050, 2015] ArduinoMPU6050 (2015). [Playground - mpu-6050 playground.arduino.cc](http://www.arduino.cc).
- [ArduinoSoftware, 2015] ArduinoSoftware (2015). [Software arduino.cc](http://www.arduino.cc).
- [Borst and Indugula, 2005] Borst, C. W. and Indugula, A. P. (2005). Realistic virtual grasping. In *Virtual Reality, 2005. Proceedings. VR 2005. IEEE*, pages 91–98. IEEE.
- [Borst and Indugula, 2006] Borst, C. W. and Indugula, A. P. (2006). A spring model for whole-hand virtual grasping. *Presence: Teleoperators and Virtual Environments*, 15(1):47–61.
- [Bowman et al., 2001] Bowman, D., Wingrave, C., Campbell, J., and Ly, V. (2001). Using pinch gloves (tm) for both natural and abstract interaction techniques in virtual environments.

- [Carrillo, 2007] Carrillo, H. O. (2007). Sistema de captura opto mecánico para la medición de movimientos del cuerpo humano. *Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (UNAM)*.
- [Cloete and Scheffer, 2008] Cloete, T. and Scheffer, C. (2008). Benchmarking of a full-body inertial motion capture system for clinical gait analysis. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 4579–4582. IEEE.
- [Denavit and Hartenberg, 1964] Denavit, J. and Hartenberg, R. (1964). An iterative method for tie displacement analysis of spatial mechanisms.
- [Díaz-Tribaldos et al., 2015] Díaz-Tribaldos, M. R., Escobar-Ocampo, J. M., and Vivas-Albán, s. A. (2015). Interfaz háptica tipo guante con realimentación vibratoria. *Revista EIA*, 12(23):29–39.
- [Fei et al., 2014] Fei, Y., Song, Y., Xu, L., and Sun, G. (2014). Micro-imu based wireless body sensor network. In *Control Conference (CCC), 2014 33rd Chinese*, pages 428–432. IEEE.
- [Fuchs and Moreau, 2006] Fuchs, P. and Moreau, G. (2006). *Le traité de la réalité virtuelle*, volume 2. Presses des MINES.
- [Garbaya and Zaldivar-Colado, 2009] Garbaya, S. and Zaldivar-Colado, U. (2009). Modeling dynamic behavior of parts in virtual assembly environment. (43376):89–99. 10.1115/WINVR2009-742.
- [Han, 2010] Han, Y. (2010). A low-cost visual motion data glove as an input device to interpret human hand gestures. *IEEE Transactions on Consumer Electronics*, 56(2).

- [Hsiao et al., 2015] Hsiao, P.-C., Yang, S.-Y., Lin, B.-S., Lee, I.-J., and Chou, W. (2015). Data glove embedded with 9-axis imu and force sensing sensors for evaluation of hand function. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pages 4631–4634. IEEE.
- [InvenSense, 2015] InvenSense (2015). Mpu6050 datasheet.
- [Iwata, 1990] Iwata, H. (1990). Artificial reality with force-feedback: Development of desktop virtual space with compact master manipulator. *ACM SIGGRAPH Computer Graphics*, 24(4):165–170.
- [Jayaram et al., 1997] Jayaram, S., Connacher, H. I., and Lyons, K. W. (1997). Virtual assembly using virtual reality techniques. *Computer-aided design*, 29(8):575–584.
- [Kessler et al., 1995] Kessler, G. D., Hodges, L. F., and Walker, N. (1995). Evaluation of the cyberglove as a whole-hand input device. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 2(4):263–283.
- [Kim et al., 2009] Kim, S.-K., Hong, S., and Kim, D. (2009). A walking motion imitation framework of a humanoid robot by human walking recognition from imu motion data. In *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, pages 343–348. IEEE.
- [Kitagawa and Windsor, 2012] Kitagawa, M. and Windsor, B. (2012). *MoCap for artists: workflow and techniques for motion capture*. CRC Press.
- [Kitamura et al., 2009] Kitamura, Y., Higashi, T., Masaki, T., and Kishino, F. (2009). Virtual chopsticks: Object manipulation using multiple exact interactions. In *Virtual Reality, 1999. Proceedings., IEEE*, pages 198–204. IEEE.
- [Koenemann and Bennewitz, 2012] Koenemann, J. and Bennewitz, M. (2012). Whole-body imitation of human motions with a nao humanoid. In *Human-Robot*

