



UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGENIERÍAS Y ARQUITECTURA

TELEOPERACIÓN DE UN BRAZO ROBÓTICO ANTROPOMÓRFICO POR MEDIO DE UN SISTEMA DE LUZ ESTRUCTURADA

TESIS PRESENTADA POR YEISON PEREA CANTILLO
PARA OBTENER EL GRADO DE INGENIERO EN MECATRÓNICA

2017

Departamento de Ingenierías Mecánica, Mecatrónica e Industrial





TELEOPERACIÓN DE UN BRAZO ROBÓTICO ANTROPOMÓRFICO POR MEDIO DE UN SISTEMA DE LUZ ESTRUCTURADA

YEISON PEREA CANTILLO

E-MAIL: ypereac@hotmail.com

Director: César Augusto Peña Cortés

PhD. Automática y Robótica

e-mail: cesarapc@unipamplona.edu.co

Universidad de Pamplona

Facultad de Ingenierías y Arquitectura

Departamento de Ingenierías Mecánica, Mecatrónica e Industrial

Ingeniería Mecatrónica

Pamplona N.S., Colombia

2017



Agradecimientos

Este trabajo se lo dedico a Dios ya que él se merece toda honra y alabanza por darme la fortaleza y la sabiduría en los momentos difíciles y sobre todo mantener siempre la calma, a mis amados padres Osmar Perea y Cecilia cantillo, por su paciencia y apoyo en todos los momentos de mi carrera, por siempre estar hay sin importar lo duro del camino siempre estuvieron brindándome esa mano de apoyo para no dejarme caer, a mis hermanos Mónica, Carina y Osmar E. por ser mi fuente de inspiración y superación. Un agradecimiento enorme a mis tíos por siempre estar ahí y alentándome a seguir adelante por regalarme siempre una voz de apoyo y confianza, por llevarme siempre en el camino correcto. A Leidy Matilde por ser esa persona muy especial en mi largo camino, por ser esa voz de aliento, esa gota de energía cuando se necesitaba, muchas gracias mor.

A mis profesores por enseñarme todo lo que se, por su paciencia y dedicación en todo momento y de llenar de sabiduría a todos los que lo necesitamos, un agradecimiento especial al Doctor Cesar Peña que ha sido mi guía para escribir este trabajo y aportar sus vastos conocimientos siempre que fue necesario.

Muchas pero muchas gracias a todas estas personas, ya que sin ellos yo no sería lo que soy ahora y no estuviera escribiendo estas bellas palabras para ustedes. Gracias por estar hay en mis victorias pero sobre todo gracias por seguir hay en mis derrotas.

Resumen

InMoov es un robot humanoide diseñado por el escultor francés Gaël Langevin con propósitos artísticos, el brazo robótico de InMoov cuenta con de 10 GDL. 6 de esos movimientos son para los dedos y el movimiento de muñeca, los otros 4 GDL son para el movimiento del hombro y el movimiento del codo, para la teleoperación del brazo robótico se usa el sistema de luz estructurada Kinect que permite mostrar el esqueleto en 2D o 3D del usuario imitando los movimientos que este haga. Utilizando cálculos geométricos se calculan los ángulos correspondientes a cada articulación mostrada por Kinect, posteriormente enviada a la cinemática que calcula las matrices de transformación homogénea de cada una de las articulaciones y la posición del efector final con respecto al origen del brazo robótico.

En este trabajo se muestra la construcción del brazo robótico de InMoov y mejoras implementadas con el propósito de simular el movimiento de un brazo humano real.

Al final de este trabajo se muestra los Resultados obtenidos del brazo robótico

Índice General

AGRADECIMIENTOS	3
RESUMEN.....	4
ÍNDICE GENERAL.....	5
ÍNDICE DE FIGURAS	7
CAPÍTULO 1.	9
1.1 INTRODUCCIÓN.....	9
1.2. DESCRIPCIÓN DEL PROYECTO.....	10
1.2.1. PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN	10
1.3. OBJETIVOS	10
1.3.1. <i>Objetivo General</i>	10
1.3.2. <i>Objetivos específicos</i>	10
1.4. ALCANCES Y LIMITACIONES	11
CAPÍTULO 2.	12
2.1. MARCO TEÓRICO	12
2.2 ESTADO DEL ARTE	13
CAPÍTULO 3.....	18
IMPLEMENTACIÓN DEL BRAZO ROBÓTICO IM.....	18
3.1 DESCRIPCIÓN DEL BRAZO INMOOV	18
3.2. CONSTRUCCIÓN	18
3.2.1 <i>Ensamble del antebrazo</i>	19
3.2.2. <i>Ensamble del bíceps</i>	22
3.2.3 <i>Ensamble del hombro</i>	28
3.3. MODIFICACIONES.....	29
3.4. IMPLEMENTACIÓN DEL SISTEMA DE CONTROL	33
3.4.1. <i>Actuadores</i>	33
3.4.2. <i>Sistema de control</i>	34
3.4.3 <i>Interfaz grafica</i>	35
CAPÍTULO 4.....	37
ALGORITMO PARA LA TELEOPERACIÓN	37
4.1 MODELO CINEMÁTICO.....	37
4.1.1 <i>modelo cinemático directo (MCD)</i>	37
4.1.2 <i>Modelo Cinemático Inverso (MCI)</i>	41
4.2. INTERPRETACIÓN DE LOS ÁNGULOS.....	46
4.2.1 <i>cambio de eje coordinado</i>	46
4.3 CÁLCULOS DE LOS ÁNGULOS REFERIDOS PARA MOVER EL ROBOT	49
4.4 CONTROL POSICIÓN – POSICIÓN	55
CAPÍTULO 5	57
RESULTADOS	57
CAPÍTULO 6.....	60
CONCLUSIONES.....	60
BIBLIOGRAFÍA	61
APÉNDICE A	63
CONEXIONES ELÉCTRICAS DEL BRAZO ROBÓTICO	63
APÉNDICE B.....	64



CÓDIGO DE OPTIMIZACIÓN DE MATLAB – KINECT	64
<i>Conexión Matlab-Kinect</i>	<i>64</i>
<i>Captura de los puntos de las articulaciones.....</i>	<i>66</i>
<i>Dibujar el esqueleto en 3D.....</i>	<i>68</i>
<i>Ángulos.....</i>	<i>69</i>
APÉNDICE C	70
CINEMÁTICA DIRECTA E INVERSA DEL BRAZO ROBÓTICO INMOOV	70
<i>Cinemática Directa</i>	<i>70</i>
<i>Cinemática Inversa</i>	<i>72</i>

Índice de Figuras

figura 2.2. 1 prótesis robótica con impulsos cerebrales	13
figura 2.2. 2 mobius bionics luke arms	14
figura 2.2. 3 les baugh controlando su brazo robótico modular	14
figura 2.2. 4 brazo biónico que se controla con sensores en los músculos y teléfonos inteligentes	15
figura 2.2. 5 brazo robótico controlado por medio de señales cerebrales	15
figura 2.2. 6 prótesis revolucionaria haptix	16
figura 2.2. 7 brazo biónico	16
figura 2.2. 8 brazo robótico amo	16
figura 2.2. 9 robonaut	17
figura 3.1. 1 robot inmoov	18
figura 3.2.1. 1 jerarquía de las piezas del brazo de inmoov	19
figura 3.2.1. 2 vista 3d del primer sub - ensamble	20
figura 3.2.1. 3 primer sub-ensamble del antebrazo	20
figura 3.2.1. 4 vista 3d sub ensamble 2	20
figura 3.2.1. 5 sub - ensamble 2 tapa superior	20
figura 3.2.1. 6 f vista 3d sub ensamble pronación- supinación de la muñeca	21
figura 3.2.1. 7 sub - ensamble pronación- supinación n de la muñeca	21
figura 3.2.1. 8 vista 3d de robservobed	21
figura 3.2. 1.9 ensamble de robservobed	21
figura 3.2.1, 10 vista 3d sub- ensamble antebrazo de inmoov	22
figura 3.2.1. 11 sub-ensamble antebrazo inmoov	22
figura 3.2.2. 1 modelo 3d del bíceps de inmoov	23
figura 3.2.2. 2 jerarquía de las piezas del bíceps de inmoov	23
figura 3.2.2. 3 vista 3d bíceps	23
figura 3.2.2. 4 ensamble de las piezas correspondientes al bíceps	24
figura 3.2.2. 5 unión entre el ante brazo y el bíceps	24
figura 3.2.2. 6 extracción del circuito	24
figura 3.2.2. 7 segunda forma de la extracción del circuito	25
figura 3.2.2. 8 partes internas del servo hs-805bb	25
figura 3.2.2. 9 truncando uno de los engranes	25
figura 3.2.2. 10 ensamble del bíceps	26
figura 3.2.2. 11 calibración del potenciómetro del codo	26
figura 3.2.2. 13 vista 3d del sub ensamble de torsión	27
figura 3.2.2. 14 ensamble piezas de torsión	27
figura 3.2.2. 15 calibración del ensamble de rotación a 90°	27
figura 3.2.3. 1 jerarquía de las piezas del hombro de inmoov	28
figura 3.2.3. 2 vista 3d movimiento de fl-ex	28
figura 3.2.3. 3 montaje de los ensambles de rotación y fl-ex	29
figura 3.2.3. 4 vista terminada del brazo de inmoov	29
figura 3.2.2. 1 modelo 3d del bíceps de inmoov	23
figura 3.2.2. 2 jerarquía de las piezas del bíceps de inmoov	23
figura 3.2.2. 3 vista 3d bíceps	23
figura 3.2.2. 4 ensamble de las piezas correspondientes al bíceps	24
figura 3.2.2. 5 unión entre el ante brazo y el bíceps	24
figura 3.2.2. 6 extracción del circuito	24
figura 3.2.2. 7 segunda forma de la extracción del circuito	25
figura 3.2.2. 8 partes internas del servo hs-805bb	25
figura 3.2.2. 9 truncando uno de los engranes	25
figura 3.2.2. 10 ensamble del bíceps	26
figura 3.2.2. 11 calibración del potenciómetro del codo	26
figura 3.2.2. 13 vista 3d del sub ensamble de torsión	27
figura 3.2.2. 14 ensamble piezas de torsión	27
figura 3.2.2. 15 calibración del ensamble de rotación a 90°	27
figura 3.3. 1 primera versión del brazo de inmoov sin el movimiento de pronación – supinación	30
figura 3.3. 2 última versión del brazo de inmoov con el movimiento de pronación – supinación	30
figura 3.3. 3 muestra del brazo inmoov con torso incluido (gael, 2017)	31
figura 3.3. 4 modificación hechas por usuario de inmoov	31
figura 3.3. 5 modificación hecha por la página de inmoov sin torso (gael, 2017)	31
figura 3.3. 6 soporte base del brazo	32
figura 3.3. 7 montaje del tornillo sin fin con la base del brazo	32
figura 3.3. 8 brazo de inmoov completo	33

figura 4.1.2. 1 posición de aducción del brazo inmoov	41
figura 4.1.2. 2 cálculo del ángulo q_1	42
figura 4.1.2. 3 cálculo del ángulo q_2	43
figura 4.1.2. 4 cálculo del ángulo q_3	44
figura 4.1.2. 5 cálculo del ángulo q_4	45
figura 4.1.2. 6 cálculo del ángulo q_5	46
figura 4.2.1. 1 vista del esqueleto según el sistema de eje coordenado del kinect (k)	46
figura 4.2.1. 2 corrección del sistema de kinect con respecto al de matlab	48
figura 4.2.1. 3 vista del sistema coordenado con los puntos del esqueleto sin el sistema de kinect	49
figura 4.3. 1 captura del esqueleto completo con numeración de cada articulación	50
figura 4.3. 2 sistema coordenado en el hombro	51
figura 4.3. 3 cálculo del ángulo q_1	52
figura 4.3. 4 cálculo del ángulo q_2	53
figura 4.3. 5 plano para cálculo de q_3	54
figura 4.3. 6 calculo del ángulo q_4	55
figura 4.4. 1 arquitectura del sistema de control	56

Capítulo 1.

1.1 Introducción

La construcción de un brazo robótico con el fin de avanzar en las nuevas tecnologías en Colombia ha llevado a creación de nuevos prototipos funcionales con el propósito de mitigar la pérdida de una extremidad superior. Estas prótesis son capaces de asemejar el funcionamiento de un brazo humano con tan solo unos cuantos grados de libertad (GDL)

A lo anterior se llega utilizando el brazo robótico de InMoov que es controlado por medio de un dispositivo de luz estructurada (Kinect), el cual permitirá enviar la información adquirida del usuario a una computadora e interpretar los datos. Esta información es analizada para calcular los ángulos de las articulaciones del usuario e enviada a un modelo matemático como lo es la cinemática del brazo robótico que se encarga de posicionar el robot en el espacio. El movimiento del robot es controlado por una arduino que se encarga enviar las señales a los actuadores (servomotor) y poder posicionar el brazo según las indicaciones del usuario.

1.2. Descripción del Proyecto

1.2.1. Planteamiento del Problema y Justificación

En la actualidad se han desarrollado diferentes prototipos de brazos robóticos, en especial en la parte de investigación. En Colombia se ha avanzado en la investigación de brazos robóticos que asemejan el movimiento de un brazo humano con tan solo unos cuantos grados de libertad (GDL). Aunque se han desarrollado varias clases de prototipos de brazos humanos aún falta mucho para tener un prototipo que sea tan precisa como lo es un brazo humano. El desarrollo del robot InMoov es un avance tecnológico en la lucha por encontrar la perfección en la robótica que asemeje lo más posible al ser humano.

El programa de Ingeniería Mecatrónica de la Universidad de Pamplona cuenta con múltiples desarrollos de manos robóticas antropomórficas. Con el propósito de seguir mejorando se propone el diseño de un brazo robótico antropomórfico InMoov que imite un brazo humano, que puede ser utilizado como material didáctico en especial en las asignaturas de robótica, para implementarlos en prácticas como son de cinemática directa e inversa, control dinámico y teleoperación. Lo cual ayudaría ya que la universidad cuenta con muy pocos recursos para este tipo de prácticas.

A nivel industrial un brazo robótico que imite los movimientos de un brazo humano es de mucha utilidad ya que mitiga los daños a seres humanos en tareas peligrosas. Otro problema que posee la industria de la robótica son sus efectores finales que no dan un agarre como los que ofrece una mano humana.

1.3. Objetivos

Para desarrollo del brazo robótico de InMoov se tienen en cuenta los siguientes objetivos para su construcción.

1.3.1. Objetivo General

- Generar un algoritmo de teleoperación de un brazo robótico basado en un sistema de luz estructurada

1.3.2. Objetivos específicos

- Construcción del brazo robótico InMoov añadiendo mejoras a su diseño.
- Implementación de los algoritmos de control de bajo y alto nivel.
- Diseño de una interfaz hombre máquina para la teleoperación del brazo robótico-
- Realizar la puesta a punto y las pruebas de verificación

1.4. Alcances y limitaciones

El brazo robótico de InMoov cuenta con ciertas características que pueden afectar a su manipulación y se mostraran a continuación:

- El brazo robótico InMoov no tendrá las cualidades que tiene un brazo humano.
- El brazo robótico InMoov solo servirá para elementos livianos
- El brazo robótico InMoov al ser de uso público se encuentra con múltiples modificaciones a su diseño.
- El brazo robótico InMoov puede diferir en escala y proporciones a la de un brazo humano.
- El dispositivo de luz estructurada que se usará será el dispositivo Kinect de la empresa Microsoft.
- Se usaran librerías de otros autores para el control del dispositivo Kinect adquiridas en el semillero SIARC
- Se pueden implementar sistemas de teleoperación unidireccional o de lazo abierto
- El brazo robótico al poseer mecanismos de reducción sus movimientos no ocurren en tiempo real.

Capítulo 2.

2.1. Marco Teórico

A continuación se muestran algunos de los términos más fundamentales a tener en cuenta en esta tesis, como son las partes de un brazo humano y los sistemas que permiten el funcionamiento de un robot teleoperado por medio de un sistema de luz estructurada.

BRAZO: El brazo es el miembro superior más importante del cuerpo humano, comprende un sólo hueso llamado húmero que va unido al hombro en su parte superior hasta el codo en su parte inferior.

ANTEBRAZO: El antebrazo es la parte de la extremidad superior situada entre la articulación del codo y la de la muñeca. El antebrazo se compone sólo de dos huesos, el radio y el cúbito, que efectúan un movimiento de rotación uno alrededor del otro. (CCM, 2014)

MANO: La mano designa la parte terminal unida al antebrazo por la muñeca. Este órgano esencialmente sirve para coger objetos. . La mano humana está constituida por varios conjuntos óseos que forman las falanges, el metacarpo y el carpo. Los dedos de la mano están formados por tres falanges cada uno, a excepción del pulgar que cuenta con solo dos falanges. Estas falanges están unidas entre ellas por articulaciones que les permiten movilidad. La mano tiene una importancia particular ya que ofrece una multitud de posibilidades de acción. Desempeña un papel primordial, particularmente a nivel del sentido del tacto. (JEFF, 2014)

El conjunto de estas extremidades permiten realizar tareas con cierta complejidad, es una parte fundamental del ser humano y no poseerlas implican una complejidad superior e incluso en tareas muy cotidianas.

TELEOPERACIÓN: Un sistema teleoperado es aquél que permite gobernar un robot Esclavo (controlar su movimiento y la fuerza ejercida) ubicado en una zona remota (puede que el esclavo esté realmente muy lejos o puede que esté en un entorno hostil del que haya que proteger al operador) a través del manejo de un robot Maestro localizado en el punto de trabajo del operador. Así, el movimiento del esclavo debe ser predecible a partir de los movimientos que el operador provoca en el maestro, es decir, el operador debe comprender de manera muy intuitiva cómo mover el maestro para conducir al esclavo por una determinada trayectoria y además realizar las labores o trabajos requeridos. (Aliaga, 2000)

Kinect: El Kinect (originalmente conocido por el nombre en clave (Project Natal). Kinect es un dispositivo desarrollado por Microsoft para sus consolas de videojuegos Xbox 360, permitiendo al usuario jugar juegos utilizando su propio cuerpo como controlador de movimientos. (Alegsa, 2017)

Luz Estructurada: El principio que usa el sensor Kinect para su funcionamiento, consiste en proyectar un patrón de luz sobre una escena y observar la deformación del patrón en la superficie de los objetos. El patrón de luz puede ser proyectado, bien por un proyector LCD (luz no coherente) o bien por un barrido laser. Una cámara ligeramente desplazada respecto del proyector, captura la deformación de las líneas (o puntos) y calcula la distancia de cada punto utilizando una técnica similar a la triangulación. (Salvatore, Osio, & Morales, 2014)

ROBÓTICA: En lo que a Robótica de servicio se refiere, se encuentra la definición propuesta por el Comité ISO TC 184/SC2 que determina como “Todo tipo de robot que no es Industrial”. Asimismo, la IFR otorga la siguiente definición: “Robot que opera de forma parcial o totalmente autónoma al servicio del bienestar de los seres humanos y de equipamientos, excluyendo operaciones manufactureras”. (ISO & IFR)

2.2 ESTADO DEL ARTE

A la hora de Replicar un brazo robótico se presentan una serie de inconvenientes que afectan a su funcionamiento y es lo que hace que no se tenga un brazo robótico que cumpla todas las funciones que cumple un brazo humano.

Actualmente existen avances tecnológicos que han ayudado a la robótica, como lo es el desarrollo de brazos robóticos que son controlados por implantes cerebrales, *Figura 2.2. 1* hecho por los Científicos las universidades de Case Western y Chicago desarrollaron una prótesis que transmite la sensación de tacto para determinar la presión que se tiene que aplicar a un objeto. Los ingenieros y científicos recrearon con impulsos eléctricos de modo que el sistema nervioso interpreta la percepción táctil (EL NUEVO DIARIO, 2016),



Figura 2.2. 1 prótesis robótica con impulsos cerebrales
(EL NUEVO DIARIO, 2016)

Uno de los grandes avances tecnológicos en el área de prótesis de las extremidades superiores humanas es el brazo LUKE, *Figura 2.2. 2* este brazo fue diseñado por la Oficina de Tecnologías Biológicas de DARPA y el

Centro Medico Militar en Bethesda, esta prótesis fue diseñada para los soldados que han perdido estas extremidades en la guerra, estas extremidades están siendo fabricadas por Mobius Bionics LLC de Manchester una empresa creada para comercializar la tecnología desarrollada por DEKA Integrated Solutions Corp. Permite un movimiento extremadamente hábil del brazo y la mano con una retroalimentación de la fuerza de agarre a través de un control sencillo e intuitivo. Cuenta con una mano que tiene seis agarres seleccionables por el usuario y un brazo que permite el control simultáneo de múltiples articulaciones utilizando una variedad de entradas, incluyendo señales inalámbricas generadas por sensores innovadores usados en los pies del usuario. (DARPA, 2016),



*Figura 2.2. 2 Mobius Bionics LUKE Arms
(DARPA, 2016)*

La Universidad Johns Hopkins tiene uno de los laboratorios más importantes en el desarrollo de nuevas tecnologías, un brazo robótico de nueva generación conectado al tronco por 26 puntos y capaz de levantar hasta 20 kilogramos de peso *Figura 2.2. 3*. Para conseguirlo se unieron las terminaciones nerviosas de un paciente que perdió sus brazos a un nuevo elemento biónica, el cual recibe directamente las instrucciones para realizar los movimientos que el cerebro manda. (xataka, 2015),



*Figura 2.2. 3 Les Baugh controlando su brazo robótico modular
(xataka, 2015)*

La empresa EXIII Inc es una empresa japonesa que se especializa en la construcción de brazos bonicos, y ha desarrollado un brazo *Figura 2.2. 4* que detecta cuando los nervios y el tejido son estimulados por las señales del cerebro y envía esos datos al teléfono inteligente del usuario para interpretar esas señales y convertirlas en movimiento del brazo. El sistema permite a los

usuarios de las prótesis abrir y cerrar la mano e incluso controlar los dedos individuales. (WEVOLNER, 2016).



Figura 2.2. 4 brazo biónico que se controla con sensores en los músculos y teléfonos inteligentes (WEVOLNER, 2016)

Utilizando una interfaz cerebro – computadora o BCIs se crean unas extremidades robóticas capaz de ser controlada por el cerebro que tiene implantes que registran las señales de la corteza motora primaria, que está relacionada directamente con la medula espinal y los músculos. Los cirujanos implantan dos pequeñas matrices de micro-electrodos, cada una con 96 electrodos que pueden registrar la actividad eléctrica de las neuronas individuales. Las rejillas se unen a dos "pedestales" metálicos que sobresalen del cráneo. Los sujetos solo tienen que concentrarse en el objeto que deseaba agarrar, donde el sistema es capaz de identificar esa meta muy rápidamente. El BCI combina esa señal de alto nivel con funciones de robótica inteligente, como reconocimiento de objetos y localización y asignación simultáneas, o SLAM. (STRICKLAN, 2015), *Figura 2.2. 5.*



Figura 2.2. 5 brazo robótico controlado por medio de señales cerebrales (STRICKLAN, 2015)

La unión de OSRF y DARPA ha dado un paso agigantado al desarrollar una prótesis de nueva generación, con simulaciones en tiempo real HAPTIX ("Hand Proprioception and Touch Interfaces"), un programa que "busca crear un sistema de mano protésica que se mueva y proporcione sensación como una mano natural". HAPTIX busca crear una sensación tan parecida a un brazo humano que los usuarios no quieran quitarse la prótesis. El programa planea adaptar uno de los sistemas de extremidades protésicas desarrollados recientemente bajo el programa Revolutionizing Prosthetics de DARPA para incorporar interfaces que proporcionen un control intuitivo y una retroalimentación sensorial a los usuarios. Estas interfaces se basarían en tecnologías avanzadas de interfaz neuronal que se están desarrollando a través del programa de

tecnología de interfaz neuronal confiable (RE-NET) de DARPA.
(ACKERMAN, 2015)



*Figura 2.2. 6 prótesis revolucionaria HAPTIX
(ACKERMAN, 2015)*

La empresa de ingeniería de COAPT pretende crear el futuro del control de las prótesis, crear un brazo robótico mejorado con nuevo sistema de control que pueda reconocer las señales nerviosas del cerebro directamente, mediante la implantación de electrodos bajo la piel o incluso directamente en el cerebro.
Figura 2.2. 7



Figura 2.2. 7 brazo biónico

Dos estudiantes de ingeniería biomédica de la Universidad de Ryerson desarrollaron un brazo protésico controlado por señales cerebrales. El brazo artificial de músculo-operado (AMO) no sólo permite a los amputados más rango de movimiento en comparación con otros brazos de prótesis. (engadget, 2011), *Figura 2.2. 8*



*Figura 2.2. 8 brazo robótico AMO
(engadget, 2011)*

Robonaut es un robot humanoide diestro construido y diseñado en el Centro espacial Johnson en Houston, Texas. El reto era construir máquinas que puedan ayudar a los seres humanos en el trabajo y explorar en el espacio. Trabajando codo a codo con los humanos, o ir donde los riesgos son demasiado grandes para la gente, robonaut ampliará la capacidad para la construcción y el descubrimiento. Fundamental para que el esfuerzo sea por medio de la manipulación encarnada por la capacidad de utilizar la mano para hacer el trabajo, y el reto ha sido la construcción de máquinas con la destreza que supera a la de un astronauta adecuado. (NASA, 2016), *Figura 2.2. 9*



Figura 2.2. 9 robonaut
(NASA, 2016)

Capítulo 3

Implementación del Brazo Robótico IM

3.1 Descripción del Brazo InMoov

Es un robot humanoide, construido a partir de impresiones 3D, y controlado por un micro controlador Arduino. InMoov es un robot desarrollado con fines artísticos por el escultor francés Gaël Langevin, Su peculiaridad es que puede ser construida con cualquier impresora 3D. InMoov es capaz de percibir el sonido, ver, hablar y moverse de forma independiente. El robot es capaz de identificar su entorno a través de micro cámaras. Cuenta con diferentes sensores de contacto, además, el Kinect permite InMoov ver y analizar el espacio de 3 dimensiones del entorno del robot. (Langevin , 2012)



Figura 3.1. 1 robot InMoov

(Gael, 2017)

A la izquierda de la *Figura 3.1. 1 robot InMoov* Se ilustra parte del robot InMoov este robot cuenta con un total de 20 GDL en los brazos, 6 GDL en la cabeza y 2 GDL en el torso, posee 2 cámaras en los ojos con movimientos independientes al movimiento de la cabeza para el seguimiento de los rostros y por ultimo tiene 2 fuertes altavoces para hablar y escuchar con claridad. En la parte derecha se encuentra el último modelo del brazo izquierdo del robot InMoov posee un total de 10 GDL. 6 de esos movimientos son para los dedos y el movimiento de muñeca, los otros 4 GDL son para el movimiento del hombro y el movimiento del codo (Langevin , 2012)

3.2. Construcción

La construcción del brazo robótico InMoov está dividida en tres partes shoulder, bicep, forearm (hombro, bíceps, antebrazo), además uno de los

objetivos es la construcción del brazo robótico el cual debe tener un ensamble preciso para su buen funcionamiento, además se requerirá un total de 5 grados de libertad para cumplir este objetivo.

3.2.1 Ensamble del antebrazo

El proyecto de antebrazo robótico utiliza 13 archivos *.stl*, del proyecto de robótica de código abierto InMoov, Para obtener el mejor resultado de impresión se siguieron las especificaciones de la página oficial donde se sugiere imprimir con un relleno de más de un 30% en ABS y los engranes a máxima capacidad (100% de relleno). Para este trabajo se utilizó material PLA con un relleno de un 40% y un 100% de relleno en los engranes.

1x robpart3
1x robpart4
1x robpart5
1x rotawrist2
1x rotawrist1
1x rotawrist3

1x WristGears
1x CableHolderWrist
1x Robservobed
1x Robcablefrontv1
1x RobcableBack
1xRobcab
1x robpart2

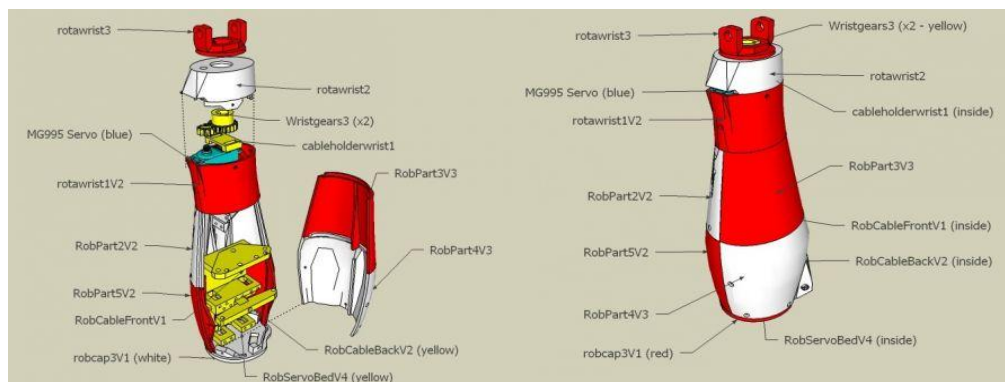


Figura 3.2.1, 1 Jerarquía de las piezas del brazo de InMoov

Teniendo el modelo jerárquico mostrado en la *Figura 3.2.1, 1*. Ensamblando las piezas robpart 2 y 5, junto con la pieza rotawrist1 en su parte superior y Robcab en la parte inferior como se muestra en la *Figura 3.2.1, 2* de la vista explosionada. Da como resultado el primer sub-ensamble de brazo, como se ilustran en la *Figura 3.2.1, 3*

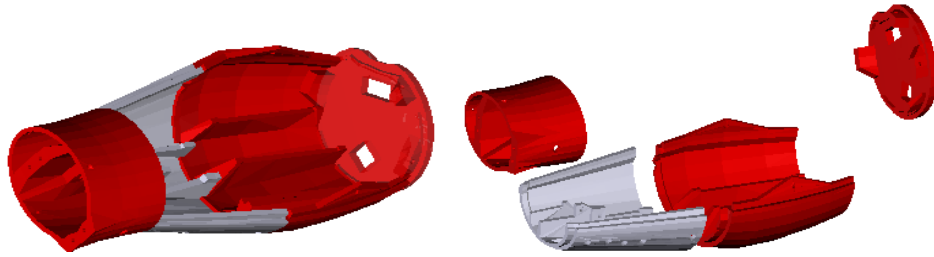


Figura 3.2.1. 2 Vista 3D del primer sub - ensamble

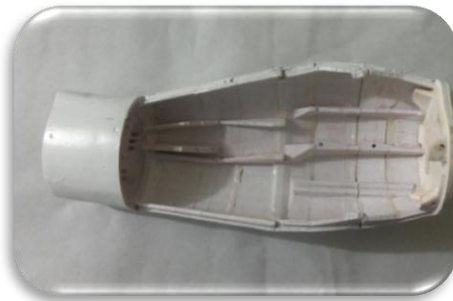


Figura 3.2.1. 3 primer sub-ensamble del antebrazo

Siguiendo con el ensamblado del antebrazo con sus últimas piezas las correspondientes a **robpart 3** y **4**, se obtiene un sub-ensamble que servirá como tapa superior del ensamble. De la *Figura 3.2.1. 4* da como resultado el sub ensamble mostrado en la *Figura 3.2.1. 5*

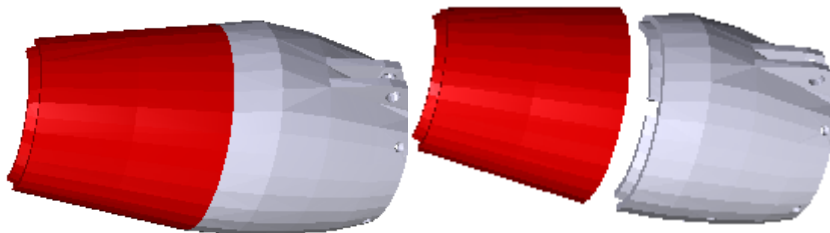


Figura 3.2.1. 4 vista 3D sub ensamble 2



Figura 3.2.1. 5 sub - ensamble 2 tapa superior

El siguiente paso es ensamblar las piezas correspondientes a la rotación de la muñeca (**rotawrist 3 y 2, WristGears, CableHolderWrist y Servomotor**) como se puede ver en la *Figura 3.2.1. 6*. Este ensamble permitirá hacer el movimiento de pronación – supinación, y debe llevar suficiente grasa para su buen funcionamiento como se puede observar en la *Figura 3.2.1. 7*

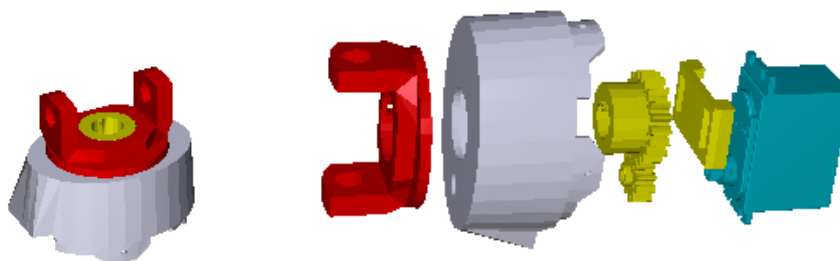


Figura 3.2.1. 6 f vista 3d sub ensamble pronación- supinación de la muñeca



Figura 3.2.1. 7 sub - ensamble pronación- supinación n de la muñeca

El ensamble de la pieza **Robservobed** *Figura 3.2.1. 8* sostienen los servos que permiten el movimiento de los dedos de la mano. Como se muestra en la *Figura 3.2. 1.9*

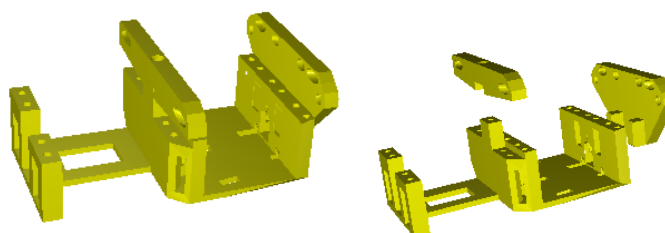


Figura 3.2.1. 8 vista 3D de Robservobed

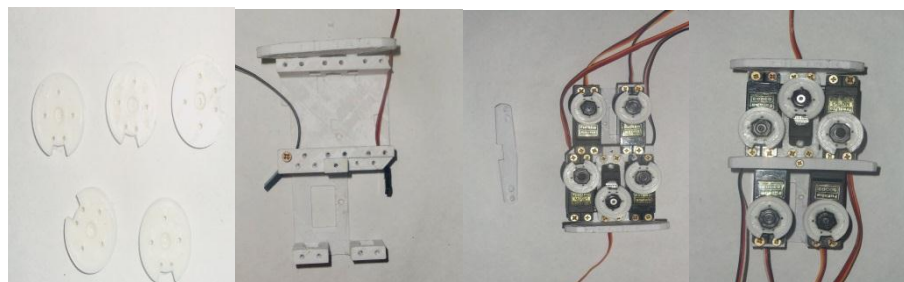


Figura 3.2. 1.9 ensamble de Robservobed

Por último se usan los sub-ensambles realizados anteriormente, y así terminar la parte correspondiente al ensamble del antebrazo de InMoov.