- Interaction (HRI)*, 2012 7th ACM/IEEE International Conference on, pages 425–425. IEEE.
- [Koenemann et al., 2014] Koenemann, J., Burget, F., and Bennewitz, M. (2014). Real-time imitation of human whole-body motions by humanoids. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2806–2812. IEEE.
- [Kuroda et al., 2004] Kuroda, T., Tabata, Y., Goto, A., Ikuta, H., and Murakami, M. (2004). Consumer price data-glove for sign language recognition. In *Proc. of 5th Intl Conf. Disability, Virtual Reality Assoc. Tech., Oxford, UK*, pages 253–258.
- [Li et al., 2016] Li, J., Li, J., Xu, Y., Xu, Y., Ni, J., Ni, J., Wang, Q., and Wang, Q. (2016). Glove-based virtual hand grasping for virtual mechanical assembly. *Assembly Automation*, 36(4):349–361.
- [Lin et al., 2014] Lin, B. S., Lee, I. J., Hsiao, P. C., Yang, S. Y., and Chou, W. (2014). Data glove embedded with 6-dof inertial sensors for hand rehabilitation. In *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2014 Tenth International Conference on*, pages 25–28. IEEE.
- [Lizárraga Reyes, 2010] Lizárraga Reyes, J. (2010). *Desarrollo y Evaluación de una Técnica de Interacción para el Agarrado y Manipulación de Objetos en Ensamble Virtual*. Tesis de maestría en informática administrativa, Universidad de Occidente.
- [Luinje and Veltink, 2005] Luinje, H. J. and Veltink, P. H. (2005). Measuring orientation of human body segments using miniature gyroscopes and accelerometers. *Medical and Biological Engineering and computing*, 43(2):273–282.
- [Mihelj et al., 2014] Mihelj, M., Novak, D., and Beguš, S. (2014). *Virtual reality technology and applications*. Springer.

- [Nakaoka et al., 2003] Nakaoka, S., Nakazawa, A., Yokoi, K., Hirukawa, H., and Ikeuchi, K. (2003). Generating whole body motions for a biped humanoid robot from captured human dances. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 3, pages 3905–3910. IEEE.
- [Popelka, 2014] Popelka, V. (2014). A self stabilizing platform. In *Control Conference (ICCC), 2014 15th International Carpathian*, pages 458–462. IEEE.
- [Rodríguez Vega, 2016] Rodríguez Vega, D. A. (2016). *Teleoperación en línea del Robot Humanoide Bioloid Premium tm mediante captura de movimientos*. Tesis de doctorado en ciencias de la información, Universidad Autónoma de Sinaloa.
- [Roetenberg et al., 2009] Roetenberg, D., Luinge, H., and Slycke, P. (2009). Xsens mvn: full 6dof human motion tracking using miniature inertial sensors. *Xsens Motion Technologies BV, Tech. Rep.*
- [Roetenberg et al., 2005] Roetenberg, D., Luinge, H. J., Baten, C. T., and Veltink, P. H. (2005). Compensation of magnetic disturbances improves inertial and magnetic sensing of human body segment orientation. *IEEE Transactions on neural systems and rehabilitation engineering*, 13(3):395–405.
- [Rowberg, 2015] Rowberg, J. (2015). I2cdevlib for c++-based mcus.
- [Sato et al., 2011] Sato, T., Nakamura, S., Terabayashi, K., Sugimoto, M., and Hashizume, H. (2011). Design and implementation of a robust and real-time ultrasonic motion-capture system. In *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, pages 1–6. IEEE.
- [Sturman and Zeltzer, 1994] Sturman, D. J. and Zeltzer, D. (1994). A survey of glove-based input. *IEEE Computer graphics and Applications*, 14(1):30–39.
- [Sutherland, 1965] Sutherland, I. E. (1965). The ultimate display. *Multimedia: From Wagner to virtual reality*.