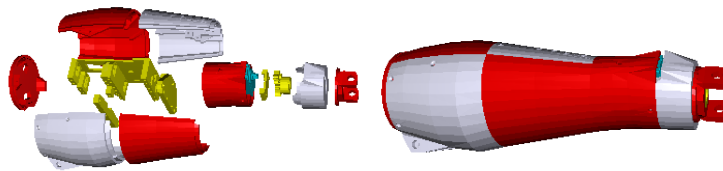


Figura 3.2.1. 10 vista 3D sub-ensamble antebrazo de InMoov



Figura 3.2.1. 11 sub-ensamble antebrazo InMoov

3.2.2. Ensamble del bíceps

El proyecto de bíceps utiliza 20 archivos *stl*, del proyecto de robótica de código abierto InMoov, Para obtener el mejor resultado de impresión, se siguieron las especificaciones de la página oficial de **InMoov**, como se indicó en el **3.2.1 Ensamble del antebrazo**

1x GearHolder	1x armtopcover1
2x HighArmSide	1x armtopcover2
1x Pistonclock	1x armtopcover3
1x Pistonbaseanti	1x elbowshaftgear
1x RotGear	1x gearpotentio
1x RotMit	2x lowarmside
1x RotPotentio	2x reinforcer
1x RotTit	1x servobase
1x RotWorm	1x servoholder
1x Rotcenter	1x spacer

El ensamble del bíceps es el más sencillo de todo el brazo, siguiendo los pasos que se muestran en la *Figura 3.2.2. 1* y en la *Figura 3.2.2. 2* la vista explosionada, para el ensamble de las piezas se recomienda poner un poco de pegamento en las uniones para evitar que las piezas se salgan con el movimiento. *Figura 3.2.2. 3*

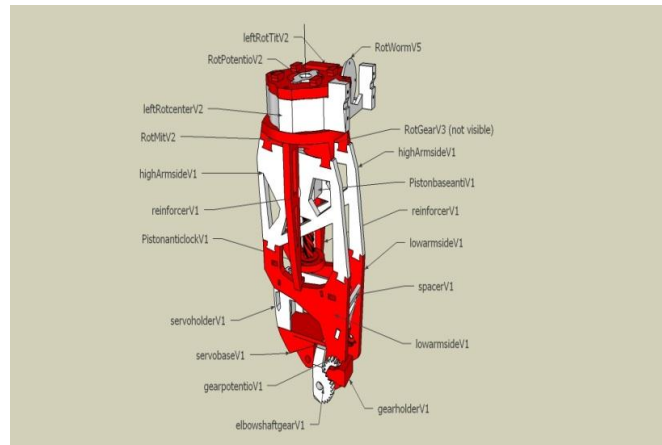


Figura 3.2.2. 1 modelo 3D del bíceps de InMoov

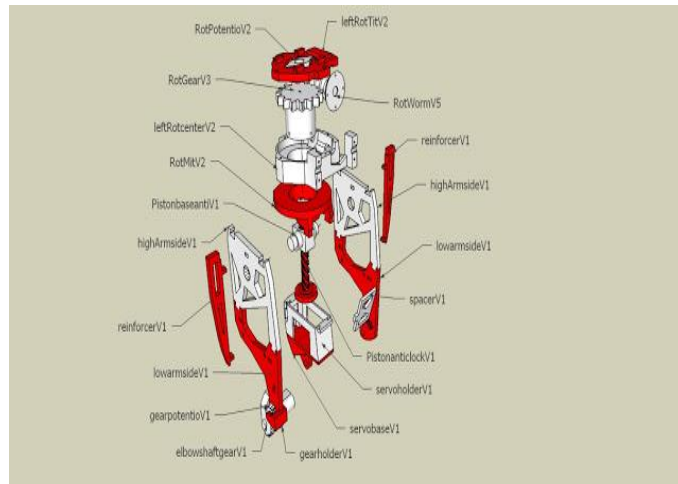


Figura 3.2.2. 2 Jerarquía de las piezas del bíceps de InMoov

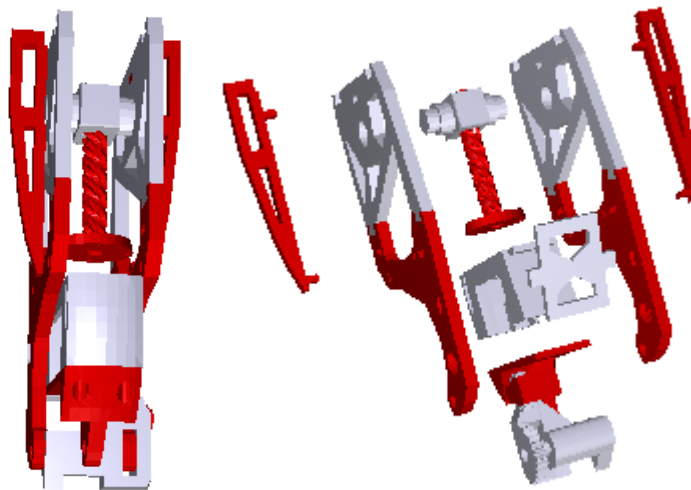


Figura 3.2.2. 3 vista 3D Bíceps

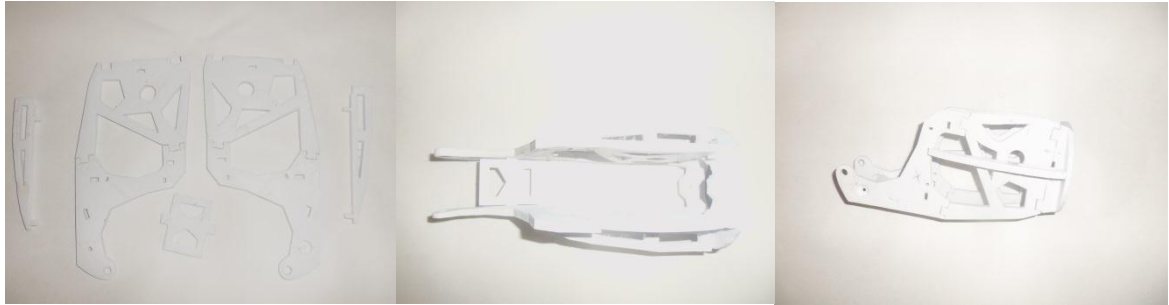


Figura 3.2.2. 4 Ensamble de las piezas correspondientes al bíceps

Ya ensambladas las piezas del bíceps se procede a unir con el ensamble del antebrazo, como se puede visualizar en la *Figura 3.2.2. 5*



Figura 3.2.2. 5 unión entre el ante brazo y el bíceps

Antes de colocar el servo que dará movimiento de flexión – extensión en el codo se tiene que modificar para que pueda cumplir con su trabajo a la perfección. Para poder llevar el antebrazo a una posición entre 0° a 90° o viceversa con respecto al bíceps, el motor debe girar alrededor de 450° es decir una vuelta y un cuarto ($1\frac{1}{4}$), algo que resulta imposible teniendo en cuenta que la mayoría de los servos solo giran 180° es decir media vuelta ($\frac{1}{2}$), en este caso el servo **HS-805BB** que se utiliza para este movimiento.

El primer paso es desmontar el circuito del servo y así extraer el potenciómetro que permite al servo posicionarse en un ángulo específico. Existen dos formas para lograr extraer el potenciómetro.

- El primer método consiste en presionar el engrane metálico del motor con un destornillador y así poder extraer el circuito con mayor facilidad. como se muestra en la *Figura 3.2.2. 6*



Figura 3.2.2. 6 extracción del circuito

- El segundo método se usa cuando el pegamento que tiene el motor impide sacarlo por el primer método, y consiste en desoldar los tres pines que sujetan al motor, y poder sacar el circuito sin necesidad de quitar el motor *Figura 3.2.2. 7*

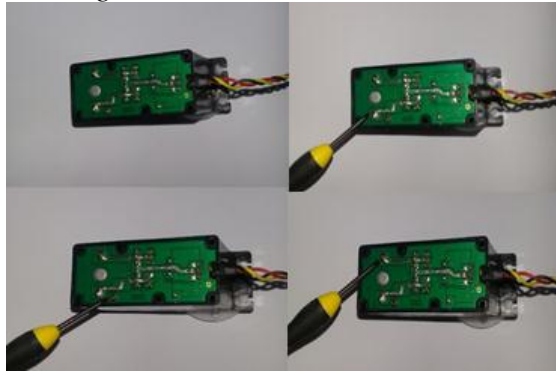


Figura 3.2.2. 7 segunda forma de la extracción del circuito

Una vez se tenga el circuito y el potenciómetro con él, es necesario cortar los cables del potenciómetro sin perder la posición en la que van cada uno, y adicionar un nuevo cable que sea de un largo superior o igual a 18cm y soldar nuevamente el potenciómetro como se encontraba originalmente. *Figura 3.2.2. 8*



Figura 3.2.2. 8 partes internas del servo HS-805BB

Por último se debe truncan el servo, para que el motor tenga un giro continuo como se muestra en la *Figura 3.2.2. 9*

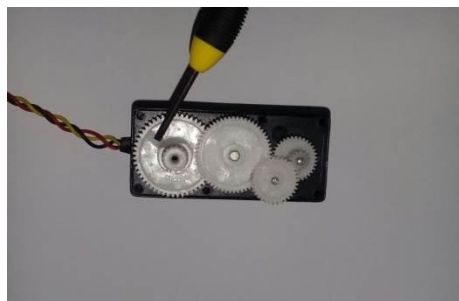


Figura 3.2.2. 9 truncando uno de los engranes

Este procedimiento se debe realizar a los 4 servomotores HS-805BB, ya que son los encargados de darle movimiento al brazo.

Ya modificado el servo se procede a unir en las pieza Servohoder y Pistonclock como se muestra en la *Figura 3.2.2. 10*, al conjunto del ensamble dl brazo

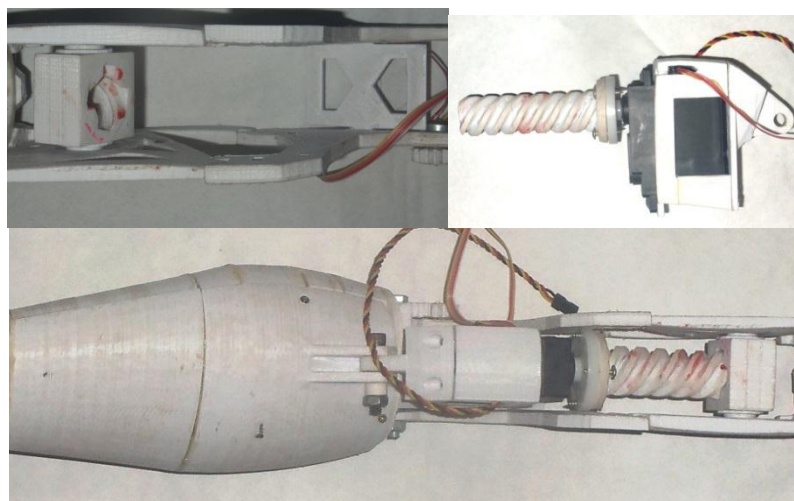


Figura 3.2.2. 10 Ensamble del biceps

Con el potenciómetro que se extrae del servo recomienda colocarlo en la posición 0° antes de montar el engrane y así poder evitar cualquier ruptura de pieza, al momento de probar el servo se recomienda mandar pequeños angulas y mirar cual es la posición inicial y la final, dependiendo de qué tanta precisión se tenga al colocar el potenciómetro. En este caso los ángulos que puede moverse este servo son de 10° a 90°, pero por seguridad los movimientos se efectúan desde 15° a 85°. *Figura 3.2.2. 11*. En caso de que el servo gire en dirección contraria intercambiar la posición de los cables tanto positivo como negativo sin mover el cable de señal.

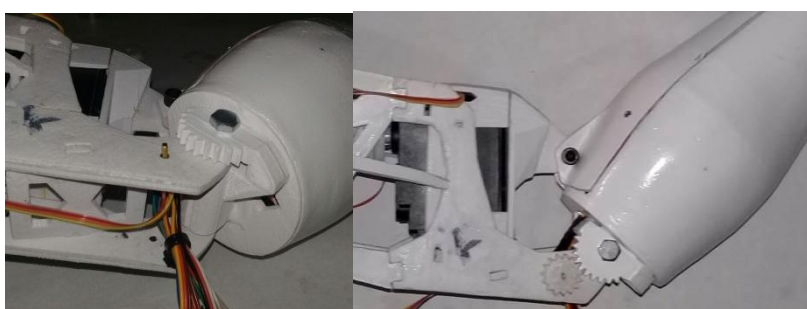


Figura 3.2.2. 11 Calibración del potenciómetro del codo

Para terminar el ensamble del bíceps hace falta ensamblar las piezas que permitirán al brazo hacer el movimiento de torsión. Rotpotentio v2, RotTitv2, RotWormv5, RotGearv3, RotCenter. Como los mostrados en a *Figura 3.2.2. 12*.

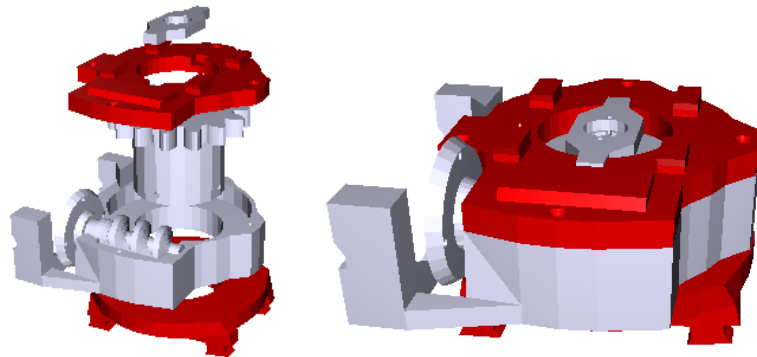


Figura 3.2.2. 12 vista 3D del sub ensamble de torsión

Siguiendo los pasos que se muestran en la *Figura 3.2.2. 13*, se procede a hacer el ensamble de torsión. Algo a tener en cuenta a la hora de hacer este ensamble es agregar una buena cantidad de grasa que permita a las piezas girar con mayor soltura.



Figura 3.2.2. 13 ensamble piezas de torsión

Por último se coloca el potenciómetro del servo con un ángulo de 0° para permitirle al brazo robótico tener un movimiento lo más parecido al de un ser humano *Figura 3.2.2. 14*

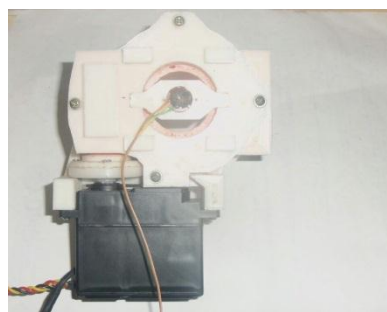


Figura 3.2.2. 14 Calibración del ensamble de rotación a 90°

3.2.3 Ensamble del hombro

El proyecto del hombro robótico utiliza 14 archivos *stl*, del proyecto de robótica de código abierto **InMoov**, para su construcción se utiliza las mismas especificaciones ya dicha en el *3.2.1 Ensamble del antebrazo*

- | | |
|------------------------|------------------------|
| 1x <u>ClaviBack</u> | 1x <u>PivPotentio</u> |
| 1x <u>ClaviFront</u> | 1x <u>PivPotholder</u> |
| 1x <u>PistonClavi</u> | 1x <u>PivTit</u> |
| 1x <u>Pistonbase</u> | 1x <u>PivWorm</u> |
| 2x <u>PivConnector</u> | 1x <u>Pivcenter</u> |
| 1x <u>PivGear</u> | 1x <u>servoHolster</u> |
| 1x <u>PivMit</u> | 1x <u>servoholder</u> |

Al igual que lo hecho en el ensamble del bíceps *Figura 3.2.2. 13*, el procedimiento para el ensamble del movimiento de flexión – extensión en el hombro es igual, lo que cambia es el nombre de las piezas. *Figura 3.2.3. 1 Jerarquía de las piezas del hombro de InMoov*



Figura 3.2.3. 1 Jerarquía de las piezas del hombro de InMoov

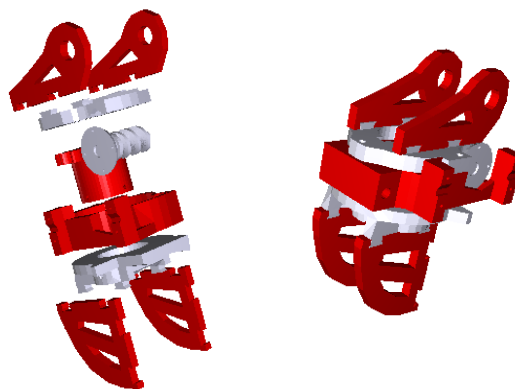


Figura 3.2.3. 2 vista 3D movimiento de fl-ex

Se agregan las piezas Pivconnectorv1 que conectara el ensamble de torsión con el de flexión - extensión. Por último se agregan las piezas de Servoholderv1 que sujetara todo el brazo a su base. *Figura 3.2.3. 3*



Figura 3.2.3. 3 montaje de los ensambles de rotación y fl-ex

De esta forma ya se ha terminado el ensamble de todas las piezas correspondiente al brazo de InMoov *Figura 3.2.3. 4*

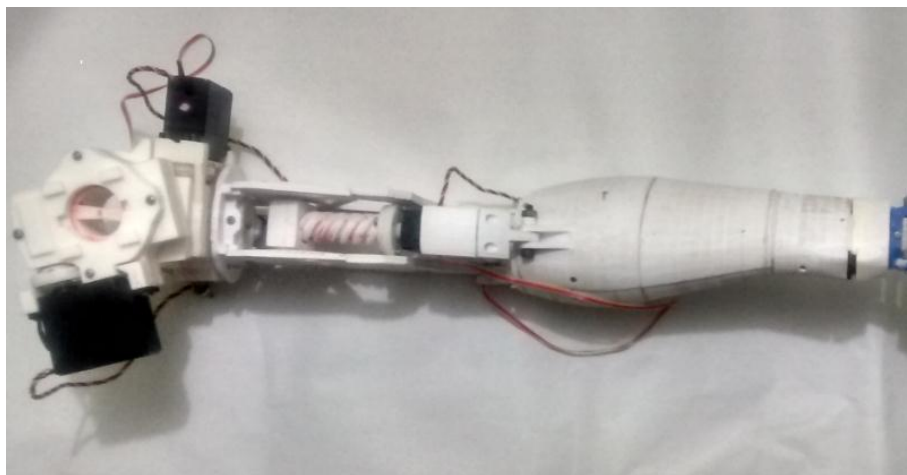


Figura 3.2.3. 4 vista terminada del brazo de InMoov

3.3. Modificaciones

Desde la construcción del robot InMoov en el año 2012 se han desarrollado múltiples modificaciones tanto por el mismo creador Gael Langevin, como también por los miles de usuarios alrededor del mundo, ya que este robot es de código abierto y las personas pueden aportar nuevos diseños con el fin de seguir mejorando, lo que ha llevado una evolución en muy poco tiempo.

Una de las modificaciones más importante en el brazo robótico es el del movimiento de pronación - supinación correspondiente al movimiento de la muñeca, en la primera versión del brazo robótico carecía de ese movimiento, como se puede ver en la *Figura 3.3. 1 versión del brazo de InMoov sin el movimiento de pron - sup* y su última modificación hecha en el año 2016 como se muestra en la *Figura 3.3. 2 versión del brazo de InMoov con el movimiento de pron - sup*

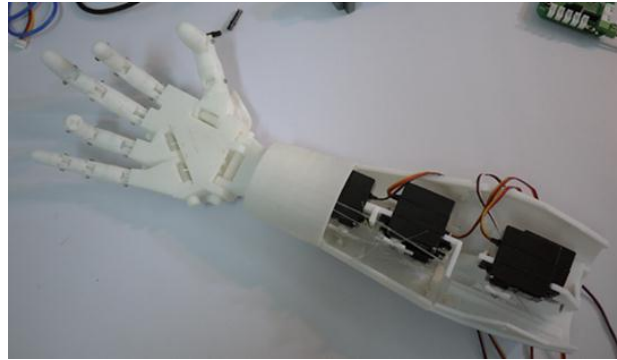


Figura 3.3. 1 versión del brazo de InMoov sin el movimiento de pron - sup

(han, 2017)

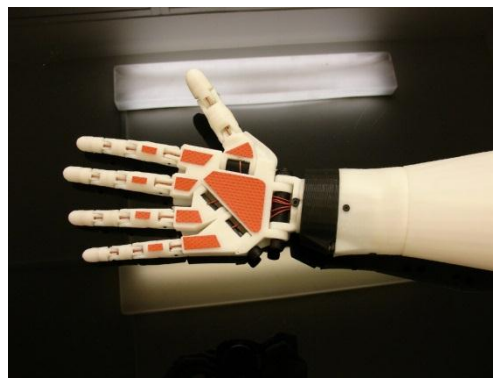


Figura 3.3. 2 versión del brazo de InMoov con el movimiento de pron - sup

(Gael, 2017)

Otras de las modificaciones importantes es en el hombro, debido a que para construir el brazo completo es necesario construir el torso que es el encargado de sostener todo el peso de los brazos como se ve en la *Figura 3.3. 3 muestra del brazo InMoov con torso incluido*, lo que muchos usuarios de InMoov no ven viable ya sea por costo o por tiempo y eliminan el ultimo grado de libertad correspondiente al movimiento de aducción - abducción como se muestra en la *Figura 3.3. 4 modificación hechas por usuario de InMoov*. La página oficial de InMoov también ofrece una opción para los que no desean construir el torso, y consiste en construir dos veces la sección del bíceps con una modificación adicional como se ilustra en la *Figura 3.3. 5*, esta modificación no elimina el movimiento de aducción - abducción pero si mitiga su movimiento sustancialmente.

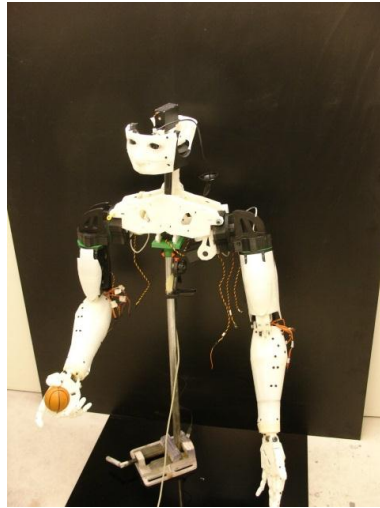


Figura 3.3. 3 muestra del brazo InMoov con torso incluido (Gael, 2017)



Figura 3.3. 4 modificación hechas por usuario de InMoov

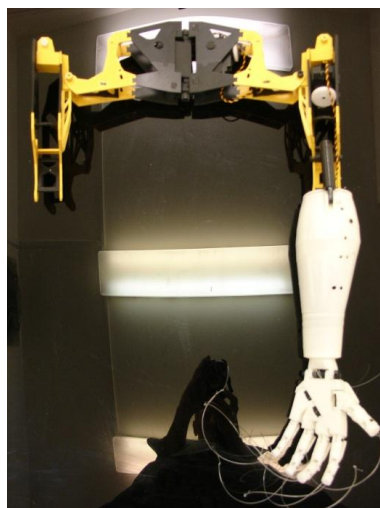


Figura 3.3. 5 modificación hecha por la página de InMoov sin torso (Gael, 2017)

Uno de los objetivos de este trabajo es el de añadir modificaciones a su diseño, la modificación hecha en este trabajo reemplaza la función del torso en solo dos piezas *Figura 3.3. 6* esta pieza soporta todo el brazo con gran firmeza debido a su diseño que permite que no se doble con el esfuerzo resultante del brazo, de esta forma se puede utilizar el ultimo grado de libertad que se elimina en las modificaciones mostradas anteriormente.

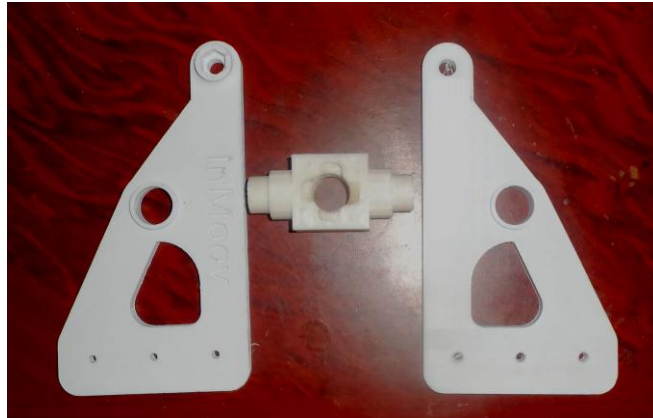


Figura 3.3. 6 soporte base del brazo

Las piezas van unidas a la base del tornillo sin fin que permite hacer el movimiento de aducción – abducción. *Figura 3.3. 7*



Figura 3.3. 7 montaje del tornillo sin fin con la base del brazo

Para culminar el ensamble del brazo de InMoov se unen los ensambles hechos en la *Figura 3.3. 7* con el ensamble hecho en la *Figura 3.2.3. 4*, dando como resultado el brazo terminado mostrado en la *Figura 3.3. 8*

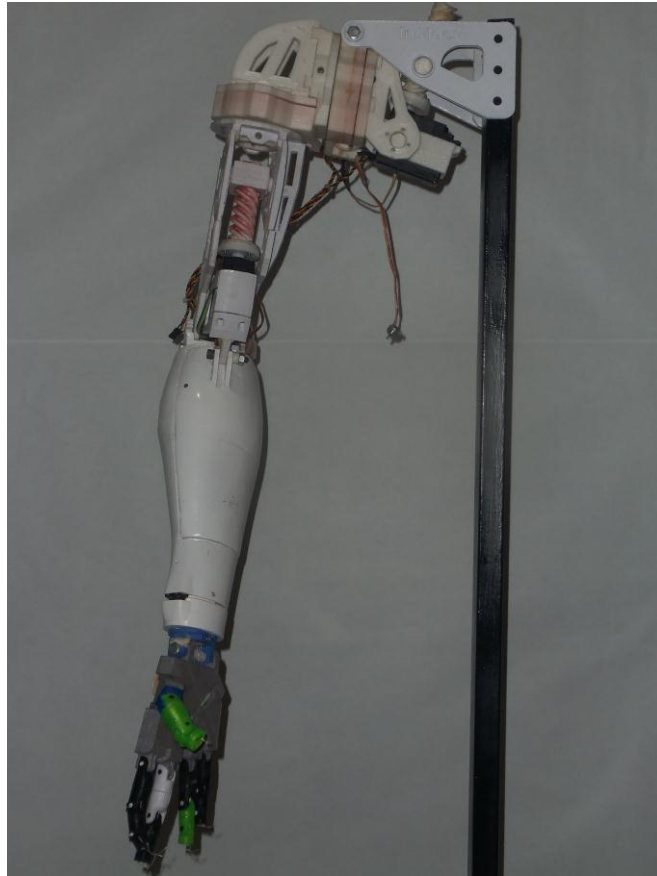


Figura 3.3. 8 brazo de inmoov completo

3.4. Implementación del sistema de control

El brazo robótico contienen componentes electrónicos que brindan entre si la locomoción que se necesita para la manipulación del brazo robótico de InMoov, a continuación se nombran los componentes que hacen posible este trabajo.

3.4.1. Actuadores

El movimiento de pronación - supinación en la muñeca y el movimiento de los dedos se realiza con el servomotor **MG995** *Figura 3.4.1 1* con piñonería Metálica que tiene un voltaje nominal de (4.8 a 6) V, un par de (9.4 a 11) Kg.cm y una corriente de (80 a 10k0) mA, con una velocidad de operación de (0.2 a 0.16) sec/60 (towerpro, 2016)



Figura 3.4.1 1 servomotor mg995

(towerpro, 2016)

HS-805BB Mega Figura 3.4.1 2 llamado ‘monster’, es uno de los más potentes de su clase, de piñonería plástica, tienen un par de (19.8 a 24,7) kg.cm con una operación de voltaje que va de (4.8 a 6) V, con una velocidad de operación de (0.19 a 0.14) sec/60 y una corriente de 700mA sin carga y 1200 mA con carga. Uno de los puntos a tener en cuenta con estos tipos de motores es el alto consumo de corriente que se necesita para el buen funcionamiento de cada uno. (dynamo, 2017)



Figura 3.4.1 2 servomotor hs805bb

(dynamo, 2017)

3.4.2. Sistema de control

El Arduino Nano es una placa pequeña, completa y amigable con un micro basado en el ATmega328, este es el controlador del brazo robótico, se encarga de recibir la información de los ángulos que interpreta el dispositivo de luz estructurada Kinect junto con MATLAB, y son enviadas las señales correspondientes al circuito de potencia. El número de pines utilizables en esta tarjeta, el procesamiento y su reducido tamaño la hace idónea a la hora de seleccionar este dispositivo por que como se puede ver en las *Conexiones Eléctricas Del Brazo Robótico* del brazo se utilizan la mayoría de los pines digitales. *Figura 3.4.2. 1*

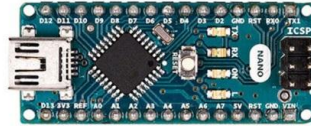


Figura 3.4.2. 1 Arduino nano v3

. (Arduino, 2017)

- | | |
|--------------------------------------|---|
| • Microcontrolador | ATmega328 |
| • Arquitectura | AVR |
| • Tensión de funcionamiento | 5 V |
| • Memoria flash | 32 KB de los cuales 2 KB utilizados por el gestor de arranque |
| • SRAM | 2 KB |
| • Velocidad de reloj | 16 MHz |
| • Pines de E / S analógicos | 8 |
| • EEPROM | 1 KB |
| • Corriente de CC por pines de E / S | 40 mA (E / S) |
| • Voltaje de entrada | 7-12 V |
| • Pines Digitales I / O | 14 (de las cuales 6 proporcionan salida PWM) |
| • El consumo de energía | 19 mA |
| • Tamaño de PCB | 18 x 45 mm |
| • Peso | 7 g |
| • Código de producto | A000005 |

Esta placa de arduino puede ser alimentado desde su puerto USB o una fuente externa desde 7 V hasta 11.8 V esto permite alimentar tanto a la placa de arduino como al circuito de potencia sin necesidad de tener una fuente para cada una.

Al final se encuentran los códigos para la teleoperación del brazo.

3.4.3 Interfaz grafica

El prototipo de brazo robótico de InMoov es controlado por medio del dispositivo de luz estructura, que interpreta los puntos de cada articulación, esta información es enviada a la computadora que se encarga de analizar los puntos que interpreta Kinect y calcular lo ángulos en los que se encuentra cada articulación del brazo.

El prototipo ha sido diseñado junto con una interfaz gráfica que permite visualizar el movimiento del brazo robótico dibujado en hilos y los ángulos en los cuales se encuentra cada articulación, la interfaz gráfica también posee un modo manual que permite al usuario posicionar el brazo en los ángulos deseados *Figura 3.4.3. 1*

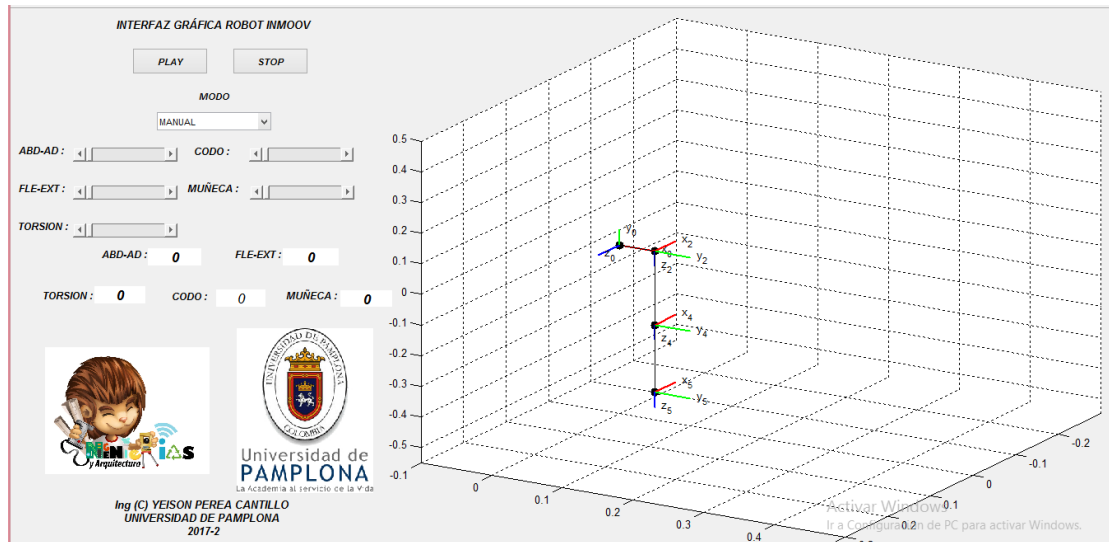


Figura 3.4.3. 1 interfaz gráfica en modo MANUAL

Capítulo 4

Algoritmo Para La Teleoperación

En este capítulo se abarcará todo lo correspondiente a la manipulación del brazo robótico de InMoov, desde el análisis cinemático hasta las capturas del movimiento del Kinect

4.1 Modelo Cinemático

El prototipo de brazo robótico que se propone en este trabajo es el brazo de InMoov que posee un total de 5GDL comprendido desde el hombro hasta la muñeca. Y un total de 5GDL en la mano que será el efector final del brazo robótico

4.1.1 modelo cinemático directo (MCD)

Un brazo humano se encuentra normalmente en posición de aducción, *Figura 4.1.1. 1* para mayor comodidad a la hora de desarrollar el modelo cinemático del brazo en especial el *4.1.2 Modelo Cinemático Inverso (MCI)*, se dispone de una posición que se encuentra girada 90° en el eje **Y** de Matlab. Como se puede ver en la *Figura 4.1.1. 2*

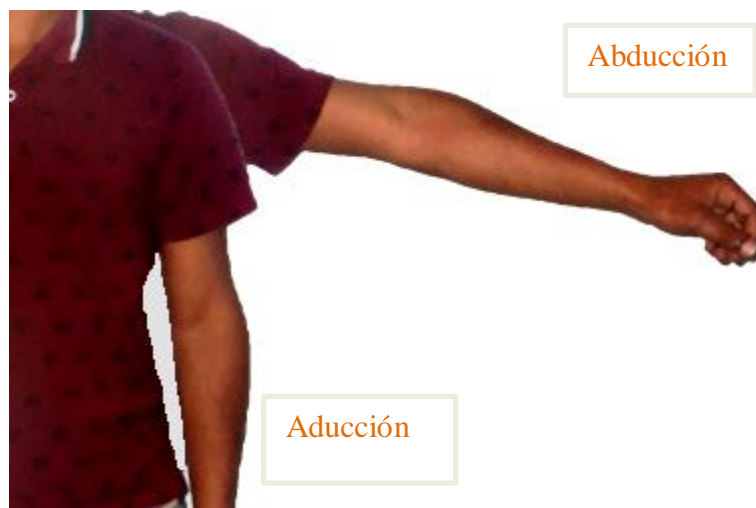


Figura 4.1.1. 1 posición de un brazo humano Aducción - Abducción

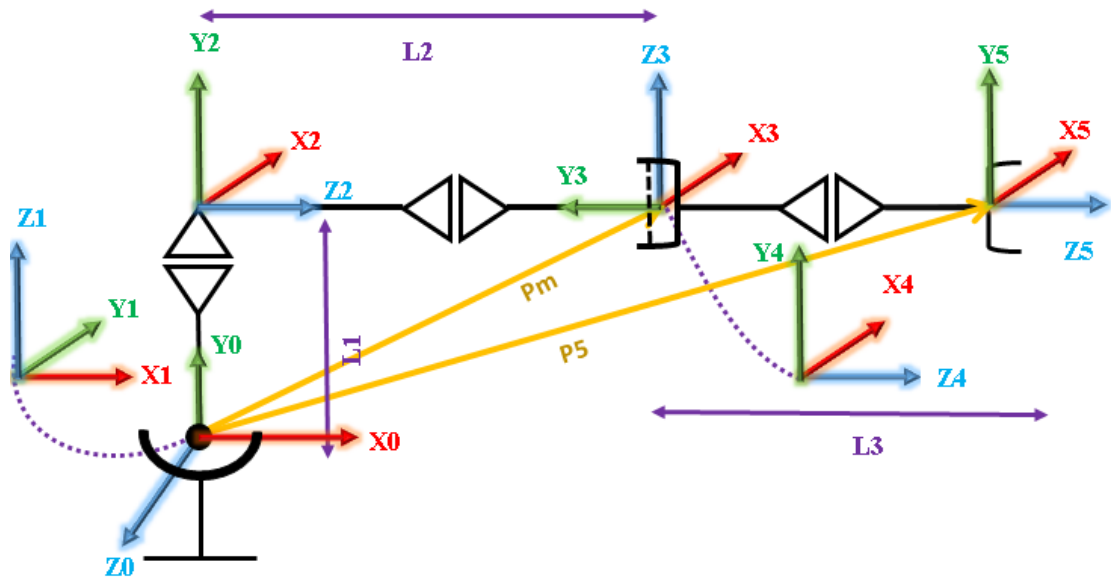


Figura 4.1.1. 2 Grafica del MCD brazo robótico InMoov

Utilizando los pasos de Denavit – Hartenberg se obtienen los siguientes parámetros de DH: q , d , a , α