- [Tamayo Serrano, 2016] Tamayo Serrano, P. (2016). *Interacción natural con realidad aumentada para la realización de tareas de manipulación y ensamble virtual*. Tesis de maestría en ciencias de la información, Universidad Autónoma de Sinaloa.
- [Tavares et al., 2016] Tavares, R., Souse, P. J., Abreu, P., and Restivo, M. T. (2016). Virtual environment for instrumented glove. In *Remote Engineering and Virtual Instrumentation (REV), 2016 13th International Conference on*, pages 311–312. IEEE.
- [Tung and Van Thanh, 2015] Tung, H. T. and Van Thanh, V. (2015). Human-computer interaction based on imu sensors. *JOURNAL OF SCIENCE AND TECHNOLOGY*, page 71.
- [Ullmann and Sauer, 2015] Ullmann, T. and Sauer, J. (2015). Intuitive virtual grasping for non haptic environments. In *Computer Graphics and Applications, 2000. Proceedings. The Eighth Pacific Conference on*, pages 373–457. IEEE.
- [Vlasic et al., 2007] Vlasic, D., Adelsberger, R., Vannucci, G., Barnwell, J., Gross, M., Matusik, W., and Popović, J. (2007). Practical motion capture in everyday surroundings. In *ACM transactions on graphics (TOG)*, volume 26, page 35. Acm.
- [Wingrave et al., 2010] Wingrave, C. A., Williamson, B., Varcholik, P. D., Rose, J., Miller, A., Charbonneau, E., Bott, J., and LaViola Jr, J. J. (2010). The wiimote and beyond: Spatially convenient devices for 3d user interfaces. *IEEE Computer Graphics and Applications*, 30(2):71–85.
- [XsensTechnologies, 2015] XsensTechnologies, X. (2015). Xsens technologies.
- [Yalzin, 2014] Yalzin, B. C. (2014). Design of a low cost motion data acquisition setup for mechatronic systems. *World Academy of Science, Engineering and Technology, International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, 8(7):1211–1214.

- [Zaldívar Colado, 2009] Zaldívar Colado, U. (2009). *Planification d'assemblage en environnement virtuel*. Thesis, Versailles-St Quentin en Yvelines.
- [Zaldívar-Colado et al., 2006] Zaldívar-Colado, U., Garbaya, S., and Blazevic, P. (2006). Spring-damper model for parts mating in virtual assembly environment. In *proceedings of International Symposium On Robotics and Automation (IEEE-ISRA 2006)*.
- [Zaldívar-Colado et al., 2017] Zaldívar-Colado, U., Garbaya, S., Tamayo-Serrano, P., Zaldívar-Colado, X., and Blazevic, P. (2017). A mixed reality for virtual assembly. In *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 739–744.
- [Zaldívar-Colado et al., 2010] Zaldívar-Colado, U., León-Sánchez, C. A., Guerrero-Cárdenas, S. J., Fregoso-Guerrero, V. M., Zaldívar-Colado, X., Murillo-Campos, D., and Martínez-Tirado, C. G. (2010). Sistema de transmisión y captura de movimientos de los dedos de la mano. In *XII Congreso Mexicano de Robótica (Comrob 2010)*, pages 238–242.
- [Zaldívar-Colado et al., 2010] Zaldívar-Colado, U., Lizárraga-Reyes, J., Garbaya, S., Zaldívar-Colado, X., Murillo-Campos, D., and Martínez-Tirado, C. G. (2010). Técnica de interacción para el agarrado y manipulación de objetos en ensamble virtual. In *XII Congreso Mexicano de Robótica (Comrob 2010)*, pages 221–226.
- [Zaldívar-Colado et al., 2011] Zaldívar-Colado, U., Zaldívar-Colado, X., Marmolejo-Rivas, C., León-Sánchez, C. A., Guerrero-Cárdenas, S. J., Murillo-Campos, D., and Martínez-Tirado, C. G. (2011). Manipulación dinámica de objetos utilizando un sistema de captura de movimientos de los dedos de la mano aplicando el modelo de resorte-amortiguador. In *Memorias del XIII Congreso Mexicano de Robótica, COARA, UASLP, Matehuala, SLP, México*, pages 19–22.

[Zannatha et al., 2011] Zannatha, J. M. I., Zaldivar Colado, U., Colado, U. Z., Sánchez, C. A. L., Cárdenas, S. J. G., Guerrero, V. M. F., Colado, X. Z., Campos, D. M., and Martínez, C. G. (2011). Seguimiento visual de las manos y dedos para actualizar un mundo virtual interactivo. In *XIV Convención de Ingeniería Eléctrica (CIE 2011)*.

[Zhao et al., 2004] Zhao, X., Huang, Q., Du, P., Wen, D., and Li, K. (2004). Humanoid kinematics mapping and similarity evaluation based on human motion capture. In *Information Acquisition, 2004. Proceedings. International Conference on*, pages 426–431. IEEE.