Ecuación 4.1. 1 Parámetros de Denavit-Hartenberg (DH)

Nº	q	d	a	α
0-1	$Q1$	0	0	-90
1-2	$Q2+90$	$L1$	0	90
2-3	$Q3$	$L2$	0	-90
3-4	$Q4$	0	0	90
4-5	$Q5$	$L3$	0	0

Utilizando la MTH de DH: ${}^{i-1}A_i$

Ecuación 4.1. 2 MTH – DH ${}^{i-1}A_i$

$${}^{i-1}A_i = \begin{bmatrix} Cq_i & -C\alpha_i S q_i & -S\alpha_i S q_i & a_i C q_i \\ S q_i & -C\alpha_i C q_i & -S\alpha_i C q_i & a_i S q_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Del primer renglón del parámetro de DH Ecuación 4.1. 1, se obtiene al remplazar en la Ecuación 4.1. 2

Ecuación 4.1. 3 MTH 0A_1

$${}^0A_1 = \begin{bmatrix} Cq_1 & 0 & -Sq_1 & 0 \\ Sq_1 & 0 & Cq_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Del segundo renglón del parámetro de DH *Ecuación 4.1. 1*, se obtiene al remplazar en la *Ecuación 4.1. 2*

Ecuación 4.1. 4 MTH 1A_2

$${}^1A_2 = \begin{bmatrix} C(q_2 + 90) & 0 & S(q_2 + 90) & 0 \\ S(q_2 + 90) & 0 & C(q_2 + 90) & 0 \\ 0 & 1 & 0 & L1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Del tercer renglón del parámetro de DH *Ecuación 4.1. 1*, se obtiene al remplazar en la *Ecuación 4.1. 2*

Ecuación 4.1. 5 MTH 2A_3

$${}^2A_3 = \begin{bmatrix} Cq_3 & 0 & -Sq_3 & 0 \\ Sq_3 & 0 & Cq_3 & 0 \\ 0 & -1 & 0 & L2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Del cuarto renglón del parámetro de DH *Ecuación 4.1. 1*, se obtiene al remplazar en la *Ecuación 4.1. 2*

Ecuación 4.1. 6 MTH 3A_4

$${}^3A_4 = \begin{bmatrix} Cq_4 & 0 & Sq_4 & 0 \\ Sq_4 & 0 & -Cq_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Del quinto renglón del parámetro de DH *Ecuación 4.1. 1*, se obtiene al remplazar en la *Ecuación 4.1. 2*.

Ecuación 4.1. 7 MTH 4A_5

$${}^4A_5 = \begin{bmatrix} Cq_5 & -Sq_5 & 0 & 0 \\ Sq_5 & Cq_5 & 0 & 0 \\ 0 & 0 & 1 & L3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Obteniendo así una matriz que indica la localización del efector final del robot.

$$T = {}^0A_5 = {}^0A_1 + {}^1A_2 + {}^2A_3 + {}^3A_4 + {}^4A_5$$

Donde cada término de la matriz 0A_5 tiene los siguientes vectores x, y, z y el vector de posición (p) del efector final:

Ecuación 4.1. 8 MTH 0A_5 simplificada

$${}^0A_5 = \begin{bmatrix} x & y & z & p \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ecuación 4.1. 9 MTH 0A_5

$${}^0A_5 = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Las variables S y C mostradas a continuación son una abreviación de los términos Seno y Coseno, al igual con las siglas q1, q2, q3, q4, q5, son abreviaturas correspondientes a los ángulos de las articulaciones del brazo correspondiente desde el 1 hasta el 5

Vector x

$$a_{11} = -C(q5) * (C(q4) * (S(q) * S(q3) + C(q) * C(q3) * S(q2))) + C(q) * C(q2) * S(q4)) - S(q5) * (C(q3) * S(q) - C(q) * S(q2) * S(q3))$$

$$a_{21} = C(q5) * (C(q4) * (C(q) * S(q3) - C(q3) * S(q) * S(q2))) - C(q2) * S(q) * S(q4)) + S(q5) * (C(q) * C(q3) + S(q) * S(q2) * S(q3))$$

$$a_{31} = C(q5) * (S(q2) * S(q4) - C(q2) * C(q3) * C(q4)) + C(q2) * S(q3) * S(q5))$$

Vector y

$$a_{12} = S(q5) * (C(q4) * (S(q) * S(q3) + C(q) * C(q3) * S(q2))) + C(q) * C(q2) * S(q4)) - C(q5) * (C(q3) * S(q) - C(q) * S(q2) * S(q3))$$

$$a_{22} = C(q5) * (C(q) * C(q3) + S(q) * S(q2) * S(q3)) - S(q5) * (C(q4) * (C(q) * S(q3)) - C(q3) * S(q) * S(q2)) - C(q2) * S(q) * S(q4))$$

$$a_{32} = C(q2) * C(q5) * S(q3) - S(q5) * (S(q2) * S(q4) - C(q2) * C(q3) * C(q4))$$

Vector z

$$a_{13} = C(q) * C(q2) * C(q4) - S(q4) * (S(q) * S(q3) + C(q) * C(q3) * S(q2))$$

$$a_{23} = S(q4) * (C(q) * S(q3) - C(q3) * S(q) * S(q2)) + C(q2) * C(q4) * S(q)$$

$$a_{33} = -C(q4) * S(q2) - C(q2) * C(q3) * S(q4)$$

Vector posición

$$a_{14} = L2 * C(q) * C(q2) - L3 * (S(q4) * (S(q) * S(q3) + C(q) * C(q3) * S(q2)) - C(q) * C(q2) * C(q4)) - L1 * S(q))$$

$$a_{24} = L1 * C(q) + L3 * (S(q4) * (C(q) * S(q3) - C(q3) * S(q) * S(q2)) + C(q2) * C(q4) * S(q)) + L2 * C(q2) * S(q))$$

$$a_{34} = -L3 * (C(q4) * S(q2) + C(q2) * C(q3) * S(q4)) - L2 * S(q2)$$

4.1.2 Modelo Cinemático Inverso (MCI)

Para el desarrollo del MCI se parte de la localización en que se encuentra el efector final del brazo robótico de InMoov es decir $T = {}^0A_5$, utilizando el método de desacople cinemático se comienza a calcular los tres primeros grados de libertad del brazo robótico a partir del punto de la muñeca. Ya obtenidas los tres primeros grados de libertad (q_1, q_2, q_3) se procede a calcular los últimos grados de libertad.

Para el desarrollo del MCI, se retoma la posición original del brazo, es decir una rotación de -90° en el eje X con respecto al eje de Matlab, en el sistema S_0 *Figura 4.1.2. 1 posición de aducción del brazo InMoov.*

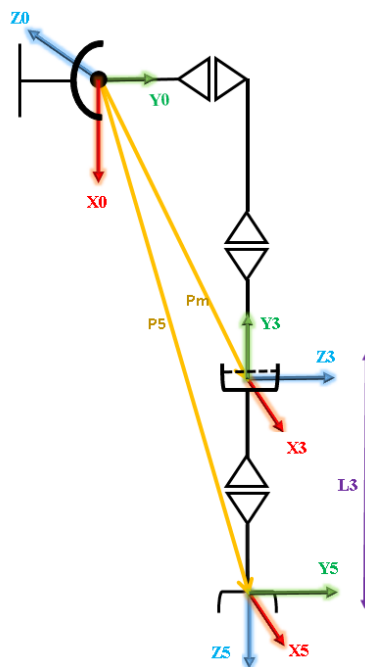


Figura 4.1.2. 1 posición de aducción del brazo InMoov

Una vez que se tiene la matriz ($T = A_5^0$) obtenida en el 4.1.1 *modelo cinemático directo (MCD)* se procede a calcular la cinemática inversa, se calculan los tres primero GDL a partir del punto de la muñeca (Pm). Y teniendo en cuenta que la dirección del sistema 5 y 6 (S_5 y S_6) siempre van a coincidir. Se puede decir que:

$$Z4 = Z5$$

$$Pm = PP + (-L5 * P)$$

Hay que tener en cuenta que el robot tiene una rotación en X de 90°, por lo tanto se debe regresar a la posición original para poder calcular la matriz A_1^0 .

$$p = \text{rotx}(-90) * A_0^5$$

$$pp = P(1:3,4) = Z5$$

El robot en su posición original se observa en longitud verdadera las longitudes L1, L2, L3, no obstante al hacer girar las articulaciones q2 y q3, las longitudes L2, L3 solo se verían sus proyecciones, sobre el plano x, y del sistema S0, por lo tanto esta proyección se denomina D.

Para calcular el ángulo q1 se pone el observador en dirección perpendicular al eje Z0, y se hace girar el robot un ángulo entre 0° y 90°

$Pm = [Pmx \ Pmy \ Pmz]$ Pm guarda las coordenadas tanto en x, y, z del punto de la muñeca del robot.

$q1 = \alpha - \beta$. El ángulo q1 es la diferencia que existe entre α y β mostrados en la Figura 4.1.2. 1.

$\alpha = \text{atan2}(Pmy, Pmx)$. Es el valor obtenido entre las coordenadas en Y e Z del punto de la muñeca.

El valor de la longitud D se deduce a partir de la línea auxiliar m.

$$\beta = \text{atan2d}(L1, D)$$

$$m = \sqrt{Pmx^2 + Pmy^2}$$

$$D = \sqrt{m^2 - L1^2}$$

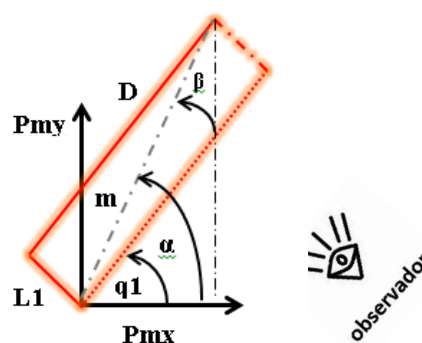


Figura 4.1.2. 2 Cálculo del ángulo q1

Ya obtenida el sistema A_1^0 , a partir de los parámetros de DH se procede a rotar el sistema 90° en torno a X, como lo hecho en la cinemática directa

$$A_1^0 = \text{rotx}(-90) * A_1^0$$

$$A_1^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * A_1^0$$

Se procede a calcular el ángulo q_2 , poniendo la vista en dirección perpendicular a D como lo mostrado en la *Figura 4.1.2. 2*. Observe que para el ángulo q_2 solo se tiene en cuenta la componente P_{mz} , y D lo que resulta fácil al momento de calcular el ángulo q_2

$q_2 = \text{atan2d}(-P_{mz}, D)$, la componente P_{mz} es negativa debido a que el sentido de P_z va hacia abajo *Figura 4.1.2. 3*

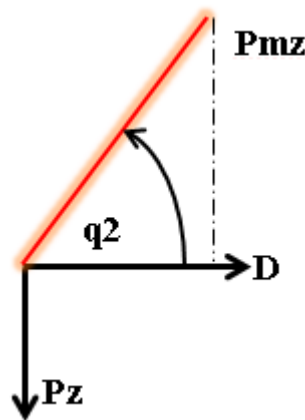


Figura 4.1.2. 3 Cálculo del ángulo q_2

Ya calculados los ángulos q_1 , q_2 y con los resultados del modelo cinemático directo se obtiene $A_2^0 = A_1^0 * A_2^1$ y se determinan los vectores Z_2 , Y_2 , Z_2 . Y debido a la configuración del brazo S_2 y S_5 están perpendiculares en todo momento. Por la configuración el vector Z_3 siempre es perpendicular a Z_2 y Z_5 , por lo tanto

$$Z_3 = \pm \frac{Z_2 \times Z_5}{\|Z_2 \times Z_5\|}$$

Se divide sobre la norma con el propósito de generar un vector unitario, que tiene dos resultados +1 y -1, lo que indica que el robot puede girar tanto en dirección positiva o negativa.

Para poder calcular el ángulo q_3 se coloca el observador en dirección a Z_2 . Como solo se conoce el valor de Z_3 y los otros valores se desconocen, por lo tanto solo se utiliza Z_3

Se aplica el producto punto entre Y_2 y Z_3

$$\begin{aligned} Y_2 \cdot Z_3 &= Cq_3 \\ X_2 \cdot Z_3 &= \cos(90 + q_3) \\ \cos(90) Cq_3 - \sin(90) Sq_3 \\ X_2 \cdot Z_3 &= Sq_3 \end{aligned}$$

Por lo cual se obtiene.

$$q_3 = \text{atan2d}(Sq_3, Cq_3)$$

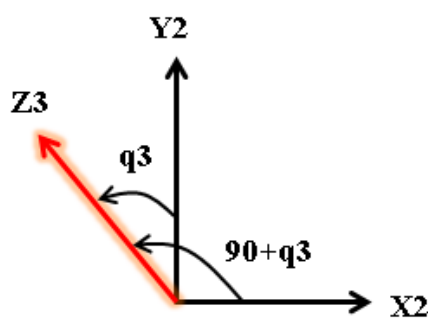


Figura 4.1.2. 4 Cálculo del ángulo q_3

Teniendo el ángulo q_3 , se obtiene a partir del modelo cinemático directo se obtiene la matriz $A_3^0 = A_2^0 * A_3^2$ y se determinan los vectores X_3, Y_3, Z_3 .

Para encontrar el S_4 hay que tener en cuenta que X_4 se desconoce y solo se conoce Z_4 aplicando la propiedad del producto punto entre vectores se deduce que

Se aplica el producto punto entre Y_3 y Z_4

$$\begin{aligned} Y_3 \cdot Z_4 &= \cos(180 - q_4) \\ \cos(180) Cq_4 + \sin(180) \sin q_4 \\ -X_3 \cdot Z_4 &= Cq_4 \end{aligned}$$

Aplicando el producto punto entre x_3 y Z_4

$$\begin{aligned} X_3 \cdot Z_4 &= \cos(90 - q_4) \\ \cos(90) cq_4 + \sin(90) Sq_4 \end{aligned}$$

$$X_3 \cdot Z_4 = Sq_4$$

Por lo cual se obtiene

$$q_4 = \text{atan2d}(Sq_4, Cq_4)$$

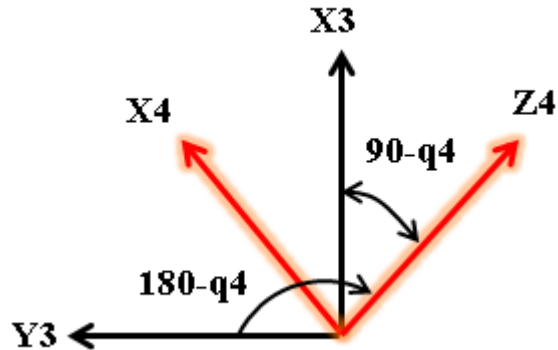


Figura 4.1.2. 5 Cálculo del ángulo q_4

Ya calculado el ángulo q_4 , y a partir del modelo cinemático directo se obtiene la matriz $A_4^0 = A_3^0 * A_4^3$ y se determinan los vectores X_4, Y_4, Z_4 .

Por último se sitúa el observador en dirección opuesta a Z_4 y se analiza los sistemas S_4 y S_5 , en este caso se conocen todos los ejes, por lo tanto el análisis se puede hacer con cualquiera de ellos.

Se aplica el producto punto entre X_4 y X_5

$$\begin{aligned} X_4 X_5 &= Cq_5 \\ Y_4 X_5 &= \cos(90 - q_5) \\ \cos(90) c q_5 + \sin(90) S q_5 \end{aligned}$$

Aplicando el producto punto entre Y_4 y X_5

$$Y_4 X_5 = S q_5$$

Se obtiene

$$q_5 = \text{atan2}(S q_5, C q_5)$$

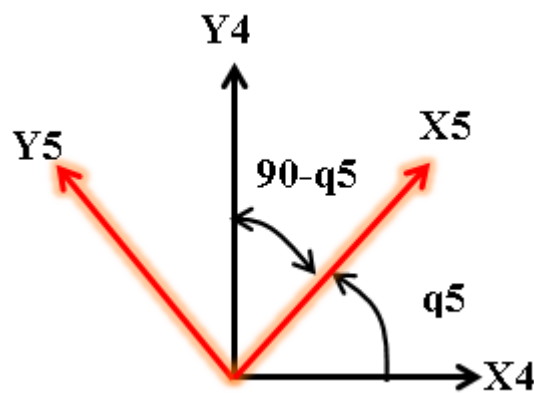


Figura 4.1.2. 6 Cálculo del ángulo q_5

Por último se organizan los ángulos en un vector para mayor comodidad

$$q = [q_1 \ q_2 \ q_3 \ q_4 \ q_5]$$

4.2. Interpretación De Los Ángulos

El dispositivo de luz estructurada (Kinect), permite identificar los puntos de un cuerpo humano y realizar la construcción del esqueleto entre esos puntos e identificar de forma geométrica los **4.3 Cálculos de los ángulos referidos para mover el robot** y así poder hacer la teleoperación del brazo robótico de InMoov propuesto en este trabajo.

4.2.1 cambio de eje coordenado

El dispositivo de luz estructurada (Kinect) y el software utilizado presentan diferentes ejes coordenados, lo cual impide tener una apreciación correcta del esqueleto. Debido a que la vista no es la más adecuada que ofrece. Como se puede ver en la siguiente *Figura 4.2.1. 1*

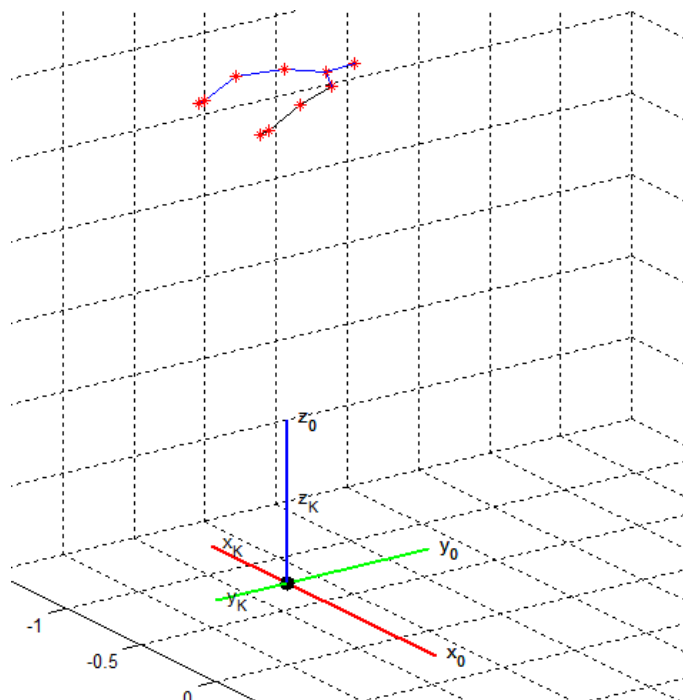


Figura 4.2.1. 1 vista del esqueleto según el sistema de eje coordenado del Kinect (k)

Para corregir el sistema coordenado y apreciar mejor los puntos del esqueleto se tienen que hacer ajustes en el sistema coordenado del Kinect, y para ello se debes hacer una rotación del sistema de Kinect con respecto al de Matlab.

De la figura anterior podemos obtener la matriz 0A_k

Ecuación 4.2.1. 1 MTH 0A_k

$${}^0A_k = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Para encontrar el punto del sistema fijo \mathbf{p}_0 necesitamos pre multiplicar la matriz correspondiente al sistema de Kinect con respecto al fijo 0A_k y con el punto del Kinect \mathbf{p}_k , de esta forma se puede calcular el punto del sistema fijo \mathbf{p}_0

Ecuación 4.2.1. 2

$$\mathbf{p}_0 = {}^0A_k * \mathbf{p}_k$$

Debido a que la vista no es la más adecuada para mostrar los puntos del esqueleto, se hace una rotación con respecto al eje fijo de -90°

Ecuación 4.2.1. 3

$$\mathbf{p}'_0 = \text{rot}_x(-90) * \mathbf{p}_0$$

Reemplazando la *Ecuación 4.2.1. 2* en la *Ecuación 4.2.1. 3* se obtiene:

Ecuación 4.2.1 4

$$\mathbf{p}'_0 = \text{rot}_x(-90) * {}^0A_k * \mathbf{p}_k$$

Donde la matriz de transformación homogénea de la rotación en x corresponde a:

Ecuación 4.2.1 5 MTH rotación en x

$$\text{rot}_x = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ecuación 4.2.1. 6 Rotación de -90° en el eje x

$$\text{rot}_{(-90)} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Reemplazando las *Ecuación 4.2.1. 1* y *Ecuación 4.2.1. 6* en la *Ecuación 4.2.1 4* obtenemos:

Ecuación 4.2.1. 7

$$p'_0 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} p_k$$

Resolviendo la matriz de la *Ecuación 4.2.1. 7* da como resultado.

Ecuación 4.2.1. 8

$$p'_0 = \begin{bmatrix} p'_{0x} \\ p'_{0y} \\ p'_{0z} \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_{kx} \\ p_{ky} \\ p_{kz} \\ 1 \end{bmatrix} = \begin{bmatrix} -p_{kx} \\ p_{kz} \\ p_{ky} \\ 1 \end{bmatrix}$$

De esta forma ya se ha corregido el sistema Kinect con respecto al de Matlab y así poder apreciar de una forma mucho más clara los puntos del esqueleto

Ecuación 4.2.1. 9

$$p'_0 = \begin{bmatrix} p'_{0x} \\ p'_{0y} \\ p'_{0z} \\ 1 \end{bmatrix} = \begin{bmatrix} -p_{kx} \\ p_{kz} \\ p_{ky} \\ 1 \end{bmatrix}$$

De esta forma se garantiza que ambos sistemas (cero y Kinect) serán iguales en todo momento. Dando como resultado una vista más adecuado para apreciar los puntos generados por el Kinect *Figura 4.2.1. 2*

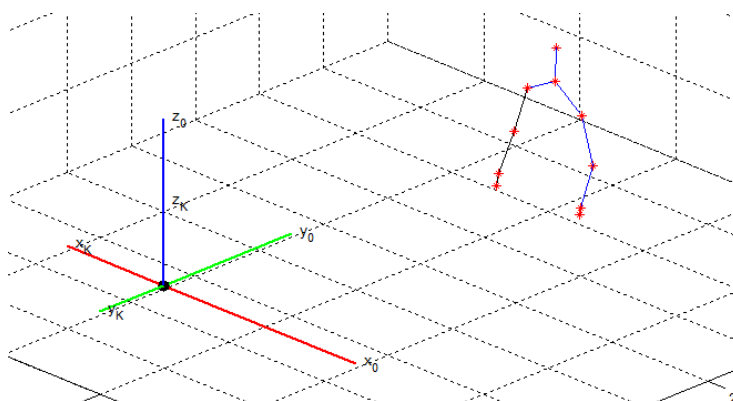


Figura 4.2.1. 2 Corrección del sistema de Kinect con respecto al de Matlab

Tener los dos sistemas tanto como el de Kinect como el de Matlab, puede llegar a confundir un poco a la hora de hacer los cálculos matemáticos, es por ello q el

sistema de ejes coordenados del Kinect se ha quitado para mayor comodidad como se muestra en la figura a continuación. *Figura 4.2.1. 3*

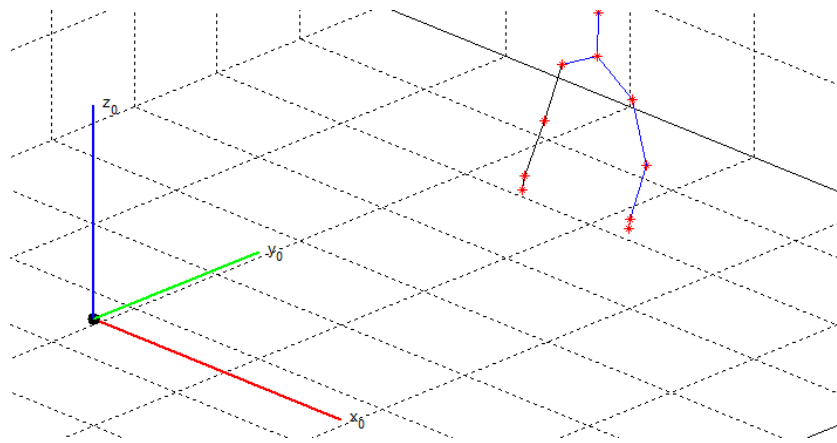


Figura 4.2.1. 3 vista del sistema coordenado con los puntos del esqueleto sin el sistema de Kinect

4.3 Cálculos de los ángulos referidos para mover el robot

El cálculo de los ángulos del brazo derecho se desarrolla con base en el sistema coordenado que se genera en el hombro para mayor facilidad.

Para el cálculo de los ángulos de las articulaciones del esqueleto, hay que tener en cuenta que el dispositivo de luz estructurada genera tres vectores (vec_x , vec_y , vec_z) de tamaño 20×1 . Estos vectores guardan las coordenadas X, Y, Z de la posición de cada articulación.

Teniendo en cuenta las coordenadas de los vectores, se genera así el vector del hombro derecho con respecto al sistema Matlab. De la misma manera el vector del hombro izquierdo con respecto al sistema Matlab. Estos puntos se obtienen dependiendo de la posición de cada articulación, en este caso los puntos 9 y 5 correspondiente al hombro derecho e izquierdo como se muestra en la Figura 4.3. 1

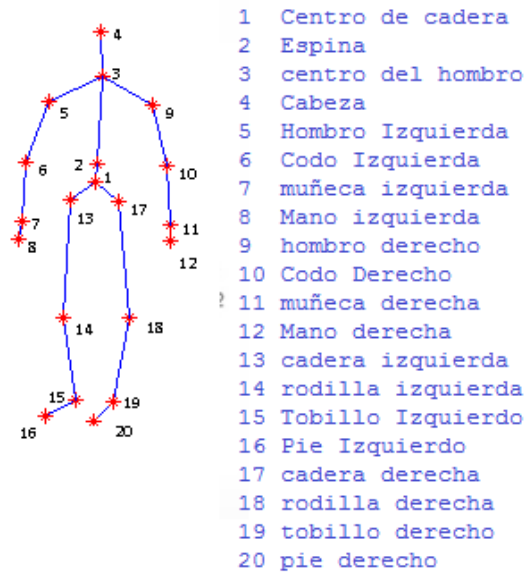


Figura 4.3. 1 captura del esqueleto completo con numeración de cada articulación

$${}^m p_h = \begin{bmatrix} \text{vecxs}(9) \\ \text{vecys}(9) \\ \text{veczs}(9) \end{bmatrix}, {}^m p_{hi} = \begin{bmatrix} \text{vecxs}(5) \\ \text{vecys}(5) \\ \text{veczs}(5) \end{bmatrix}$$

Se genera un vector resultante entre la resta del hombro derecho e izquierdo con respecto al sistema Matlab con el propósito de tener una mayor precisión a la hora de generar un sistema de eje coordenado en el hombro

$${}^m p_{mh} = {}^M p_h - {}^M p_{hi}$$

Teniendo en cuenta que el primer grado de libertad es el de aducción-abducción se genera la posición del vector z^{Mh} en la dirección de ${}^m p_{mh}$

$$\widehat{z^{Mh}} = \frac{{}^m p_{mh}}{\|{}^m p_{mh}\|}. \text{ Vector unitario, } z \text{ hombro con respecto a Matlab}$$

Para posicionar el vector y^{Mh} se posiciona en la misma dirección de Z_o , con el propósito de que los movimientos de los ángulos sean siempre positivos

$$y^{Mh} = [0 \ 0 \ 1]'$$

Con los vectores z^{Mh} y y^{Mh} se calcula x^{Mh} para cumplir de esta forma la teoría de dextrógiro

$$x^{Mh} = (|y^{Mh} \times z^{Mh}|).$$

Con los vectores x^{Mh} , y^{Mh} , z^{Mh} se genera la matriz donde se almacenara los vectores XYZ del hombro con respecto a Matlab

$${}^m A_h = \begin{bmatrix} x^{Mh} & y^{Mh} & z^{Mh} & p^{Mh} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Con la matriz ${}^M A_h$ se dibuja el sistema coordenado en el hombro. Figura 4.3. 2

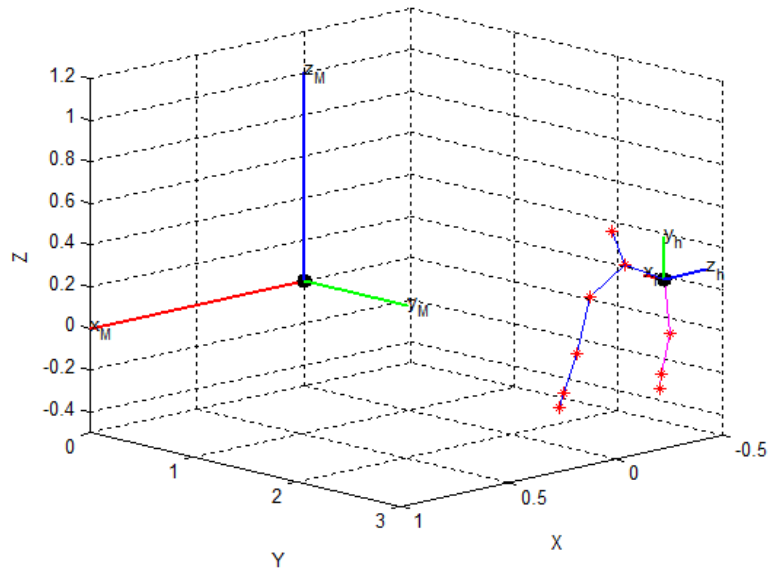


Figura 4.3. 2 Sistema coordenado en el hombro

Ahora se tiene dos sistemas de ejes coordenados para poder calcular los ángulos de cada una de las articulaciones del brazo, el sistema S_0 correspondiente al sistema de Matlab y el sistema SMh correspondiente al sistema del hombro con respecto a Matlab. Para mayor comodidad se trabaja con el sistema en el hombro SMh en inversa para que los ángulos dependan del sistema en el hombro y no del sistema del Matlab

$${}^h A_M = {}^M A_h^{-1} \text{ Inversa de la matriz } A_{mh}$$

Al momento de calcular el ángulo q_1 correspondiente al movimiento de aducción – abducción se necesita calcular la dirección en la cual se encuentra en el codo, utilizando la **Figura 5.1. a** se sabe con exactitud cuáles son los puntos correspondientes al codo

$${}^m p_c = \begin{bmatrix} \text{vecxs}(10) \\ \text{vecys}(10) \\ \text{veczs}(10) \\ 1 \end{bmatrix}$$

Vector del codo con respecto al sistema de referencia

El vector ${}^m p_c$ corresponde al vector del codo con respecto a Matlab, no es el adecuado para solucionar el ángulo q_1 , es ahí donde entra la matriz

correspondiente al sistema del hombro hA_M , de esta forma el punto en el codo dependerá del sistema en el hombro Sh y no del sistema de Matlab $S0$

$${}^h p_c = {}^h A_M * {}^M p_c .$$

Punto hombro codo

Ya solo queda calcular el ángulo $q1$, y el movimiento q este efectúa depende del movimiento en eje ${}^h p_c$ en Z y el ${}^h p_c$ en Y *Figura 4.3. 3*

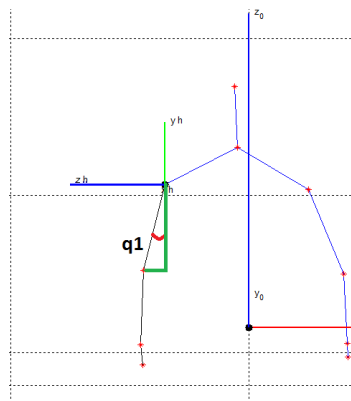


Figura 4.3. 3 Cálculo del ángulo q1

$$q1 = \text{atan2d}({}^h p_c(3), -{}^h p_c(2))$$

Ángulo $q1$ de abducción – aducción

El cálculo del ángulo de flexión – extensión depende en gran medida del ángulo $q1$, ya que este movimiento se encuentra conectada en un mismo punto en el hombro, por lo tanto

$$yp = \text{norm}(\text{Rotx}(-q1) * {}^h p_c(2)) \text{ Punto del codo con rotación del Angulo } q1$$

Una forma para saber si el movimiento es positivo o negativo se analiza el valor que posee la posición en el punto ${}^h p_c(2)$ de esta forma se sabe si el movimiento va hacia arriba o hacia abajo

$$sig = \text{sign}({}^h p_c(2)) \text{ Sig es el signo } \pm \text{ que puede tomar } phc(2)$$

Ya solo queda calcular el ángulo $q2$ correspondiente al movimiento de flexión – extensión *Figura 4.3. 4*

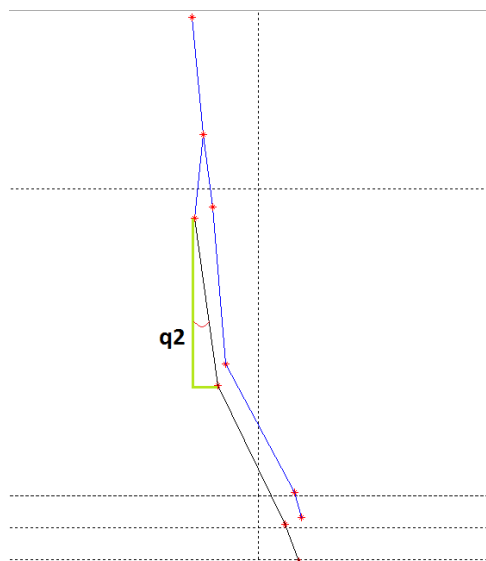


Figura 4.3. 4 Cálculo del ángulo q_2

$q_2 = sig * (atan2d(y_p, h_p_c(1)) + 90$ Angulo de flexión - extensión del hombro

De la misma manera como fueron analizados los punto anteriores, teniendo en cuenta la **figura 5.1.e** se observa que el punto correspondiente a la muñeca dado por el Kinect es el siguiente.

$${}^M p_m = \begin{bmatrix} vecxs(11) \\ vecys(11) \\ veczs(11) \\ 1 \end{bmatrix} \text{ Vector Matlab muñeca}$$

Para analizar el ángulo correspondiente al de torsión, primero debemos pasar el punto de muñeca con respecto a Matlab ${}^M p_m$ al del hombro, de esta forma dependerá del sistema Sh y no del $S0$

$${}^h p_m = {}^h A_M * {}^M p_m \text{ Punto hombro muñeca}$$

El ángulo de torsión es un poco más complejo si se analiza su movimiento con respecto al movimiento de aducción- abducción y el movimiento de flexión extensión, una forma de hacer más cómoda el análisis es crea un plano entre el hombro y el codo de esta forma el análisis solo dependerá del plano sin importar los movimientos anteriores.

El cálculo del ángulo de torsión q_3 se calcula en base a un plano ubicado en el brazo, este plano contiene 4 puntos que varía según los cambio en las coordenadas X, Y, Z (p_1, p_2, p_3). De los puntos hombro y codo con respecto al sistema $S0$ de Matlab

$$p_1 = [pMh(1), pMc(1), pMc(1), pMh(1)]$$

$$p_2 = [pMh(2) + 0.06, (pMc(2) + 0.06), (pMc(2) - 0.06), pMh(2) - 0.06]$$

$$p3 = [pMh(3), pMc(3), pMc(3), pMh(3)]$$

Por último se aplica el teorema del coseno y encontrar el ángulo producido entre el plano y la muñeca. Figura 4.3. 5

$$q3 = 180 - \text{acosd} \left(\frac{[p1(2) p2(2) p3(3)] p hm(3)}{\| [p1(2) p2(2) p3(3)] \| \| p hm(3) \|} \right) \text{ Angulo de torsión del brazo derecho}$$

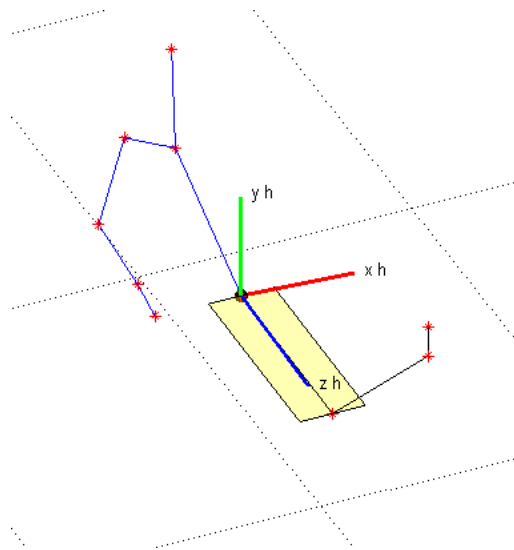


Figura 4.3. 5 plano para cálculo de q3

Para terminar con el análisis de los cálculos de los ángulos, se analiza el movimiento de flexión – extensión pero esta vez en el codo

Para comenzar hacemos una diferencia entre el codo y la muñeca y así tener una referencia entre los dos puntos

Diferencia entre vectores codo-muñeca

$$vcm = {}^h p_c - {}^h p_m$$

$$phc(1:3) = {}^h p_c = \begin{bmatrix} {}^h p_{cx} \\ {}^h p_{cy} \\ {}^h p_{cz} \end{bmatrix}$$

$$, vcm(3) = vcm = \begin{bmatrix} vcm_x \\ vcm_y \\ vcm_z \end{bmatrix}$$

Aplicación teorema del coseno para el ángulo del codo y encontrar el ángulo que se forma, a este valor se le resta un valor de 180. Para que el ángulo quede de 0 a 180 y no viceversa. *Figura 4.3. 6*

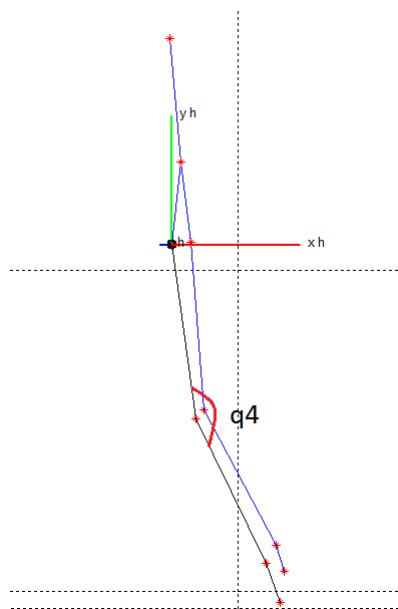


Figura 4.3. 6 Calculo del ángulo q_4

$$q_4 = 180 - \frac{\text{acosd}(phc(1:3) \cdot vcm(1:3))}{\|phc(1:3)\| * \|vcm(1:3)\|}$$

El sistema de luz estructura da Kinect tiene ciertas limitaciones, una de ellas es la captura del movimiento de pronación - supinación en la muñeca, este movimiento es tan pequeño que Kinect no alcanza a captarlo, por ello el ángulo q_5 correspondiente al último grado de libertad del brazo robótico es dado por el usuario

4.4 Control Posición – Posición

El movimiento del brazo robótico de InMoov ocurre de tal forma que los movimientos son de articulación por articulación y no de forma multiarticular,

para que el brazo pueda ser controlado se necesita de un usuario que haga los movimientos necesarios para la teleoperación del brazo robótico, esta información es analizada por una computadora que tiene conectado a su vez el dispositivo de luz estructurada Kinect que interpreta los puntos de las articulaciones, la computadora se encarga de calcular por medio de los puntos los ángulos que son enviados a la arduino que hace a su vez la función de cerebro e indica a los servomotores el ángulo al cual deben llegar y darle movimiento al brazo.

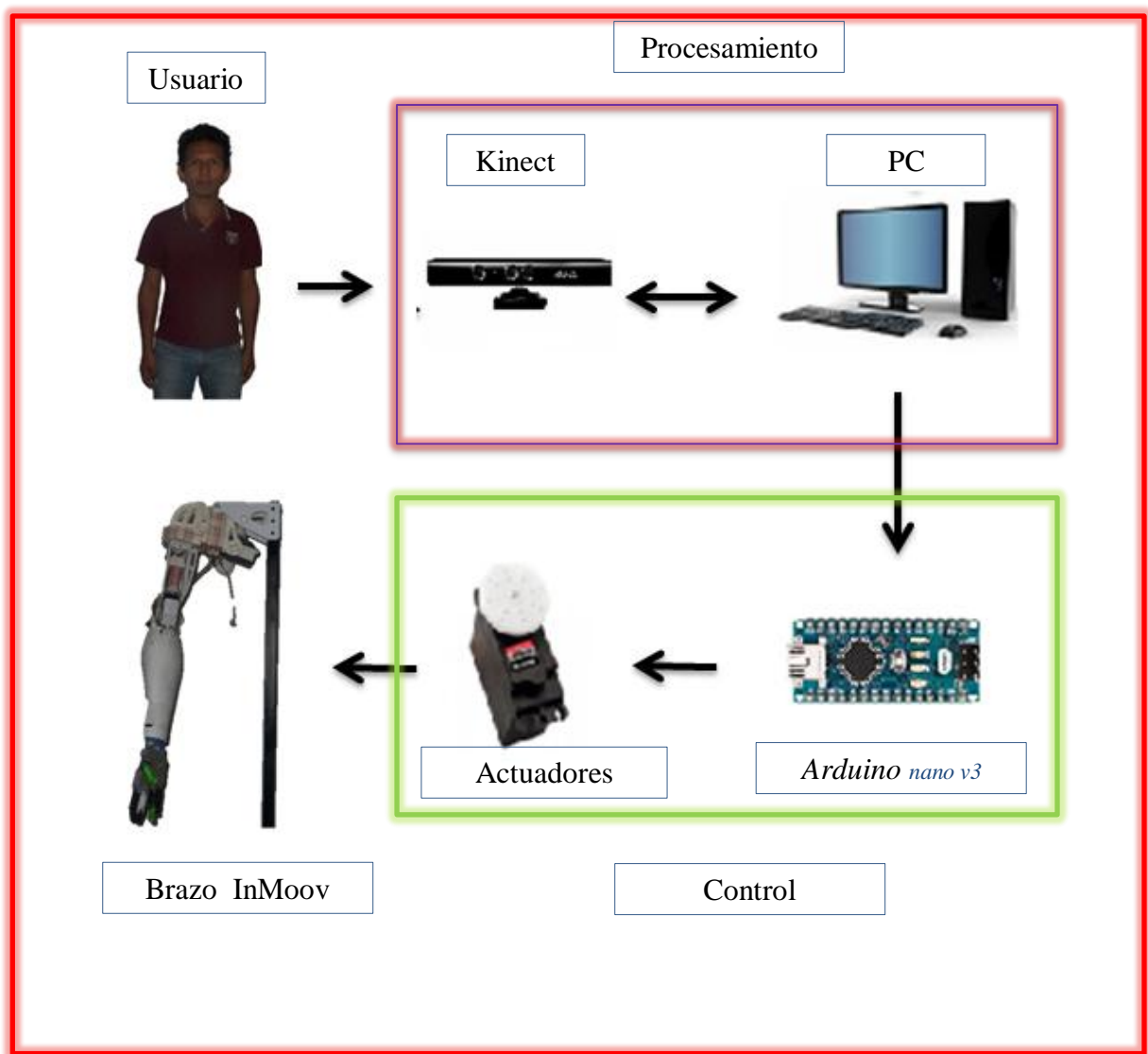


Figura 4.4. 1 arquitectura del sistema de control

Capítulo 5

Resultados

En este capítulo se presenta los resultados obtenidos en este proyecto de grado, y para avalar los resultados se dan pruebas para el cumplimiento de los objetivos de este trabajo que era realizar la puesta a punto y las pruebas de verificación. Esto se llega gracias a la construcción del brazo robótico de InMoov. En la Figura 5. 1 se puede apreciar el resultado final del brazo de InMoov

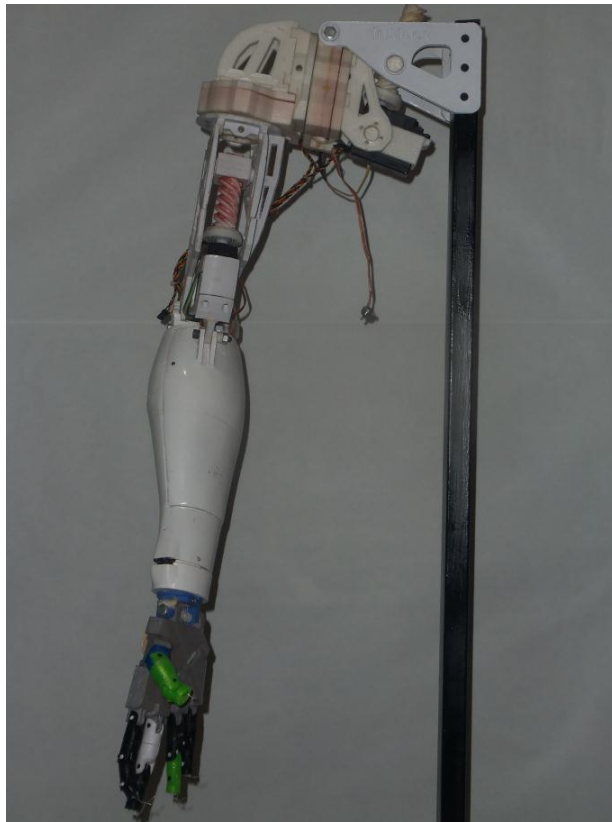


Figura 5. 1 resultado final del brazo robótico InMoov

Este tipo de movimiento se logra debido al gran torque que tienen los motores y a la relación que cuenta en los engranes del hombro, este movimiento es muy usual en el ser humano un ejemplo de ello es al momento de caminar realizamos el movimiento de flexión extensión

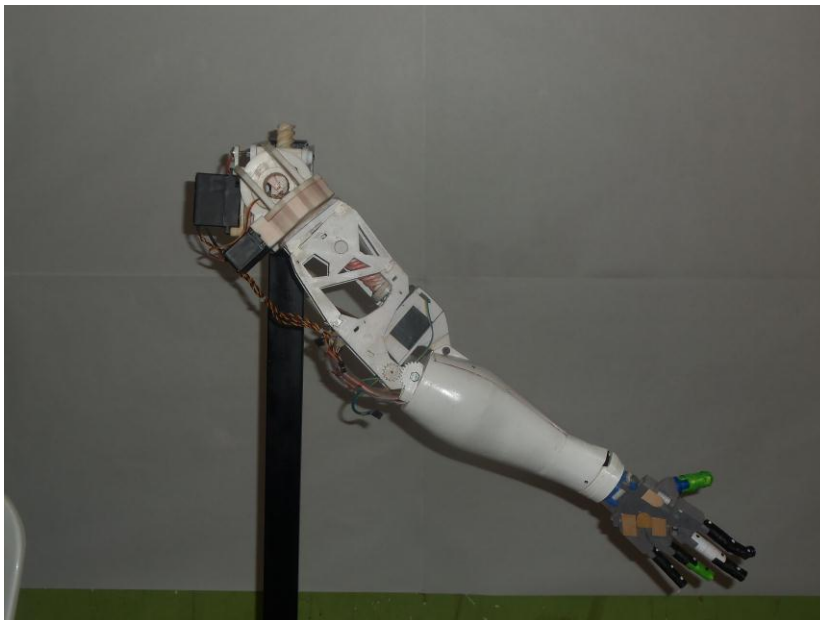


Figura 5. 2 movimiento de flexión extensión

El movimiento de torsión permite hacer cambio de orientación del brazo entre 0° a 90° *Figura 5. 3* lo que permite agarrar objetos que se encuentran desplazados a un lado



Figura 5. 3 movimiento de torsión

En la misma *Figura 5. 3* podemos apreciar también el movimiento del codo, este movimiento tiene una peculiaridad y es el tiempo de estabilización del motor esto es debido que el servomotor cuenta con un circuito diferente al original lo que ocasiona oscilaciones hasta lograr a su setpoint



El movimiento de pronación supinación permite al usuario girar la muñeca y orientar el efector final (mano) en la posición deseada

DQS is member of:



Capítulo 6

Conclusiones

El tiempo de respuesta del movimiento del brazo de InMoov es lento debido a la relación de engranes que posee alrededor de 14:1, esto conlleva que las acciones no ocurran en tiempo real y se tenga que esperar que el brazo llegue a la posición deseada e incluso se llegan a perder posiciones si la señal es demasiado rápida en comparación con el movimiento efectuado por el brazo robótico

Los cálculos realizados para la interpretación de los ángulos, son aproximaciones geométricas esto debido al error que posee el sistema de luz estructurada Kinect que es alrededor de un 7% es decir más o menos 13°, por ende la interpretación entre el ángulo ofrecido entre la parte mecánica y el movimiento real del sujeto no es completamente precisa.

La interfaz gráfica permite al usuario tener una comprensión más notable del movimiento efectuado por las articulaciones, escoger el tipo de postura o de vista que se desee sin necesidad de interactuar directamente con el programa

Los resultados obtenidos con el brazo robótico fueron muy satisfactorios, se cumplieron en gran medida los objetivos planteados en este trabajo, lo que lleva a plantearse un nuevo reto y es seguir mejorando su parte mecánica y poder construir el brazo izquierdo con el propósito de añadir la parte de haptica y visión

Bibliografía

- ACKERMAN. (02 de 2015). *spectrum iee*. Obtenido de <http://spectrum.ieee.org/automaton/robotics/medical-robots/darpa-and-osrfdeveloping->
- Alegsa. (16 de 02 de 2017). Obtenido de http://www.alegsa.com.ar/Dic/microsoft_kinect.php
- Aliaga. (12 de 2000). *tecnun*. Recuperado el 15 de 02 de 2017, de <http://www4.tecnun.es/asignaturas/control1/proyectos/teleop2D/teleoperacion>.
- Arduino. (2017). Obtenido de <https://www.arduino.cc/en/pmwiki.php?n=Main/ArduinoBoardNano>
- CCM. (09 de 2014). *salud y bienestar*. Obtenido de <http://salud.ccm.net/faq/12670-antebrazo-definicion>
- DARPA. (12 de 2016). Obtenido de <http://www.darpa.mil/news-events/2016-12-22>
- dynamo. (2017). Obtenido de <https://www.dynamoelectronics.com/servomotores-rc/286-servo-hs805bb-hitec.html>
- EL NUEVO DIARIO. (12 de 2016). Obtenido de <http://www.elnuevodiario.com.ni/suplementos/tecnologia/414045-protesis-roboticas-e-implantes-cerebrales/>
- engadget. (05 de 2011). Obtenido de <https://www.engadget.com/2011/04/05/amo-arm-pneumatic-prosthetic-does-mind-controlon->
- Gael. (2017). *InMoov Galeria v2*. Obtenido de <http://inmoov.fr/gallery-v2/>
- han, T. (09 de 2017). *robinhsieh*. Obtenido de http://robinhsieh.com/inmoov_arm/
- ISO, & IFR. (s.f.). *AER*. Obtenido de <http://www.aeratp.com/aer-atp/robotica-industrial-y-de-servicio/>
- JEFF. (08 de 2014). *ccm salud*. Obtenido de <http://salud.ccm.net/faq/19063-mano-definicion>
- Langevin , G. (2012). *InMoov Open Source 3D Printed*. Obtenido de <http://inmoov.fr/>
- NASA. (10 de 2016). Obtenido de <https://robonaut.jsc.nasa.gov/R2/>
- Salvatore, J., Osio, J., & Morales, J. (22 de 07 de 2014). Detección de objetos utilizando el sensor Kinect. *LACCEI*, <http://www.laccei.org/LACCEI2014-Guayaquil/RefereedPapers/RP178.pdf>. Obtenido de <http://www.laccei.org/LACCEI2014-Guayaquil/RefereedPapers/RP178.pdf>



STRICKLAN. (06 de 2015). *spectrum iee*. Obtenido de <http://spectrum.ieee.org/biomedical/bionics/a-better-way-for-brains-to-control-roboticarms>

towerpro. (2016). *tower pro*. Obtenido de <https://www.towerpro.com.tw/product/mg995/>

WEVOLNER. (2016). Obtenido de <http://blog.wevolver.net/post/121037950004/the-bionic-arm-you-control-with-muscle-sensors-and-smarthphone>

xataka. (05 de 2015). Obtenido de <https://www.xataka.com/robotica-e-ia/los-brazos-y-piernas-roboticas-con-sensores-ycontrolados->

Apéndice A

Conexiones Eléctricas Del Brazo Robótico

Esquemático de las conexiones de los componentes para el funcionamiento del brazo robótico de InMoov.

Si se desea trabajar de forma inalámbrica, se debe habilitar la conexión al jumper para la alimentación de la arduino y el modulo bluetooth.

VCC	Alimentación 5 a 6 V
SP	servomotor mg995
SG	servomotor hs805bb
ARD	Arduino nano v3
BLT	Modulo bluetooth
JUM	Jumper

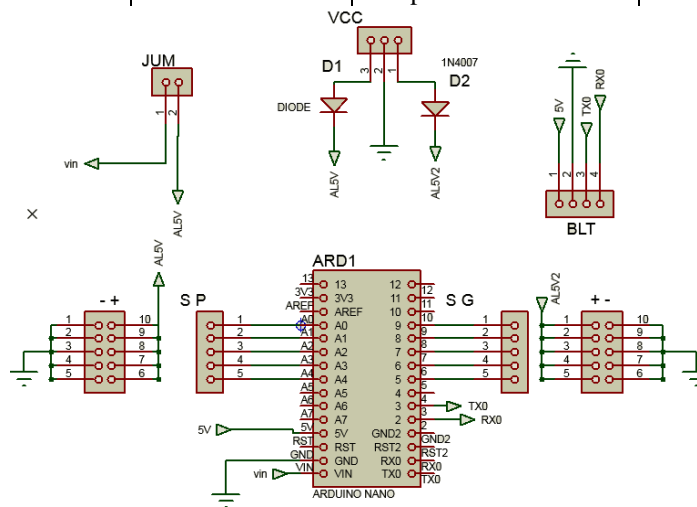


Figura A 1 conexión de los componentes del circuito

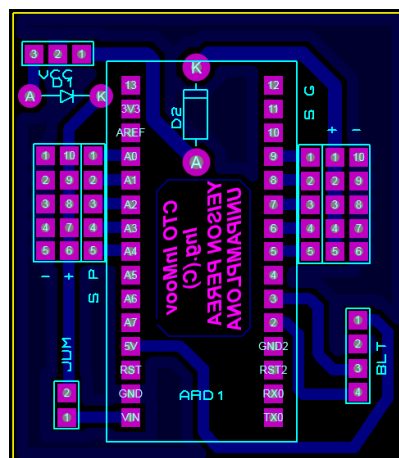


Figura A 2 conexión del circuito

Apéndice B

Código de optimización de Matlab – Kinect

Conexión Matlab-Kinect

% Esta función permite hacer conexión con el dispositivo de luz estructurada (Kinect) ,Matlab y Arduino, guardar las variables que este detecta, escoger el tipo de postura que desea detectar y el tipo de vista 3d o 2d

```
clc; clear; close all;clear port;
```

```
%targetinstaller
```

```
% Variables globales
```

```
global funcionando
```

```
funcionando = 1;
```

```
% Variables
```

```
POSTURA_CUERPO = 1; % Sentado: 0 % Parado: 1
```

```
DIBUJAR_ESQUELETO_2D = 0;%bandera
```

```
parar = 0;
```

```
% ard=1;
```

```
ard=arduino('COM3');
```

```
%% Configuración kinect analisis continuo
```

```
imaqreset; % Inicializando configuración de la cámara
```

```
depthVid = videoinput('kinect', 2);
```

```
triggerconfig(depthVid, 'manual');
```

```
depthVid.FramesPerTrigger = 1;
```

```
depthVid.TriggerRepeat = inf;
```

```
set( getselectedsource(depthVid), 'TrackingMode', 'Skeleton');
```

```
colorVid = videoinput('kinect', 1);
```

```
triggerconfig(colorVid, 'manual');
```

```
colorVid.FramesPerTrigger = 1;
```

```
colorVid.TriggerRepeat = inf;
```

```
src = getselectedsource (depthVid);
```

```
%% configurar angulo de la camara
```

```
set(src, 'CameraElevationAngle',10);
```

```
% set(src, 'CameraElevationAngle',10);
```

```
%% Selecciona la postura del cuerpo
```

```
if POSTURA_CUERPO ==1
```

```
set(src, 'BodyPosture','Standing');
```

```
else
```

```
set(src, 'BodyPosture','Seated');
```

```
end
```

```
% Inicio
```

```
start(depthVid);
```

```
start(colorVid);
```

```
% Crea figura donde se va visualizar el esqueleto
```

```
himg = figure(1);
```

```
set(himg,'Name', 'SKELETON','Color',[1,1,1]);
```

```
% Crea botones de Cerrar, Parar y configura el de salir
```

```
uicontrol('String','Cerrar','Position',[400 15 120 20],...
```

```
'Callback','funcionando=0;, delete(gcf);');
```

```
uicontrol('String','Parar','Position',[200 15 120 20], ...
```

```
'Callback','funcionando = 0;, parar=1');
```

```
set(gcf,'CloseRequestFcn','funcionando=0;, delete(gcf);')
```

```
% Ciclo de captura del esqueleto
```

```
while funcionando
```

```
est_esqueleto
```

```
=captura_esqueleto(colorVid,depthVid,POSTURA_CUERPO,DIBUJAR_ESQUELETO_2D);
```




```
if est_esqueleto.nrSkeleton > 0
    clf;
    dibujar_esqueleto_3d_s3(est_esqueleto,POSTURA_CUERPO,ard);%sistema corregido
    %dibujar_esqueleto_3d(est_esqueleto);
end
end
% finaliza
if not(parar)
    close(gcf);
end
stop(depthVid);
stop(colorVid);
```

Captura de los puntos de las articulaciones

```
% función que permite guardar cada uno de los puntos detectados por el Kinect
function est_esqueleto = captura_esqueleto(colorVid,depthVid,postura_cuerpo,dib2d)
% Captura de datos
trigger(colorVid);
trigger(depthVid);
est_esqueleto.image=getdata(colorVid);
[est_esqueleto.depthMap,~, est_esqueleto.depthMetaData]=getdata(depthVid);
est_esqueleto.nrSkeleton=sum(est_esqueleto.depthMetaData.IsSkeletonTracked);
if dib2d
    % Visualiza la imagen RGB
    figure(1);
    imshow(est_esqueleto.image);
end
% En caso que detecte al menos un esqueleto se ejecuta:
if est_esqueleto.nrSkeleton > 0
    % Captura posición de puntos del esqueleto en la imagen
    est_esqueleto.skeletonJoints = ...
        est_esqueleto.depthMetaData.JointImageIndices...
        (:, :, est_esqueleto.depthMetaData.IsSkeletonTracked);
    % Búsqueda del primer esqueleto de los detectados
    est_esqueleto.esq_encontrados = ...
        find (est_esqueleto.depthMetaData.IsSkeletonTracked == 1);
    est_esqueleto.esqueleto_seleccionado= est_esqueleto.esq_encontrados(1);
    % Posición general del esqueleto
    est_esqueleto.posicion_esqueleto_seleccionado = ...
        est_esqueleto.depthMetaData.PositionWorldCoordinates...
        (:,est_esqueleto.esqueleto_seleccionado);
    % Visualiza el esqueleto en la imagen
    if dib2d
        if postura_cuerpo==1
            util_skeletonViewer(est_esqueleto.skeletonJoints,...
                est_esqueleto.image, est_esqueleto.nrSkeleton);
        end
        if postura_cuerpo==0
            util_skeletonViewer(est_esqueleto.skeletonJoints,...
                est_esqueleto.image, est_esqueleto.nrSkeleton);
        end
    end
    % OPCIONAL POR SI SE DESEA TRABAJAR CON LOS PUNTOS DEL ESQUELETO
    % Almacena de forma cómoda los puntos del esqueleto (posición relativa)
    est_esqueleto.relvecx=...
        est_esqueleto.depthMetaData.JointWorldCoordinates...
        (:,1,est_esqueleto.esqueleto_seleccionado);
    est_esqueleto.relvecy=...
        est_esqueleto.depthMetaData.JointWorldCoordinates...
        (:,2,est_esqueleto.esqueleto_seleccionado);
    est_esqueleto.relvecz=...
        est_esqueleto.depthMetaData.JointWorldCoordinates...
        (:,3,est_esqueleto.esqueleto_seleccionado);
    % Almacena de forma cómoda los puntos del esqueleto (posición absoluta)
    est_esqueleto.vecx=est_esqueleto.relvecx+...
        est_esqueleto.depthMetaData.PositionWorldCoordinates...
        (1,est_esqueleto.esqueleto_seleccionado);
    est_esqueleto.vecy=est_esqueleto.relvecy+...
        est_esqueleto.depthMetaData.PositionWorldCoordinates...
        (2,est_esqueleto.esqueleto_seleccionado);
    est_esqueleto.vecz=est_esqueleto.relvecz+...
```



```
est_esqueleto.depthMetaData.PositionWorldCoordinates...  
(3,est_esqueleto.esqueleto_seleccionado);  
End
```

DQS is member of:



Dibujar el esqueleto en 3D

%esta función permite dibujar el esqueleto en 3d

```
function dibujar_esqueleto_3d_s3(est_esqueleto,POSTURA_CUERPO,ard)
```

%pasa el sistema coordenado del kinect con el de matlab

```
est_esqueleto.vecxs = -est_esqueleto.vecx;
```

```
est_esqueleto.vecys = est_esqueleto.vecz;
```

```
est_esqueleto.veczs = est_esqueleto.vecy;
```

```
hold on
```

% dibuja el brazo izquierdo

```
plot3(est_esqueleto.vecxs(5:8),est_esqueleto.vecys(5:8), est_esqueleto.veczs(5:8));
```

```
plot3(est_esqueleto.vecxs(5:8),est_esqueleto.vecys(5:8), est_esqueleto.veczs(5:8),'*r');
```

% dibuja el brazo derecho

```
plot3(est_esqueleto.vecxs(9:12),est_esqueleto.vecys(9:12), est_esqueleto.veczs(9:12),'k');
```

```
plot3(est_esqueleto.vecxs(9:12),est_esqueleto.vecys(9:12), est_esqueleto.veczs(9:12),'*r');
```

% dibuja hombros

```
plot3(est_esqueleto.vecxs([5,3,9]),est_esqueleto.vecys([5,3,9]), est_esqueleto.veczs([5,3,9]));
```

```
plot3(est_esqueleto.vecxs([5,3,9]),est_esqueleto.vecys([5,3,9]), est_esqueleto.veczs([5,3,9]),'*r');
```

% dibuja cabeza

```
plot3(est_esqueleto.vecxs(3:4),est_esqueleto.vecys(3:4), est_esqueleto.veczs(3:4));
```

```
plot3(est_esqueleto.vecxs(3:4),est_esqueleto.vecys(3:4), est_esqueleto.veczs(3:4),'*r');
```

```
if POSTURA_CUERPO==1
```

% dibuja pierna izq

```
plot3(est_esqueleto.vecxs(13:16),est_esqueleto.vecys(13:16), est_esqueleto.veczs(13:16));
```

```
plot3(est_esqueleto.vecxs(13:16),est_esqueleto.vecys(13:16), est_esqueleto.veczs(13:16),'*r');
```

% dibuja pierna der

```
plot3(est_esqueleto.vecxs(17:20),est_esqueleto.vecys(17:20), est_esqueleto.veczs(17:20));
```

```
plot3(est_esqueleto.vecxs(17:20),est_esqueleto.vecys(17:20), est_esqueleto.veczs(17:20),'*r');
```

% dibuja cadera

```
plot3(est_esqueleto.vecxs([1,2,3]),est_esqueleto.vecys([1,2,3]), est_esqueleto.veczs([1,2,3]),'k');
```

```
plot3(est_esqueleto.vecxs([1,2,3]),est_esqueleto.vecys([1,2,3]), est_esqueleto.veczs([1,2,3]),'*r');
```

% dibuja cadera izq der

```
plot3(est_esqueleto.vecxs([13,1,17]),est_esqueleto.vecys([13,1,17]),  
est_esqueleto.veczs([13,1,17]));
```

```
plot3(est_esqueleto.vecxs([13,1,17]),est_esqueleto.vecys([13,1,17]),
```

```
est_esqueleto.veczs([13,1,17]),'*r');
```

```
end
```

```
A=eye(4);
```

```
dibujar_sistema(A, 1, 'M')% sistema de cordenado matlab kinect
```

```
xlabel('X');
```

```
ylabel('Y');
```

```
zlabel('Z');
```

```
axis vis3d equal;
```

```
view(45,24);
```

```
grid on;
```

```
angulos(est_esqueleto.vecxs, est_esqueleto.vecys, est_esqueleto.veczs, ard);
```

```
end
```

Ángulos

% esta función interpreta los puntos que detecto Kinect en la función anterior y por medio de funciones geométricas se tiene una aproximación de los ángulos en las que se encuentra cada articulación del brazo robótico

function angulos(vecxs, vecys, veczs, ard)

% punto del hombro derecho con respecto al sistema M

pmh=[vecxs(9);

vecys(9);

veczs(9)];

% punto del hombro izquierdo con respecto al sistema M

pmhi=[vecxs(5);

vecys(5);

veczs(5)];

% vector entre hombro izq - der con respecto al sistema M

vmeh=pmh-pmhi;

% vectores Y, X, Z del hombro der con respecto al sistema M

ymh=[0 0 1]'; % ymh=yk=Zm

zmh=vmeh/norm(vmeh); % vector z con respecto a matlab hombro

xmh=cross(ymh,zmh); % producto cruz entre el vector Y*Z

% genera matriz donde se almacenara los vectores XYZ hombro con respecto a

% matlab

Amh=[xmh,ymh, zmh, pmh;

0, 0, 0, 1];

dibujar_sistema(Amh, 0.2, 'h') % dibuja sistema h en el hombro

Ahm=inv(Amh); % inversa de la matriz Amh

pmc=[vecxs(10);

vecys(10);

veczs(10);

1];

phc= Ahm*pmc; % punto hombro codo

q1=round(atan2d(phc(3),-phc(2))); % angulo de abd-ad

%% -----

yp=norm(rotx(-q1)*phc(2)); % vector del codo con rotacion del angulo q1

q2=sign(phc(2))*round(atan2d(yp,phc(1)))+90;

% punto matlab muñeca

pmm=[vecxs(11);

vecys(11);

veczs(11);

1];

phm= Ahm*pmm; % punto hombro muñeca

vcm=phc-phm; % diferencia entre vectores codo-muñeca

%% -----

q3=plano(pmh,pmc,phm) % funcion que genera el plano para encontrar q3

%% -----

q4=180-round(acosd((dot(phc(1:3),vcm(1:3)))/(norm(phc(1:3))*norm(vcm(1:3)))); % aplicación teorema

% del coseno para el ángulo del codo

ang=[q1 q2 q3 q4 0]'

end

Apéndice C

Cinemática Directa E Inversa Del Brazo Robótico InMoov

Cinemática Directa

```
%% función definitiva
% función que permite por medio de los parámetros de DH calcular las matrices de transformación
homogénea e enviar la matriz A05 a la función cinemática inversa
function robot=cin_dir_inmoov2(q)
% figure(2)
L1=0.05;L2=0.24;L3=0.22;
tam=0.05;
A =eye(4);
A0 = [A(1:4,2) A(1:4,3) A(1:4,1) A(1:4,4) ];
qa = [ q(1),90+q(2),q(3),q(4),q(5)];
d = [ 0, L1, L2, 0, L3 ];
a = [ 0, 0, 0, 0, 0 ];
al = [ -90, 90, -90, 90, 0 ];
%-----
A01=dnavit(qa(1), d(1), a(1), al(1));
A12=dnavit(qa(2), d(2), a(2), al(2));
A23=dnavit(qa(3), d(3), a(3), al(3));
A34=dnavit(qa(4), d(4), a(4), al(4));
A45=dnavit(qa(5), d(5), a(5), al(5));
%-----
A01=Roty(90)*A01;
A02=A01*A12;
A03=A02*A23;
A04=A03*A34;
A05=A04*A45;
%-----
p0=[0, 0, 0, 1]';
p01=A01*p0;
p02=A02*p0;
p03=A03*p0;
p04=A03*p0;
p05=A05*p0;
%-----
robot.A01=double(A01);
robot.A02=double(A02);
robot.A03=double(A03);
robot.A04=double(A04);
robot.A05=double(A05);
%-----
dibujar_sistema(A0,tam,'0');
dibujar_linea( p0, p01, 'k', 1 )
% -----
% dibujar_sistema(A01,tam,'1');
dibujar_linea ( p01, p02, 'k', 1 )
% %-----
dibujar_sistema(A02,tam,'2');
dibujar_linea( p02, p03, 'k', 1 )
% %-----
% dibujar_sistema(A03,tam,'3');
dibujar_linea( p03, p04, 'k', 1 )
% %-----
```



```
dibujar_sistema(A04,tam,'4');  
dibujar_linea( p04, p05, 'k', 1 );  
% %-----  
dibujar_sistema(A05,tam,'5');  
axis vis3d equal;  
axis([-0.30 0.30 -0.10 0.50 -0.55 0.50])  
view(152,32);  
xlabel('x');ylabel('y');zlabel('z');  
end
```


Cinemática Inversa

%%% función definitiva.

% recibe la matriz de transformación homogénea A05 que contiene la posición del efector final, y por medio de cálculo geométricos se hallan los ángulos de cada articulación.

function qa = cin_inv_inmoov2(T)

% Parámetros Denavit-Hartenberg del robot

view(130,42);

L1=0.05;L2=0.24;L3=0.22;

d = [0, L1, L2, 0, L3];

a = [0, 0, 0, 0, 0];

al = [-90, 90, -90, 90, 0];

codo=1;

p = Roty(-90)*T; % Posición de la mano del manipulador

pp = p(1:3,4); % Posición de la mano del manipulador

pm =pp-(L3*p(1:3,3));

px=pm(1);

py=pm(2);

pz=pm(3);

% Solución de la primera articulación: q1

m=sqrt(px.^2 + py.^2);

D=sqrt(m.^2 - L1.^2);

alfa=atan2d(py,px);

beta=atan2d(L1,D);

q1=alfa-beta;

% Solución de la segunda articulación: q2

alfa2=atan2d(-pz,D);

beta2=atan2d(0,D);

q2=alfa2-beta2;

% % Solución de la tercera articulación: q3

A01=denavit(q1, d(1), a(1), al(1));

A12=denavit(90+q2, d(2), a(2), al(2));

A01=Roty(90)*A01;

A02=A01*A12;

x2=A02(1:3,1);

y2=A02(1:3,2);

z2=A02(1:3,3);

z5=T(1:3,3);

s=cross(z2,z5);

if s~= 0

z3=(s)/(norm(s));

cq3=dot(y2,z3);

sq3=dot(x2,z3);

q3=atan2d(-sq3,cq3);

else

q3=0;

end

% Solución de la cuarta articulación: q4

A23=denavit(q3, d(3), a(3), al(3));

A03=A02*A23;

x3=A03(1:3,1);

y3=A03(1:3,2);

z3=A03(1:3,3);

z4=T(1:3,3);

cq4=-dot(y3,z4);

sq4=dot(x3,z4);

q4=atan2d(sq4,cq4);

% Solución de la quinta articulación: q5

A34=denavit(q4, d(4), a(4), al(4));



A04=A03*A34;
x4=A04(1:3,1);
y4=A04(1:3,2);
x5=T(1:3,1);

cq5=dot(x4,x5);
sq5=dot(y4,x5);
q5=atan2d(sq5,cq5);

qa = [q1 q2 q3 q4 q5]';

DQS is member of:

